The Integrated Design Exploration and Analysis (IDEA) Process

by

Andrew Masur
B. Eng (Aerospace Engineering)
Ryerson University, Canada, 2004

A thesis

Presented to Ryerson University

In partial fulfillment of the

requirements for the degree of

Master of Applied Science

in the Program of

Mechanical Engineering

Toronto, Ontario, Canada, 2006

Author's Declaration

I hereby declare that I am the sole author of this thesis.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this thesis by photocopying or other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Abstract

The Integrated Design Exploration and Analysis (IDEA) Process
Andrew Masur, B. Eng. (Aerospace Engineering) 2004
MASc (Mechanical Engineering)
Ryerson University

Concept design is one of the most important and confusing phases in engineering design. The IDEA process was created to help reduce this confusion and improve concept design practices within the engineering industry. An analysis of existing concept design methods was conducted in order to identify areas for improvement. A systematic process called IDEA was created to address these areas and assist designers with concept design. In addition, a software interface was created in order to support the IDEA process and improve concept design efficiency. The use of automated tools in the interface helps to alleviate many of the "bookkeeping" tasks in the concept design process. The interface was written in the open source program Compendium. Three multi-disciplinary case studies were conducted to validate the process. The use of IDEA lead to an increase in the number and variety of concepts generated.

Acknowledgements

It seems like yesterday that I wrote the acknowledgements for my undergraduate thesis, but here I am two years later, trying to think of the best way to thank people who have helped me so much. Without further ado here it goes...

Professor F.A. Salustri: What can I say, you've gone above and beyond the call of what any advisor should do. Between fighting to get me funding in that first hard year, to helping extricate this paper from the soup that I had created all I can really say is thank you. The last three years have been some of the most trying of my life, and through it all you have been there for me, you've been more than an advisor, you've been a friend who has helped me when things were at their hardest. I will always be more thankful than you can ever know.

Mom and Dad: What really needs to be said? You've walked side by side with me throughout this 25 year odyssey that has been my life. In that time, you've always put my interests above your own, did what was best for me even if you did not want to do it. Through your dedication and love you have given me a past that I can always look fondly on, but more than that you have given me the greatest gift that any parents ever could: a future. A future full of possibilities, limited only by my own potential. So I say thank you from the very fibre of my being. Thank you for your love, thank you for your support, and I hope I have at least partially lived up to the expectations you set for me so many years ago.

Mikey: It my acknowledgements page and I'll call you Mikey if I want ☺. Although we haven't always seen eye to eye, I hope that the last few years have brought us closer together. As I've grown older I've watched you grow from a tiny squirt who followed in my shadow, to a young man whose talents and accomplishments will far exceed my own. I am so very proud of you and what you have done in life. May it treat you well and give you all of the rewards that you so richly deserve. Just be sure to give me a wave on your way by.

Grandma and Grandpa: This is not so much a thank you for school, as it is a thank you for my childhood. Thank you to Grandma for always listening when I complained, and for providing good advice even if I didn't want to hear it. Don't worry, I'm sure we'll grow all of my hair back before you're done with me ☺. Grandpa, what you have done for me does not need to be said. When Mom went back to work all those years ago you took care of Mikey and I like we were your own sons. You went so beyond what is expected of a Grandpa that words cannot describe the gratitude and love that Mikey and I have for you. You gave us stability and love when most people would look the other way. So thank you for babysitting, thank you for keeping secrets, thank you for all the lunches, and thank you for those great Nutella sandwiches that you made.

This work is dedicated to my Grandpa.

Although you were only able to see me finish one degree, this one was done in your memory. You always saw potential in me, and made sure that I was reaching it. I hope that I have made you proud, and wherever you are I hope that I will always do so. Until that day that we can meet again, sleep well and I love you. Always.

Tears are sometimes an inappropriate response to death. When a life has been lived completely honestly, completely successfully, or just completely, the correct response to death's perfect punctuation mark is a smile.

***Julie Burchill** (b. 1960), British journalist, author. Quoted in: Independent (London, 5 Dec. 1989).*

Table of Contents

List of Figures

List of Tables

List of Appendices

1.0 Introduction to Concept Design

1.1 The Engineering Process

Engineering is ubiquitous in the modern world. From the cars we drive, to the buildings we work in, to the mobile phones that keep us connected, almost any technology that we use in our daily lives has been touched in some way by an engineer. Indeed, an engineer is simply an individual who applies science and technology to address some perceived need in order to improve the quality of life for society [1]. The main question that most people have is how does an engineer do this?

Many see engineering as simply nothing more than taking advantage of a "eureka" moment, a brilliant instant of invention that solves a problem that society did not even know it had. While these flashes of brilliance have historically lead to some of the greatest advances in human technology, and continue to do so today, engineering is more commonly a methodical process of identifying a problem and solving it in a systematic manner. The vast majority of technical progress is evolutionary, rather than revolutionary. Designers (from this point forward the term "designer" will stand for "design engineer") look at problems with current products and solve them in the next revision. New designs are built upon the foundations of old ones, leveraging what worked before, and improving upon what did not.

Finding concepts that embody solutions to design problems is the essential heart of the engineering process. However there are many different approaches to this process. To appreciate this, one may consider a few exemplars.

Figure 1 illustrates the engineering design processes currently in use at the ESA (European Space Agency), NASA, and the U.S. DoD (Department of Defence) [2]. While the fine details of each process differ, the overall outlines are remarkably similar, a fact that most engineering processes share.

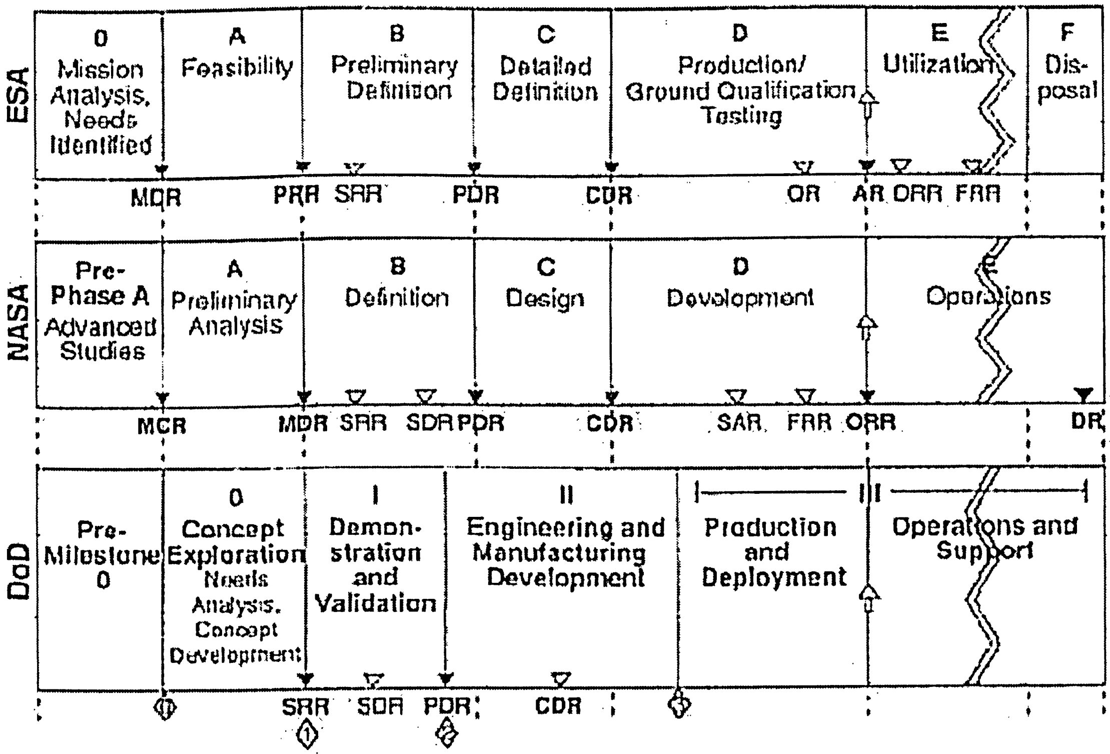


Figure 1: ESA, NASA, and DoD Engineering Processes [2]

Each of these processes can be broken down into four distinct phases. The first of these is a problem exploration and concept design phase. In this phase, engineers gain a better understanding of the problem itself. Indeed, in many situations determining the problem is the most difficult task, and once accomplished the solution is often readily obtained.

With a better understanding of the problem, the next task facing designers is to determine a solution. Unlike mathematics problems, however, engineering problems rarely have a single correct solution. A number of seemingly disparate solutions will usually be acceptable for the problem at hand, and the designer's task becomes selecting the "best" one.

Once the "best" concept has been selected, the design process enters its second phase: detailed design. While the concept design phase outlines the overall vision for the final product, the detailed design phase of a project is tasked with bringing those visions to reality. Components and interfaces are designed and selected, and analyses are carried out to determine the performance of designs. In large projects, it is likely that prototyping is performed in order to validate the design as it progresses [2]. The detailed design phase also usually plays host to a seemingly endless supply of difficulties and delays as a more in-depth analysis of the selected concept invariably reveals weaknesses not conceived of in the earlier stages of design.

However, if the concept design phase of the project has been performed with reasonable care, many of these problems and delays are easily overcome and thus, do not threaten the completion of the project. The final result of the detailed design phase is a complete specification of the final product, including blueprints, to move the project to the next phase of the engineering process: production.

The production phase is concerned with manufacturing the product and its distribution to market. Initially, these may not seem like areas where an engineer would be involved. However, closer inspection reveals many areas of potential engineering involvement, albeit on a much smaller scale than in the design phases. It is quite probable that difficulties may arise during manufacture, ranging from the clarification of blueprints to minor redesigns of a component in the design. Ideally, these errors can be reduced by the inclusion of manufacturers during the design process, but this is not always possible; and even if manufacturers were involved, other unforeseen problems can arise. Another possible area of engineering involvement during manufacturing is the design of the actual manufacturing processes. This is the domain of the industrial engineer and is as important as the design of the product itself.

The final phase of the engineering process, operations and support, deals with the day-to-day operation of the product [2]. In the case of simple consumer products, the engineering team's involvement with the product after it is shipped is likely to be minimal. However, more complex products may require involvement if unforeseen problems arise. For example, if a common flaw is found in a model of an automobile, the engineering team may be required to determine if it is merely a manufacturing defect, or a design flaw. In addition, the team may need to determine the appropriate solution for the problem or provide advice on how the company should handle the situation.

In another example, space missions often have large engineering teams involved in the operations phase. Space exploration is still a relatively new area of technology involving the use of extremely complex machines. Problems often emerge during a mission, and it is the job of the engineering team to determine solutions quickly to ensure the success of the mission. The fact that the Mars exploration rovers Spirit and Opportunity have, at the time of this writing, operated almost nine times [3] longer than their design lifetimes is a testament to the continued involvement of the engineering teams during the operations phase of the mission [26] [27].

1.2 The Importance of Concept Design

Now that we have a better understanding of the overall structure of the engineering process, the next question that naturally comes to mind is "Which of these phases is the most important?" This is, of course, an unfair question in that it has no real answer. The detailed design phase is pointless without the overall direction defined in the concept design phase, and likewise the whole engineering process becomes somewhat academic without a manufacturing phase to bring products into reality.

In light of this, the real question becomes "Which of these phases has the most impact on the success of the final product?" This is a more reasonable question and the answer is far more useful when trying to optimize the design process. Because all of the phases of design are so highly interconnected, a weakness in any one phase also affects all of the others. Logically, any improvements to the phase that has the greatest effect on the design will also greatly improve the entire design process. The problem becomes identifying this phase. This can be accomplished by looking backward through the design process on a number of hypothetical designs.

First let us imagine a case where a product has made it to market, but does not capture the public's interest and thus does not sell well. The product is well manufactured and engineered, thus build quality and functionality is not an issue; it simply seems that the buying public is uninterested in what the product has to offer.

By looking back through the phases of the design process, we can try to find where this problem occurred. Operations and support will have little effect in this case. The product is of good quality; therefore, bad word of mouth about the product caused by manufacturing defects is not an issue. The third phase of the design process, manufacturing, is discounted since product quality is acceptable; thus, any problems encountered during manufacturing are transparent to the consumer. Similarly, the problem could not have originated during detailed design. The product is known to function as intended; thus, once again there is little chance of bad word of mouth being spread by the consumer.

This leaves only the concept design phase as the source of the problems, and indeed this is where they are to be found. The sales problems for the new product have arisen because it is not tailored to its target market properly. Customers see the product either as redundant, or of no use to them. In any case, they are disinterested in the product, and thus do not buy it.

Another hypothetical case could involve a product that is experiencing large development time and budget overruns during detailed design. The concept design phase of the project seemed to go smoothly and very quickly, but the engineering team working on the detailed design is having difficulty integrating certain parts of the product. These problems have lead to many redesigns, which increase both the time and money required to complete the project greatly. In addition, the time delays cause scheduling difficulties with both manufacturing and distribution. In the real world, a project encountering this difficulty would be in danger of being cancelled to minimize their losses.

The problems in this case can again be traced directly back to the concept design process. The clue in this case is that the concept design process is said to have gone very quickly. While it is desirable to minimize the length of any phase of the design process, it is not acceptable to cut corners in order to achieve this. By moving through the concept design process too quickly it is likely that the designers did not explore all possible design options and instead selected a concept which, in retrospect, is less than ideal. Again, a mistake made in the early stages of the design process is having detrimental effects on all downward streams. If the product does eventually make it to market, it will be at a far greater cost than originally intended and in the end these cost overruns will have to be borne by the consumers.

While there are many other possible cases which could be considered, it is already clear that the concept design phase of any engineering project has a significant impact on the subsequent phases. Due to the connected nature of the design process, any mistakes made during concept design are amplified as they move through subsequent phases. If these mistakes are only discovered in the later, more detailed, phases of the design process, they may be very difficult and costly to resolve. This can lead to both time and cost overruns for a project, and can sometimes lead to its cancellation. In many respects, the concept design phase has the most significant impact on the overall success of any design project since it is the first phase and decisions made here have consequences that can be felt throughout the rest of the design process. Because of this, every attempt should be made to conduct concept design as thoroughly as possible; any mistakes that can be avoided here will greatly help to reduce potential problems later on in the design process.

With this in mind, the overall goals of concept design are threefold.

1.) Accurately determine the needs and desires of the customer.
2.) Thoroughly explore all possible options for the final design in order to maximize the possibility of finding the "best design".

3.) Select the "best" concept in a rational manner which engenders confidence in the results.

Each of these goals can be viewed as a pillar for a concept design process. If a proposed concept design process is able to meet all three of these criteria, then the probability of a successful concept design phase is increased.

1.3 Project Motivation

Because of the importance of concept design there has been a great deal of study conducted on how to make it more effective. These studies have resulted in the development of a number of different tools to help designers. Section 2 will explain some of these tools in detail, however a complete review of all techniques is beyond the scope of this work simply because there are so many of them.

One of the drawbacks of many of these techniques is that they are very similar to each other with only subtle differences to distinguish them. This can cause a great deal of confusion when trying to select the best tool for the current situation. By its very nature concept design is a confusing process. There is very little hard information available to designers at this stage of design, and when confronted with the vast array of techniques available to them designers can have a difficult time deciding which tools to use. So while it is generally desirable to have a large assortment of tools available for a task, in this case it can actually be detrimental since the vast array of available techniques is actually part of the reason for designer confusion.

A good way to minimize this confusion would be to create a systematic process for concept design. Such a process would use a fixed set of concept design techniques in a specific order, regardless of the design problem. This would help designers by allowing them to focus their creative energies on the *problem*, rather than the *process*. In addition, if such a process adhered to the three pillars of concept design it would not only reduce designer confusion, but would also help to improve the overall results of concept design.

These goals led to the creation of the Integrated Design Exploration and Analysis (IDEA) process for concept design. IDEA uses existing tools to help designers thoroughly explore all aspects of a design problem. This not only leads to better concepts through a better understanding of the problem, but the use of existing tools and techniques allows IDEA to be familiar and flexible enough for integration into a company's current engineering practices. Instead of forcing designers to learn yet another tool, IDEA is an attempt to show designers when they should employ the tools they already know. It also forces them to look at many different aspects of the design problem which helps to reduce oversights and improves the quality of generated concepts.

2.0 Current Techniques in Concept design

During the background research for this thesis it quickly became evident that many of the techniques currently used in concept design and engineering decision making seemed like they could be used together to help solve design problems. In many cases the outputs from one technique could be used as the inputs for another with no, or very minor, modifications. By using these tools in the proper order it was theorized that a process could be created which would adhere to the three pillars of concept design. In this section, some of the tools and methods used in IDEA are reviewed.

2.1 Capturing Customer Needs

As previously written, the first goal in any design problem is determining what the problem is. While this may sound obvious it is often forgotten in the rush to create designs and the problem is that designers rarely have any idea exactly what the specifics of the problem are and what those designs should be. Engineering is nothing more than finding solutions to problems. Designers do not wake up in the morning and proclaim, "Today I will design something!" Instead, companies or individuals become aware of some need in society, or something about a current product that makes people unhappy, and attempt to solve it. The difficulty is to separate what customers *want* out of a new product from what they *need* from that product. While the difference between the two may seem subtle, it can actually have very significant impact on the design process.

Customer wants and needs are often derived from very similar desires, but what a customer wants is often more unrealistic than what they really need. For example, most of us would want a car that gets around 80 miles per gallon, but at the same time many people are not willing to give up performance or pay the necessary price to get such a car. Therefore, what these people *want* is a car that gets 80 miles per gallon, but what they actually *need* is a car that gets better fuel economy than it is getting now, with acceptable performance and selling for, at most, a small premium. In reality, the designer is trying to increase the *perceived quality* of a product from the viewpoint of the customer [4].

What is quality exactly? In his paper "Stimulating Creative Design Alternatives Using Customer Values" Keeney states that quality is defined by *those features and aspects of the product that are more highly valued by potential customers* [4]. Therefore, the issue now becomes determining exactly what features and aspects will deliver the desired quality to customers and entice them to purchase it.

The answer to this problem is to simply ask potential customers exactly what characteristics they would like to see in the product [4]. Because the customer determines whether a product is successful, it makes sense to determine exactly what will interest them.

When done properly, customer information is gathered from a wide variety of individuals. These individuals can have many different characteristics including lifestyle, economics, technical literacy, and other factors. Indeed, information on why customers might *not* want a particular product is often just as valuable as information about why they would. If possible, interviewing people involved with the design of similar products can also be very useful. Conducting face-to-face interviews with prospective customers is also a good way of determining this information [4]. The level of interaction between the interviewer and the interviewee cannot be duplicated with any other method because it allows the interviewer to probe the individual's knowledge more deeply and gain a better understanding of that person's preferences while letting the interviewee focus on the questions [4], further improving results.

In the case of widely marketed consumer products, samples sizes of fifty or more individuals are not uncommon [4], and standardized questionnaires or group interviews become a far more feasible method.

In cases of products that will not see widespread consumption (e.g. aircraft), detailed discussions and negotiations with the client is the norm, particularly to determine whether the requirements are realistic. These products tend to be more technologically advanced and usually costlier, making careful determination of the requirements even more important.

2.2 Exploring Design Options: Brainstorming and Tradespace Analysis

With a better understanding of customer needs, designers now have a better understanding of the problem. The task now is to find solutions to that problem. Several tools exist to engineers to assist in this task. These will be reviewed below.

2.2.1 Brainstorming

Brainstorming is one of the first tools to which most engineers are introduced, and happens to be one of the most useful. The thinking process during concept generation is often chaotic as designers generate and then reject many potential concepts. Designers often perceive this to be the most enjoyable part of the design process because of its creative aspects. Brainstorming is a successful and established way to promote creative conceptualisation and then to organise and record the resulting information. It is often the case during a brainstorming session that one designer's partial idea will spark the thinking of another designer [5] who will then take that idea in directions not imagined by the original designer. This effect can increase exponentially in well-functioning teams.

The basic rules and techniques for brainstorming are very simple, which is why, at least in the experience of this author, it is introduced to students as early as grade school. Because most people have been introduced to the technique at such a young age, designers feel a great deal of familiarity and confidence in the technique when it is applied to a design project, accounting for its popularity.

The basic overriding rule [5] for a brainstorming session is that all ideas, no matter how strange they may seem, are discussed without criticism. This is very important since it creates an environment where people are not afraid to share their ideas, which in turn leads to a better chance of the group generating novel ideas.

The second important factor for successful brainstorming is the make-up of the group itself [5]. The group should includes one or two of what Colwell terms "idea-fountain" types, at least one engineer with a great deal of project experience, at least one person who strongly wants results out of the brainstorming session, and finally a group leader who is able to keep up with the "idea-fountain" individuals [5]. The role of the "idea fountain" individuals is to create many concepts quickly. Most of the novel concepts during the brainstorming session will come from these individuals, yet there should be no more than two of them in any group since they also tend to take over the discussion which can lead to internal friction [5]. The rest of the group members are placed in various roles in order to act as checks to and support the "idea fountain" individuals.

The experienced engineer ensures that the brainstorming session remains relevant to the project at hand. Due to their experience, these individuals have a basic understanding of what might work. The results-oriented team member ensures that the discussion stays on track. The final individual in the group, the team leader, organizes the meeting and serves as a gatekeeper for all of the concepts generated by the group. The team leader should be able to keep up with the "idea fountain" individuals, but also take a step back and analyze the merits of each concept to catch potential flaws as early as possible [5].

Brainstorming usually involves some simple visualization techniques, such as the technique IBIS demonstrated here [6]. The process begins with a blank surface that can be drawn upon. This can be a blank sheet of paper, a blackboard, or a blank file in a drawing program. The problem under discussion is written at the center of the page and a circle or oval is drawn around it forming a central node. Possible solutions to the problem are then arrayed around the central node in a ring. Each node is connected to the central node with a line so that the diagram now resembles a spoked wheel. Each possible solution may have additional levels of arguments and questions arrayed around it so that the entire brainstorming diagram resembles a large web of information. An example of this type of structure is shown in Figure 2.

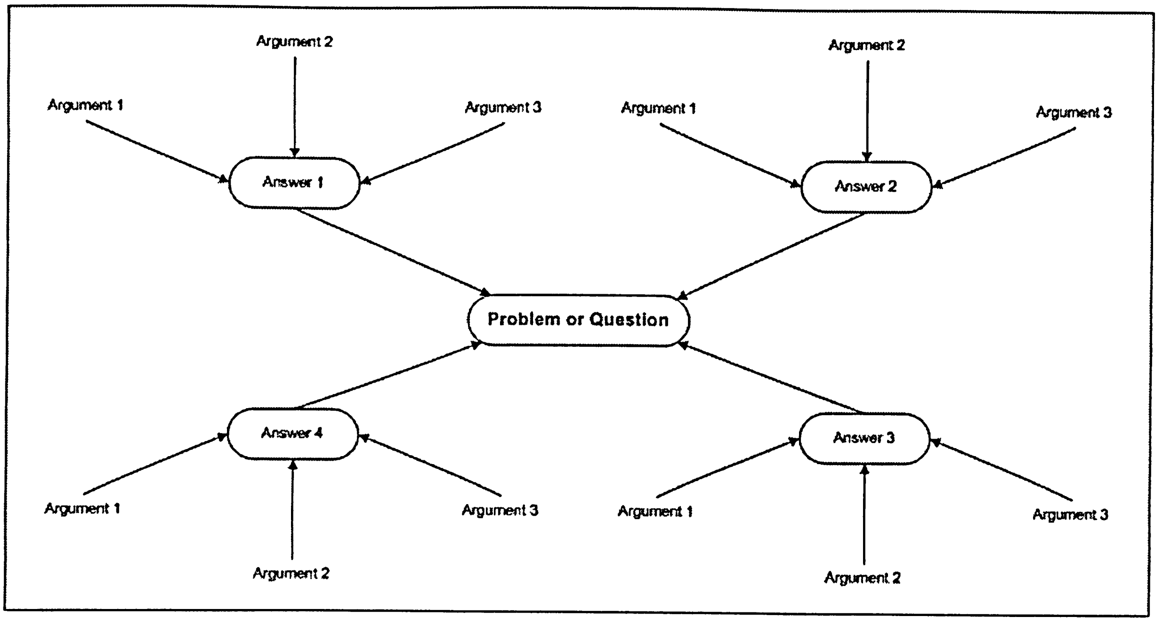


Figure 2: Basic Brainstorming Setup

This type of configuration not only helps to organize the large amounts of data that are generated during the brainstorming process, but also assists with the actual generation of ideas. The structure of the diagram forces the designers to look at the problem in a very thorough and stepwise fashion which helps the designers to not only catch flaws in the current concepts, but also helps them to identify areas of potential improvement. This will naturally lead to the generation of improved concepts, which, as we have already seen, is of great benefit to the overall design process.

2.2.2 Trade Space Analysis

Conceptualizing new design solutions can be thought of as searching through an abstract space of all possible designs. Each design in this space represents a different set of "trade-offs" between design characteristics; one then searches for the design that provides the best trade-off with respect to a given set of requirements. Thus, "trade-space analysis" is a general framework to help designers find suitable designs. This abstract space is known as an *Unconstrained Trade Space*.

When we study the trade space for a particular design problem we are actually looking at a subset of the entire trade space, where the boundaries are defined by the problem requirements. This is known as a *Constrained Trade Space* (see Figure 3).

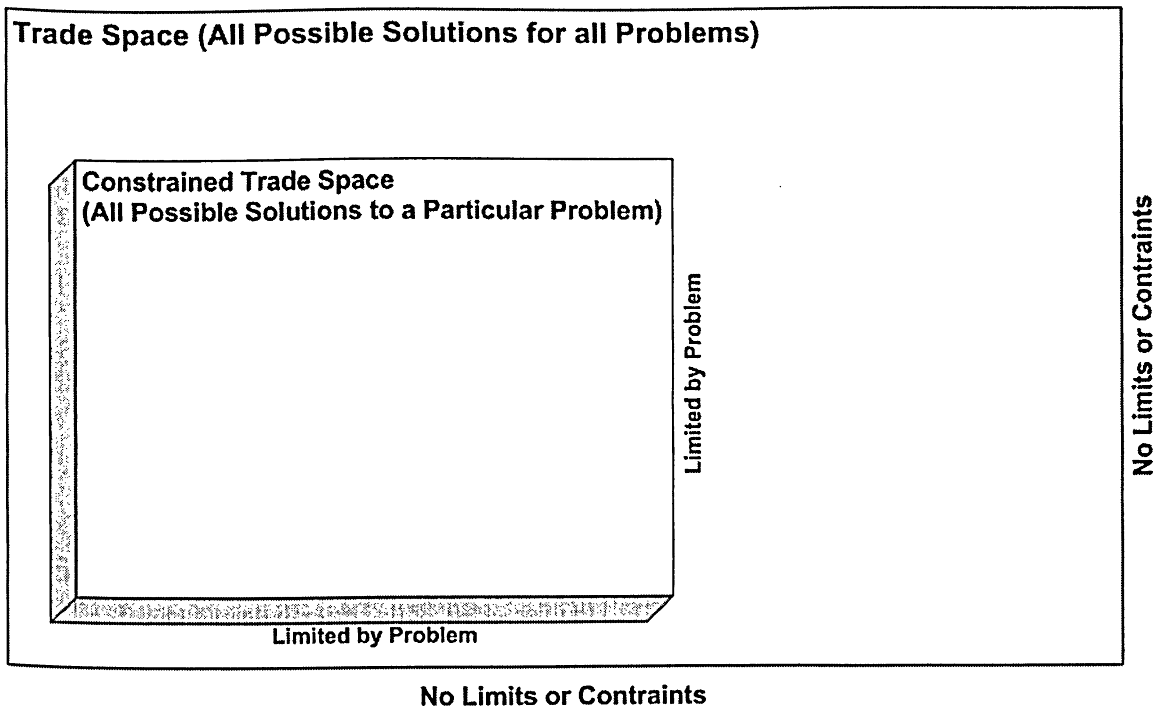


Figure 3: Trade Space Representation

The constrained trade space contains all of the possible solutions to the current design problem. The design team must determine which solutions are the best. From this point of view, we can see that the designer's job is not to *generate* new concepts, but to *discover* concepts that already exist.

Searching through the constrained trade space to find the "best" solution can be difficult. Even through the size of the constrained trade space has been greatly reduced from the original unconstrained trade space, it is still far too large to make a simple brute force search feasible. The key of concept design is to focus in on the right part of the Trade Space to increase the likelihood of finding the best solution. This is demonstrated below in Figure 4.

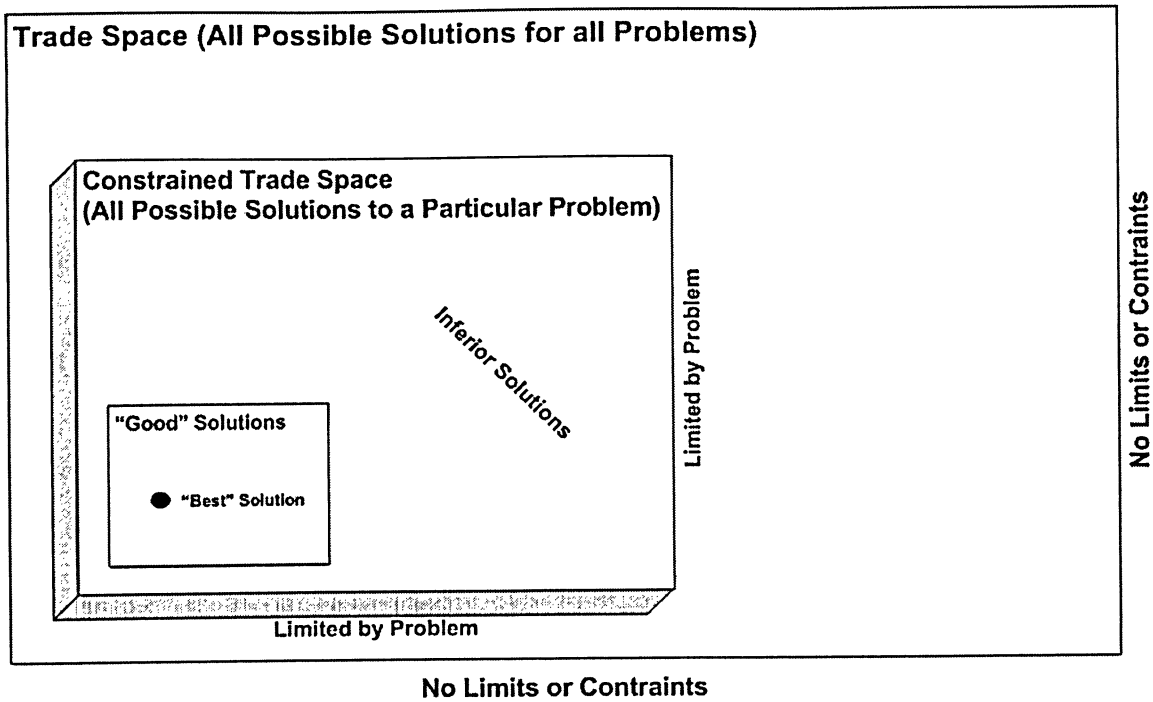


Figure 4: Locating "Good" Solutions in the Trade Space

Therefore, there are two tasks in trade space analysis. First designers must gain a better understanding of the design problem in order to reduce the size of the trade space as much as possible, and second they must somehow identify the "best" concept out of those that remain.

There is no simple solution to gaining a better understanding of the problem. Because each design problem is different there is no standard procedure; the factors which are important to one design may have no bearing on another. Thus, the analysis required to reduce the size of the trade space also changes from problem to problem.

For example, in his study of low cost robotics for space exploration, Smith [7] was interested in the number of small biomorphic robots which would have to be sent to other planets in order to ensure various probabilities of mission success. Biomorphic robots are small devices that mimic the characteristics of various types of animals [7] in order to move throughout their environment. Several key parameters for their design were identified: cost, reliability, and number of robots. A trade space covering feasible values for these parameters was created by plotting the probability of mission success against the parameters. This identified a constrained region that was valuable to help designers select the best combination of these parameters.

As stated previously, the second task of trade space analysis is to assist designers in identifying the most suitable concept for the current design problem. One of the most straightforward and rigorous methods for finding the "best" concept is to simply generate as many concepts as possible [8]. Generating more concepts allows a more thorough exploration of the design space that in turn increases the probability of generating the "best" concept. However, generating large numbers of different concepts is quite difficult. It is natural to try to look at the "whole picture" at once when attempting to solve a problem; a similar phenomenon occurs when generating concepts. Engineers tend to envision concepts in their entirety, or focus on one aspect of the design and create concepts to suit it at the expense of all others. In both cases, there is often an initial spark of creativity leading to several concepts, and then concept generation drops off. This is caused by the fact that a designer's mind often becomes locked into concepts that are very similar to those already generated; moving away from that line of thinking can be quite difficult. Trade space analysis can help to break this mental impasse which can be thought of as a "designer's block". Using trade space analysis to generate *Trade Space Diagrams*, like the example shown in Figure 5, can help to reduce "designer's block".

ARCHITECTURAL APPROACH:	Sciencecraft	Traditional AO	JIT	Modular	Redundant	Single String
IMPLEMENTATION APPROACH:	Integ. Prod. Team	Co-Location	Paperless	Traditional	Fire & Ice	Inter-program buys
LAUNCH VEHICLE.	Proton	Molnya	Delta	2 L/V's	1 L/V	
UPPER STAGE(S):	PAM-D	PAM-D/STAR 30	STAR 63/37/27	none		
TRAJECTORY TYPE:	Direct	VVVJGA	JGA	3+ΔVEJGA	SEVGA	SEEGA
LAUNCH & ARRIVAL DATES:	varies with trajectory type					
RADIATION ISSUES:	Low Radiation	High Radiation	Rad Hard Parts	Ta Shielding		
SCIENCE SCOPE:	1A Science	Grav. Assist Sci	Cruise Sci.	Probes	Asteroid	Kuiper Object(s)
	Long Exposure Imaging (1A)		Spin Scan Imaging		Imaging Resolution	
MISSION OPS APPROACH:	Beacon Cruise	Minimum Passes	Rip van Winkle	Flyby Automation		
S/C CONFIGURATION:	Non-Stackable	Stackable for 2 or more S/C on 1 L/V				
POWER SOURCE:	Cassini-based RTG	Advanced RPS	Solar	Powerstick	Dvmt. Schedule	
TELECOM ARCHITECTURE:	Ka	X	LGA/MGA ?	Optical	Ground Infrastructure	
DATA RATE OPTIONS:	HGA Size & Mass	Telecom Power	Data Compression			
ELECTRONICS TECHNOLOGY:	MCM μ-Electronics	Stacked MCM's	MIMIC's	Currently Available Technology		
ATTITUDE CONTROL:	3-Axis Stabilized	Spin Stabilized	Pointing Capability		Slew Capability	
	Body Pointed Sci.	Scan Platform				
PROPULSION:	Hydrazine	Cold Gas	Tridyne	SEP	Modular	Integral to S/C
THERMAL DESIGN:	Room Temp.	Cold	RHU's	"Thermos"		

Figure 5: Pluto Express Trade Space Diagram [9]

Figure 5 was generated for the proposed Pluto Express mission, the precursor to the New Horizons spacecraft that is currently en route to Pluto and the Kuiper Belt. We can see that the diagram is arranged as a series of rows, with each row representing some function or sub-system that is important to the final design. The diagram contains 17 distinct rows, resulting in a 17-dimensional trade space [9]. Next to each dimension are listed all of the possible options for that aspect of the design. This organization is where much of the power of the trade space diagram originates. Rather than generating whole concepts all at once, the structure of the diagram directs designers to look for *partial* concepts on a function by function basis. The creative energies of the entire team can be focused on finding all the alternatives for one particular functional sub-system. When all reasonable ideas have been found for one sub-system, the team moves onto the next one. The design team can also use this task to identify sub-system options which will not work. These options can then be disregarded, which has the advantage of reducing the Trade Space to a more manageable size. Conceptualizing on a sub-system or functional basis also helps to keep the design process fresh and interesting for the team, which can help in avoiding "designer's block."

Once all dimensions have been fully explored, whole concepts can be generated by mixing and matching the options from the various sub-systems. The previous work to determine the boundaries of the constrained trade space is valuable here since it allows designers to discount combinations that lead to unsuccessful concepts. This is very important because the Trade Space in Figure 5 contains several *billion* different combinations [9]. Clearly, this is many more concepts than could be generated otherwise. This allows for a more thorough exploration of the trade space, which in turn increases the chances of generating the "best" design.

While trade space analysis is very useful to enumerate and organize a design solution space, it does not offer guidance to select the "best" possible solution. For this, we must use other tools.

2.3 Making Design Decisions

As has been previously discussed, one of the greatest challenges in concept design is making that final choice of which concept will be selected for more thorough analysis. There are many tools available to assist designers with generating concepts, yet the entire concept design process is nothing more than an academic exercise unless a final choice can be made. The greatest difficulty with making this choice, however, is that the concept design phase, while arguably the most influential phase of the engineering process, is also the phase with the least amount of hard information. Much of the work is done based on supposition and educated guesses. Thus, we need to find techniques that not only help to alleviate this uncertainty from decision making, but also allow for traceability so that the design team can present evidence to back-up their decisions.

2.3.1 Arrow's Theorem

Before considering engineering decision making techniques, it is beneficial to review Arrow's Impossibility Theorem. Postulated by Kenneth J. Arrow in 1951 [10], the impossibility theorem deals with the issues inherent with social choice or group decision making problems which have very different properties than decision making by a single individual. This is obviously relevant to design decision making by teams.

When conducted by a single person, the decision making process reflects the values and judgements of that individual eliminating conflicts between different points of view.

In contrast, a social choice or group decision-making problem involves multiple participants, each with their own values and judgements. Problems arise from trying to reach agreement between all participants while still respecting the sovereignty of each individual decision maker [10].

While Arrow's Theorem was originally postulated for the study of social decision-making situations, such as the selection of candidates in a multi-party election, there have been a number of studies [10][11] to determine whether it applies to decision making within engineering design. As will be shown below social decision-making and engineering design decision-making are quite similar. Because of this similarity, it is important to gain a better understanding of the effects of Arrow's Theorem on engineering decision-making problems. Some of these effects can be quite detrimental and a better understanding of what causes them allows for the selection of engineering decision-making techniques to avoid these problems.

Social choice problems and engineering decision-making problems have one main characteristic in common: the aggregation of several weak orders into a single order [10]. An *order* is a ranking of alternatives according to some criteria. A *weak order* is defined as follows [10]:

Weak Order: A weak order on a set of alternatives X = {A,B,C,...} is a transitive binary relation $\succeq$ such that for any two elements A and B, either A $\succeq$ B (A is at least as preferable as B), or B $\succeq$ A (B is at least as preferable as A). Indifference is possible: if A $\succeq$ B and B $\succeq$ A, then one writes A ~ B (A is indifferent to B). If A $\succeq$ B but B $\not\succeq$ A, then A is (strictly) preferred to B, written A $\succ$ B.

That is, a weak order is a simple ordinal ranking. Thus, we know that alternative A is preferred to alternative B, but not by *how much.*

Consider a multi-party election. Let us imagine that a voter has a choice amongst three candidates A, B, and C. In an election, only the most preferred choice matters.

On the other hand, in many engineering problems there are situations where the order of all candidates is important, since deciding which candidate is second or third is just as important as deciding which one is in first. For example, knowing that mass is more important to a final product than cost can be just as important as knowing that cost is more important than size. This type of knowledge allows a design team to make informed decisions when conducting trade-offs against various design criteria.

Regardless of whether we are dealing with a social choice or an engineering decision, several weak orders created by different individuals are still being combined into a single social order. This similarity has lead to the continuing research into the application of Arrow's Theorem to engineering problems.

Arrow's theorem defines five axioms for the social choice problem and states that a voting procedure can only be considered "fair" if all five axioms are obeyed [12]. These five axioms are as follows:

1.) **Unrestricted Domain:** Each individual is free to order the alternatives in any way [10]. *This axiom ensures that individuals are free to make whatever choice they desire from the options presented.*

2.) **Positive Response:** If a set of orders ranks A before B and a second set of orders is identical except that individuals who ranked B before A are entitled to switch, then A is before B in the second set of orders [10]. *This axiom is important in cases where complex relationships are used to determine overall rankings.*

3.) **No Dictator:** The system does not allow one voter to impose their ranking as the group's aggregate ranking [12]. *This supports axiom 1 and ensures that all individuals are free to make any choice from the available options.*

4.) **No Imposed Orders:** There is no pair A, B for which it is impossible for the group to select one over the other [12]. *More plainly, this axiom states that each participant has a similar chance of their individual order being selected as the overall social order.*

5.) **Independence of Irrelevant Alternatives (IIA):** If the aggregate ranking would choose A over B when C *is not* considered, then it will still choose A over B when C *is* considered [12]. *This axiom ensures that each of the options is independent from one another and that the removal or addition of an option will have no effect on the rankings between the existing options.*

Arrow notes that for any problems of reasonable size (i.e. at least three voters expressing only ordinal preferences among more than two alternatives), one of these axioms must be broken in order for a realistic vote to take place [10]. Identifying which axiom should be violated in particular situations is an important and open issue.

For example, Scott and Antonsson [10] argue that engineering decision-making procedures that use a quantified scale to rank the performance of alternatives violate axiom 1. The very nature of many engineering problems ensures that the alternatives *cannot* be placed in any arbitrary order because of engineering requirements and constraints.

Arrow and others have attempted to resolve the intractability of the theorem by weakening the first axiom [10]. Their argument is that real political and economic systems are structured in such a way that the participants in these systems consequentially structure their choices in some logical manner that keeps contradictions and other similar problems from arising [10]. This is similar to the argument presented by Scott and Antonsson, that engineers will rank alternatives logically based upon available design information. In both cases, this removes the possibility of limitless choice on the part of participants within the system, thus technically breaking the first axiom while still keeping its spirit.

Let us now consider the effects of violating the other axioms. The effect of breaking axiom 2 is straightforward. Any decision-making procedure that breaks this axiom is inconsistent and of little value. Breaking axiom 3 implies that only one participant's vote really counts. This can happen in engineering settings (e.g. if a design process is behind schedule and a manager makes an "executive decision" to expedite it), but these cases are extraordinary and atypical. Violating axiom 4 would imply that each participant does not have a reasonable chance for their individual ranking to be selected as the overall ranking. Such a system would discourage participation since there would be no real reason for anyone to vote.

This leaves axiom 5 (usually referred to as IIA). A decision-making procedure that violates IIA has the possibility of changing the ranking between two alternatives if another alternative is added or removed. For example, let us assume that we have three alternatives ranked B $\succeq$ C $\succeq$ A. If alternative A were removed, we would expect the remaining two alternatives to retain the ranking B $\succeq$ C. In a decision-making technique that violates IIA, the removal of alternative A could cause a *rank reversal* between the remaining options, resulting in C $\succeq$ B. Clearly, this makes little sense; however it is not as devastating a result as violating one of axioms two through four. In many cases a decision making technique can be designed to avoid violating IIA, or at the very least to minimize the impact of the consequences.

2.3.2 Borda Counts and Pairwise Comparison Charts

One of the goals of any concept design process is to assist designers with making decisions in a rational and structured manner. Arrow's Theorem outlines many of the properties that any such decision making process must have. This section will introduce a decision-making technique known as *pairwise comparison* [12], one of the most widely used techniques in engineering design. It is especially useful for problems that require ordering a large number of concepts. This section will descibe and compare some of the popular variations of pairwise comparison.

The first instinct of most designers when asked to order a list of items is to look at the whole list at one time. Except for very short lists, this is a daunting task since there are typically many criteria that can affect the order. Pairwise comparison makes this task easier and more efficient by breaking the problem down into smaller and more manageable pieces. A number of different pairwise comparison techniques are available and each differs in not only the procedure for the technique, but also in the underlying logic that drives it. While the goal of each technique is the same, it is quite possible that different techniques will determine different orders for the same list. This begs the question of which one of these techniques is actually producing the true "best" list. Finally, each of these techniques differs in its sensitivity to the axioms of Arrows Impossibility Theorem.

One common pairwise comparison method is known as the *drop and revote* [12] technique. In this technique, the list of the items under consideration is placed into an initial weak order by each participant. These weak orders are then compared to some filtering criterion and the "losing" items are eliminated. This continues iteratively until only one item is left.

Two common filtering techniques are: (a) select the best of the best from amongst the available choices, and (b) to avoid the worst of the worst [12]. Both of these can be clarified through the following example based on the work of Dym, Wood and Scott [12]. Let us assume that we have twelve designers who have been asked to rank three design criteria, A, B and C, in order of importance. Each designer ranked the three criteria privately with no input from the other designers. Say the results are as follows [12].

1 preferred $A \succ B \succ C$ 4 preferred $B \succ C \succ A$

4 preferred $A \succ C \succ B$ 3 preferred $C \succ B \succ A$

According to the drop and revote technique, each of these weak orders must be filtered and recalculated. Let us proceed according to the best of the best option where we will drop the criteria which has received the least first place votes. We can see that criterion A receives 5 first place votes, B receives 4 first place votes, and C receives only 3. Thus, according to best of the best, we must drop C from further contention. This leaves us with the following results.

1 preferred $A \succ B$ 4 preferred $B \succ A$

4 preferred $A \succ B$ 3 preferred $B \succ A$

Applying the *best of the best* technique again we find that A receives 5 first place votes while B receives 7 making it the "winner" and the most important to the final design.

Let us now retry this problem using the *avoid the worst of the worst* option where the criterion that receives the most last-place votes is eliminated. Using the initial weak orders, we find that A receives 7 last place votes, B receives 4 last place votes, and C receives only 1. Thus, according to the rules for this option, we must eliminate criterion A. This leaves us with the following.

1 preferred $B \succ C$ 4 preferred $B \succ C$

4 preferred $C \succ B$ 3 preferred $C \succ B$

We can see from the above that C has received 5 last place votes while B has received 7; thus, avoiding the worst of the worst, we find the criterion C is the "winner".

Notice that the different filtering techniques produce different winners. This is actually quite problematic since the original weak orders were the same. The fact that the results are different not only raises questions as to which of the criteria is actually the most important to the design, but also as to the validity of the comparison technique itself.

A closer examination of the previous example reveals a number of problems with the *drop and revote* method. Firstly, the use of two different filtering techniques has resulted in two completely different results from the same input. The reason for this discrepancy can be found in the method itself. The *drop and revote* method of pairwise comparison is built on the elimination of the least desirable item and working with the remainder. In essence, it "throws away" data to converge on a result. Since different filtering options lead to different results, the *drop and revote* method violates the IIA axiom of Arrow's Theorem. This means the *drop and revote* method is of limited value in engineering design. Secondly, the drop and revote only results in the "best" alternative, while a complete ranking of all options is needed in design situations. Therefore, a different method is needed for engineering design situations.

Another pairwise comparison method that has been shown [12] to be consistent is the *Borda Count*, which can determine the "best" option in a single iteration. This is done by using a numerical rating scale that assigns points to each position in a weak order. These values are not meant to infer the level of preference between positions in the list; they are merely there to reduce the Borda Count method to a single iteration.

It is important to keep the point system linear. For example the values for a three element list can have the form (3,2,1) or (20,10,0). The highest ranked item receives the most points, the second highest item the second most points and so on down the list. Once numerical values have been assigned to the weak orders the points for each item are added across all weak orders and the total number of points that each option receives determines the final rankings.

Let us return to the previous example to better clarify this procedure: twelve engineers choosing amongst three options A, B and C. The results are as follows.

1 preferred A $\succ$ B $\succ$ C   4 preferred B $\succ$ C $\succ$ A

4 preferred A $\succ$ C $\succ$ B   3 preferred C $\succ$ B $\succ$ A

Now let us assume that we use the numerical set (3,2,1) to assign points to the various positions within the weak orders. When applied to the above preferences we find the following results.

$$1*(3,2,1)*(A,B,C)=(3A,2B,1C) \quad (1)$$
$$4*(3,2,1)*(B,C,A)=(12B,8C,4A) \quad (2)$$
$$4*(3,2,1)*(A,C,B)=(12A,8C,4B) \quad (3)$$
$$3*(3,2,1)*(C,B,A)=(9C,6B,3A) \quad (4)$$

Summing the points of each option in equations (1) through (4) gives the following totals.

$$3A+4A+12A+3A=22A \quad (5)$$
$$2B+12B+4B+6B=24B \quad (6)$$
$$1C+8C+8C+9C=26C \quad (7)$$

These totals result in the single aggregated order is as follows.

C $\succ$ B $\succ$ A

Thus, according to the Borda Count, criterion C is the most important one. While this is the same as the result from the worst of the worst technique previously described there are several important differences which make this result much more useful.

First, this result was achieved in one iteration. Second, all of the criteria were used throughout all of the calculations, so no information was lost. This is important because it helps to shield the Borda Count from violations of Arrow's Theorem. Third, while the Borda Count violates the IIA axiom [12], since it does so without discarding any information there are no consequences of the violation. This insulation from the consequences of violating the IIA axiom helps to offset the fact that in an engineering design problem the Borda Count also violates the first axiom. Normally, any voting procedure that violates two axioms would not be considered fair, but since the consequences of doing so are minimal in this case; we find that the Borda Count is acceptable and robust.

A fourth advantage of the Borda Count is that it provides a relative ranking of *all* criteria, not just the "best" one. Since the rankings of all criteria are of importance in an engineering design problem the Borda Count is more useful than other techniques.

The Borda Count does have some disadvantages. The first of these is that while the Borda Count is technically a pairwise comparison, there is nothing very "pairwise" about the procedure. The whole point of a pairwise comparison is to break a large problem down into smaller and more manageable parts. The Borda Count requires that each individual decision maker place all of the available options into a weak order before the Borda count is done. With smaller problems this is not an issue, but becomes so as problems increase in complexity.

The other disadvantage of the Borda Count is that while it provides the order of preference of the various options, it does not provide any information on the level of preference between them. In our example, we found that criterion C was deemed more important than criterion B, but *how much* more important is it? This information is not provided, yet it can be of great importance to the concept design process.

The solutions to these problems can be found in the Pairwise Comparison Chart (PCC). The PCC is a graphical form of standard pairwise comparison. The PCC technique has been shown to have results that are identical to the Borda Count [12] and thus shares its robustness. While there are many different types of pairwise comparison charts, the form presented in this paper is based upon the type taught to first year engineering students [13] at Ryerson University by professor Filippo A. Salustri. This form not only leverages the strengths of the Borda Count, but also overcomes many of its usability shortcomings. Its construction and use will be demonstrated though the following example.

Let us imagine that a design team must determine the relative rankings between six different design criteria labelled A, B, C, D, E and F. The empty pairwise comparison chart is first constructed by labelling the leftmost column A through F and doing likewise to the top row of the chart as can be seen in Table 1.

Criteria	A	B	C	D	E	F
A						
B						
C						
D						
E						
F						

Table 1: Empty Pairwise Comparison Chart

The value in each cell represents a comparison of the two criteria standing at that cell. By this logic, only the upper triangle of the matrix has to be completed since the results are mirrored about the diagonal. Furthermore, one need not compare the cells on the diagonal since comparisons of a thing to itself are meaningless. This leaves us with Table 2.

Criteria	A	B	C	D	E	F
A	-					
B	-	-				
C	-	-	-			
D	-	-	-	-		
E	-	-	-	-	-	
F	-	-	-	-	-	-

Table 2: Partially Filled Pairwise Comparison Chart

As designers step through the cells of the matrix and choose amongst each pair, a symbol identifying the winner of each vote (in this case the letter of the winning criterion) is recorded in the corresponding cell. In the case of a tie, both letters are recorded in the appropriate cell. One possible completed matrix for our example is presented in Table 3.

Criteria	A	B	C	D	E	F
A	-	A	A	D	E	F
B	-	-	B	D	B	F
C	-	-	-	D	C/E	F
D	-	-	-	-	E	F
E	-	-	-	-	-	F
F	-	-	-	-	-	-

Table 3: Completed Pairwise Comparison Chart

The weak order is determined simply by counting the number of times each criterion appears in the matrix. For our example, we get the following.

A → 2 Times

B → 2 Times

C → 1 Time

D → 3 Times

E → 3 Times

F → 5 Times

This translates into the following weak order.

$$F \succ (D \sim E) \succ (A \sim B) \succ C$$

Like the Borda Count, this result was achieved in a single iteration without the deletion of any alternatives. This is important because the PCC method, like other pairwise comparison methods, also violates the IIA axiom. However, because there was no deletion of data, and full results were obtained in a single iteration, the PCC method does not suffer for this violation. Furthermore, PCC breaks a ranking problem down into smaller, more manageable cognitive elements for its users, making it more usable than Borda Counts.

In addition, rather than asking designers to generate their own weak orders and then aggregate those results, the PCC method can be done by the entire design team at once. The outcome of the comparison in each cell can be debated by the entire team. This not only helps to open communication within the group, but also ensures that multiple viewpoints are expressed in the results.

Another advantage of the PCC method is that it determines a *value function* between the various alternatives. A value function is a map of the *level* of preference between members of a weak order. While a weak order only shows the order of preference between various alternatives, a value function can show how much one alternative is preferred over another. The precise definition [10] of a value function is as follows.

> **Value Function:** A value function is an assignment of real numbers to alternatives that preserves a weak order of acceptability of those alternatives. A value function maps a set together with a weak order $\{X, \succeq\}$ to the real numbers with its usual ordering {P≥}. For a value function v, $v(A) \geq v(B)$ iff. $A \succeq B$, with equality for indifference.

A value function is typically assigned to a weak order after debate with all stakeholders. However, before entering such a discussion it is useful to have a first approximation of the function [13]. Such an approximation can be built easily from the data in the PCC.

One simple method for generating the initial value function uses the PCC to develop relative weights. Weights are allocated in proportion to the number of occurrences of each criterion in the PCC. Let us return to the previous example in order to better demonstrate this procedure [13]. The equation for the example problem is shown below.

$$5x + 3x + 3x + 2x + 2x + 1x = 100 \qquad (8)$$

This can be reduced down to,

$$16x = 100 \qquad (9)$$

Solving for x obtains,

$$x = \frac{100}{16} = 6.25 \qquad (10)$$

Finally, the value of x can be multiplied by the constant for each criterion to obtain the relative weight of each as shown below.

$$5x = 5(6.25) = 31.25 \rightarrow \text{F=31.25\%} \qquad (11)$$

$$3x = 3(6.25) = 18.75 \rightarrow \text{D=18.75\%} \qquad (12)$$

$$3x = 3(6.25) = 18.75 \rightarrow \text{E=18.75\%} \qquad (13)$$

$$2x = 2(6.25) = 12.5 \rightarrow \text{A=12.5\%} \qquad (14)$$

$$2x = 2(6.25) = 12.5 \rightarrow \text{B=12.5\%} \qquad (15)$$

$$1x = 1(6.25) = 6.25 \rightarrow \text{C=6.25\%} \qquad (16)$$

While these are not necessarily the final relative weights, they can be viewed as the engineering team's assessment of what is important in the design. It may be that the assessments of other stakeholders will differ, but it does create a good basis for further discussion on the subject.

2.3.3 Weighted Decision Matrices in Concept design

While pairwise comparison charts are a very useful tool, they are somewhat limited in the amount of detail that they can provide. They allow decision makers to rank large numbers of alternatives and even to determine the level of preference between items, yet they provide little detail on exactly *why* one alternative was chosen over another. It is important to know that alternative A is superior to alternative B, or that mass is four times more important to a design than cost. Yet it can be even more important to know the reasons for these decisions. Pairwise comparison does not provide this kind of information.

Another limitation of pairwise comparison is that while it is well suited for solving simple problems, the technique is of limited use in more complex situations. Asking an individual whether they prefer a red or the blue car is a far simpler problem than asking that same person whether they prefer the Boeing 777 or the Airbus 320. While the choice of car colour has really only one criterion under consideration, the question of which airplane to select has many more variables such as the number of people the plane carries, maximum range, speed, wing span, and cargo area. The airplane question is a multiple criteria decision making problem.

To further complicate matters, multiple criteria decision-making problems rarely have only a single decision maker. These types of problems typically have many people making the final decision collaboratively; and quite often several groups also have input. These participants will often have conflicting points of view and agendas. Pairwise comparison can only be as good as the intent of its users.

Clearly, solving such a problem requires a far more advanced decision making tool than pairwise comparison alone. Another tool, *a weighted decision matrix (WDM)*, is well suited to this situation. A WDM is a tool that is based upon a table of values [14] and is used to decide amongst alternatives in a multiple criteria decision-making problem.

A key advantage of the WDM is that it ranks alternatives against a predetermined set of criteria rather than against each other.

The problem with ranking alternatives against each other is that there is no constant metric. The alternative design concepts themselves are only vaguely defined at this point, and the design team's own perspective may vary during the process because of external influences (e.g. meeting with experts outside the design team) and more evolved thinking over time. When comparing one alternative against another we are, in fact, comparing two variables [15] and as the variables change the results of those comparisons lose all meaning.

On the other hand, a WDM compares all alternatives against a common and unchanging set of criteria. Typically, the criteria used in the matrix are concrete, measurable properties such as mass, speed, power efficiency and cost. This not only simplifies the ranking process, but also allows for meaningful comparisons between alternatives.

A typical use of weighted decision matrices in concept design is for selecting the most suitable concept from amongst the available alternatives. When using WDMs in such a fashion, one selects a "reference design" as a baseline and compares all new concepts to it. The reference design may be an existent similar product, or it may be symbolic of the kinds of technology the new product is supposed to replace. For instance, the reference design for the original Palm Pilot PDA was a small leather bound agenda. The criteria used in a design decision matrix are crafted from the requirements for the product.

The construction of a weighted decision matrix is best illustrated by an example.

Assume a design team has been asked to design a new type of soil tilling device for home garden use. The team has identified three concepts; a modified hoe design, a rotating mixer, and an electric powered autotiller. Also, the product requirements lead the team to set the following criteria; mass, speed, operating comfort, ease of operation, reliability, and cost.

A WDM for this problem is shown in Table 4. Design characteristics, which are representations of requirements, are arranged in rows. Each characteristic is given a weight to represent its importance relative to the other characteristics. Design concepts occupy the columns. Each "cell" of the table contains a rank, which is assigned by the design team, and a score, which is a mathematical combination of the rank and the corresponding weight. Pairwise comparison is an ideal tool for calculating the weights in a decision matrix. Finally, the last row of the matrix contains arithmetic sums of the scores for each concept. The concept with the highest total score is the concept most preferred by the team.

		Concept					
		A		B		C	
		Modified Hoe		*Rotating Mixer*		*Electric Autotiller*	
Design Characteristic	**Relative Weight**	**Rank**	**Score**	**Rank**	**Score**	**Rank**	**Score**
Mass	0.2						
Speed	0.1						
Operating Comfort	0.25						
Ease of Operation	0.15						
Reliability	0.15						
Cost	0.15						
Total Score							

Table 4: Prepared Decision Matrix

The key activity in completing a weighted decision matrix is assigning a rank for each concept and characteristic. Numerical ranking scales are used to do the rankings. Different scales have been used by different industries and companies. Scales are generally selected based on experience; there has been no research to evaluate the different scales scientifically. In the absence of particular experience, the simplest and most obvious scale to use is a linear symmetric scale like the one in Table 5.

Rating	Meaning
2	Greatly superior with respect to characteristic
1	Somewhat superior with respect to characteristic
0	Satisfactory with respect to characteristic
-1	Somewhat inferior with respect to characteristic
-2	Greatly inferior with respect to characteristic

Table 5: Automotive Industry Linear Scale

It is considered good practice to assign ranks to concepts in a team setting, such that consensus is reached for each rank. This helps ensure that different points of view are all made explicit from the outset. These discussions usually depend heavily on requirements specifications. The use of the baseline concept can also be very useful at this stage. To use this technique the design team can either insert the baseline concept into the matrix, or simply keep it in mind as a point of reference. The baseline concept is then assigned a score of zero in all design characteristics. The other concepts are then rated against this concept. If they are better than the baseline they are assigned a positive score, and if they are inferior they are assigned a negative. For the example problem, we could use a regular garden shovel as the baseline, and then rate how the other three concepts improve on this design. An example of possible rankings for the sample problem is presented in Table 6 below.

		Concept					
		A		B		C	
		Modified Hoe		*Rotating Mixer*		*Electric Autotiller*	
Design Characteristic	**Relative Weight**	**Rank**	**Score**	**Rank**	**Score**	**Rank**	**Score**
Mass	0.2	2		1		0	
Speed	0.1	-2		2		2	
Operating Comfort	0.25	0		1		2	
Ease of Operation	0.15	1		1		2	
Reliability	0.15	2		1		0	
Cost	0.15	2		1		-1	
Total Score							

Table 6: Decision Matrix with Concept Rankings

Once all the rankings are assigned, an *aggregation function* is used to derive a corresponding score. In its most basic form, an aggregation function is a mathematical function that combines the rankings for each concept with the corresponding weights in some manner. There are many possible aggregation functions. Good aggregation functions for engineering design should adhere to the seven following properties [14].

1.) **Dominance:** The alternative with the highest aggregate score is the dominant alternative, assuming all relevant issues have been included in the aggregation.

2.) **Boundary Conditions:** If a concept scores the highest/lowest against all characteristics than the designers will prefer it most/least.

3.) **Continuity:** As an individual characteristic's preference (score) is changed slightly, then the overall preference for the design will change at most slightly. This does not mean that the preference for any characteristic must be continuous.

4.) **Monotonicity:** If an individual characteristic's preference is raised or lowered, then the alternative's overall preference is raised or lowered in the same direction, if it changes at all.

5.) **Commutativity:** The aggregate score must not change if the characteristics are reordered without changing preference ratings or weights.

6.) **Idempotency:** If a designer has the same preference for all individual characteristics in a design, then the aggregate score must have this degree of preference as well.

7.) **Annihilation:** An alternative is unacceptable if at least one of its characteristic preference ratings is less than or equal to zero.

Given these properties, there are two aggregation functions that are commonly used in engineering design.

The first of these is known as Simple Additive Weighting (SAW) and has the form below.

$$score = \sum_{i=1}^{n} w_i R_i \qquad (17)$$

In Equation 17, w_i is the weight of the ith criterion and R_i is the rating given to that criterion. The greatest advantage of SAW is simplicity. The user simply steps through the cells of the decision matrix and multiplies each rating by its corresponding weight. The aggregate score is the sum of those products.

The weakness of SAW is that it does not satisfy the seventh property for engineering design aggregation functions: annihilation. This means that SAW does not automatically remove from contention concepts that do not meet one or more of the criteria.

Another aggregation function that can be used is *Multiplicative Exponential Weighting* (MEW) shown in Equation 18 [16].

$$score = \prod_i R_i^{w_i} \qquad (18)$$

The main advantage of MEW is that it satisfies all seven properties of engineering aggregation functions.

Unlike SAW, MEW removes concepts that fail to meet a criterion from contention. However, the price for this automatic removal of concepts is increased computational complexity. Furthermore, in his paper "Multi-Attribute Decision Making: A Simulation Comparison of Select Methods" Zanakis demonstrates [16] that apart from the ability to screen out alternatives, MEW produces very similar results to those obtained using SAW. Therefore, MEW requires more work to get similar results, clearly an unappealing situation. By simply examining the results generated by SAW in order to discount any failing concepts, the advantages of MEW are minimized while maintaining the simplicity offered by SAW.

The fact that SAW does not meet the annihilation property also increases the number of potential rating scales that it can use. If, for example, one were to use the numerical scale presented above with MEW, then many concepts would be removed from contention inadvertently. MEW assumes that a concept that is assigned a score of zero in any single design characteristic is a failure. However, the previous scale implies that a zero rating is satisfactory with respect to that characteristic. MEW would eliminate this alternative needlessly. In addition, the MEW function would not allow for the use baseline concepts in the decision matrix since they have zeros for all design characteristics.

The completed example WDM, using SAW, is shown in Table 7. In order to reduce the size of the table for inclusion in this report, the reference concept is not shown. It was merely kept in mind during the scoring process, and the other three concepts were ranked against it.

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

Table 7: Completed Decision Matrix

Examination of the completed matrix reveals not only the "winning" design, but also several other features of the matrix that are of great use to the designer.

The WDM not only identifies the most suitable concept, but also contains ample evidence as to why this is the case. Inspection of the matrix revels that both the modified hoe and the electric autotiller excel in certain areas, but also have significant drawbacks. On the other hand, while the rotating mixer only really excelled in the speed characteristic, it also did not suffer the weaknesses of the other designs; instead it took a more balanced approach to the design problem.

Because all of the potential concepts are visible at once, it becomes quite simple to identify quickly the strengths and weaknesses of each design. Not only does this allow for identification of areas of improvement in each concept, but can also be of great benefit in the creation of *combined concepts.*

Combined concepts are designs that attempt to blend the best attributes of two or more concepts into a single, superior concept. For example, in Table 7 it is clear that the electric tiller outperforms the other two designs in operating comfort and ease of operation, while mass and cost are its main weaknesses. If the weak areas could be improved, then it is the electric autotiller could become the most suitable concept, Thus not only has the WDM helped to solve a difficult multiple criteria decision making problem, but it can also provide evidence to support that decision and may even stimulate further, more creative design.

The techniques introduced in this section are able to tackle a wide variety of engineering design problems. The main difficulty with so many different options available is that designers can become confused when trying to select the most suitable tool for a given situation. The next section will introduce the IDEA process, which has been designed to solve this problem.

3.0 The IDEA Process

3.1 Motivation

Section 2 described the most popular tools for concept design. Selecting the best tools for specific instances is problematic. Even once tools are selected, ensuring that they are properly integrated into an organization can be difficult and expensive. One goal of developing IDEA is to help practitioners overcome this difficulty by addressing these problems.

The initial inspiration for this project came from the author's experiences at MDA Space Missions in Brampton, Ontario. During undergraduate studies at Ryerson University, a thesis and team design project were executed with assistance with engineers from the company. During that time, the author noticed several features of the concept design process that raised concerns about how concept design was practiced. The concept design practices in place at MDA at the time did an excellent job of outlining the Trade Space for a design problem as well as generating concepts from that Trade Space. When it came time to actually select a concept, however, there did not seem to be any structured system in place. The "winning" concept seemed to be selected arbitrarily which this author found disturbing. Further research into concept design practices in the engineering industry revealed many of the weaknesses outlined in Section 1.

While current best practices are obviously satisfactory – the many existent successful products prove this – the design activity itself is subject to iterative improvements.

One problem with current concept design techniques is a lack of standardization. As noted in Section 1, most companies follow very similar engineering design processes. However, details of workflow within the process at different companies will vary greatly. This is especially true in concept design.

This observation raised the question of whether or not it would be possible to create a systematic process for concept design. The main advantage of a systematic process is that it helps alleviate uncertainty by selecting techniques that research has established as superior, and integrating them into a usable, effective process. This allows the designer to focus on the design itself and not the process used to find it, which helps to improve creativity and final results.

Another shortcoming of existent concept design tools is that many of them are incomplete with respect to the goals of concept design processes given in Section 1.2. That is, one or more of the goals remain unfulfilled by existent tools.

In one such case a company's concept design process was very good at determining customer needs and then exploring the resulting design space, but when it came time to actually select a concept there was no real procedure in place. Instead, computerized simulations were developed in an attempt to understand the performance of products that were still at a very early phase of design. This is both time-consuming and unreliable since these simulations were based upon very preliminary data which made their results suspect. Basing design decisions on the results of these simulations not only casts doubt onto these decisions, but also provides little meaningful supporting evidence.

Another common occurrence is that the actual design of a product is undertaken in detail, yet little time is spent analyzing whether or not customers will actually want the product. This can be seen in the many cases of consumer products that seemed to be a good idea and were well implemented; yet they did not succeed in the marketplace.

A good example of this would be the high resolution audio formats, DVD-Audio and Super Audio CD (SACD). Both were released in late 2000 and from a technical perspective either one of the formats should have been a resounding success. Both offered *vastly* improved sound quality over the standard audio CD and offered high-resolution 5.1 channel surround sound at a time when surround was becoming increasingly popular. And in the case of SACDs, they were even compatible with existing

CD players. Yet even with all of these features, up until now both formats have been what can only be called a dismal failure; the question is why?

The answer is that neither format addresses what most customers actually want. Both formats were released based upon the assumption the people would embrace them like they had audio CDs. What designers failed to recognize however, was that CDs were becoming increasingly unpopular. The new distribution method of choice was compressed audio formats like MP3 and WMA. Whether purchased legitimately or more commonly, downloaded illegally from the internet, people were shifting away from physical media to files that they could simply download. It did not matter that MP3s sounded worse than even the lowly audio CD, people no longer wanted to pay for music when they could get a "good-enough" version for free with just a few clicks of a mouse. The increasing popularity of iPods and other MP3 players is evidence of this.

This scenario may manifest itself again with the release of HD DVD and Blu-ray discs in 2006. Both offer drastically improved performance over DVD yet the question remains whether the new formats will succeed. No one is really sure, but we may be seeing yet another example of the right product at the wrong time.

Given these examples, the objectives for the IDEA process are quite clear.

Firstly, IDEA should adhere to the three goals of concept design given in Section 1.2. This will allow for a more thorough understanding of the problem and customer needs, and will increase the probability of a better design through a more thorough exploration of the design space.

Secondly, the IDEA process should follow a series of standard, but flexible, steps, to increase designer comfort with the process. Since the same steps will be used with any design, engineers can spend more time and energy designing and less time worrying about what tools to use. This not only helps to increase creativity, but also reduces concept design time and costs.

Thirdly, the IDEA process should use well-established concept design techniques wherever possible. The use of proven and well understood techniques will not only increase the robustness of the overall process, but will also assist with designer comfort. Using tools that designers are already familiar with will reduce the time spent learning the new process and increase the time spent actually using IDEA. By achieving these three goals, IDEA should help designers achieve greater success in concept design by reducing time, costs, and uncertainty.

3.2 Integrated Design Exploration and Analysis: An Overview

IDEA is a systematic process for concept design that was created to solve the problems with current concept design processes as described above. IDEA is designed to lead the user through the concept design process, from the determination of customer requirements, to the final step of selecting the most promising concept to meet those requirements. An overview of the process is shown in Figure 6.

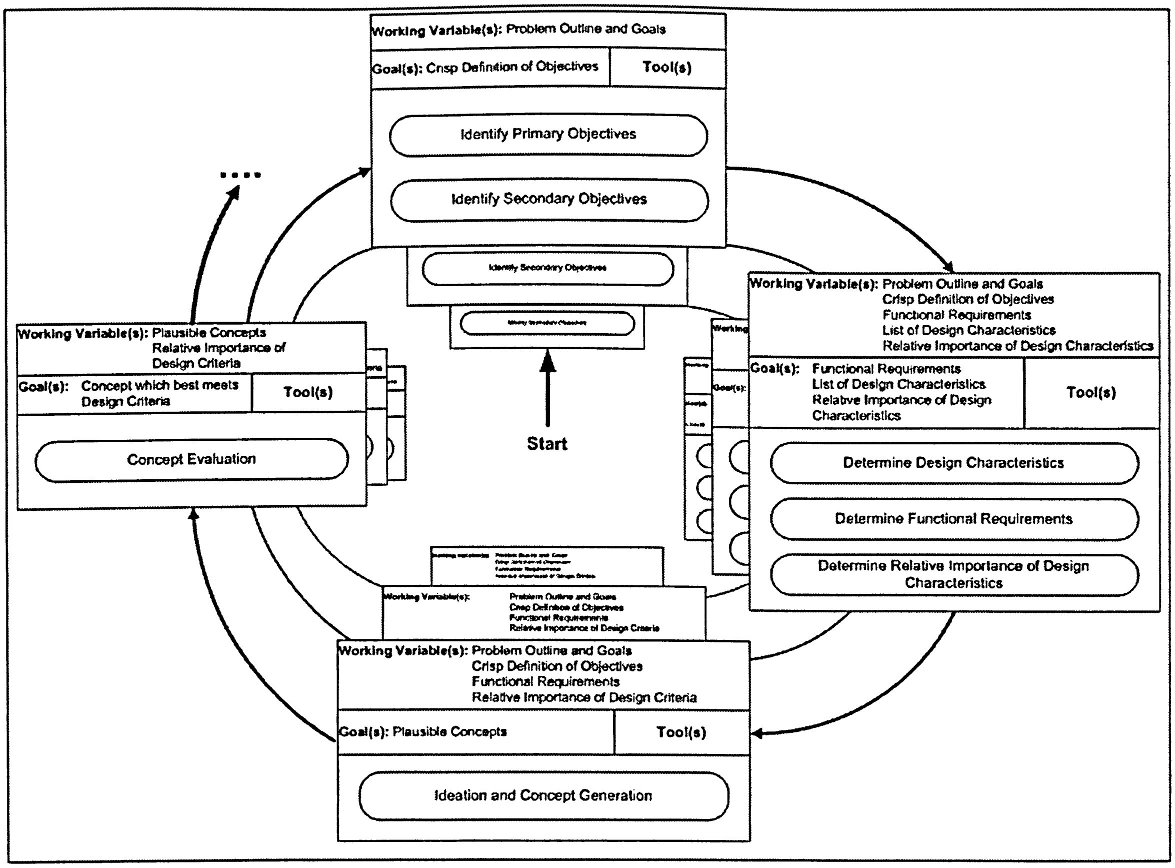


Figure 6: IDEA Overall Outline

As can be seen in Figure 6, IDEA is divided into four main modules, arranged in a spiral pattern. The user begins the process at the first module known as *Identify* and then moves through the remaining three modules in the sequence; *Determine*, *Explore*, and finally *Analyze*.

IDEA is arranged as a series of levels and each level is a repetition of the previous loop. Concept design is rarely accomplished in a single step; quite often, many iterations of the concept design process are necessary before a suitable concept can be found. Each time around the loop, the design team gains a better understating of the problem and what solutions are possible. The multiple levels and increasing font clarity in the diagram represent this increasing understanding.

While the design team must complete all four modules on the first iteration, subsequent loops may omit certain modules, although at least cursory glance at all modules is recommended in order to identify any problems that may have been missed in previous loops. The number of iterations required depends on the design problem.

Each module is divided into a number of sections. The first of these sections is called *working variables* and details information available at that stage of the concept design process – the "input" to that module. As can be seen in the diagram the amount of available information grows very quickly early on and then seems to undergo a sudden drop in the later modules; however this is not actually the case.

The early modules are concerned with gaining a better understanding of the problem itself. This causes the level of available information to grow very quickly as the design team better understands the needs and desires of the customer and what the design must do to meet those needs.

The latter two modules deal with using this information to generate possible solutions. The overall direction for concept generation is guided by the customer needs determined in the earlier modules. Any concepts which result are nothing more than a representation of these needs. Therefore, no information has been lost, it has merely been incorporated into the concept which will be selected.

The second section of each module outlines the overall goals for that module. The third section at the bottom of each box outlines the tools for that module. At high levels of detail this area contains the tools that can be used to achieve the goals presented in the second section; however, at this global level, it only outlines what has to be done to achieve the goals. Because the information used in a module is generated in those that preceded it, the modules must be executed in their designated order. On the other hand, the items in the tools section are typically not in any specified order and may be completed in a manner that makes sense for the actual project.

The details of the four modules and their sections are provided in the following sections. An expanded view of the diagram in Figure 6 can be seen in Figures 7 and 8.

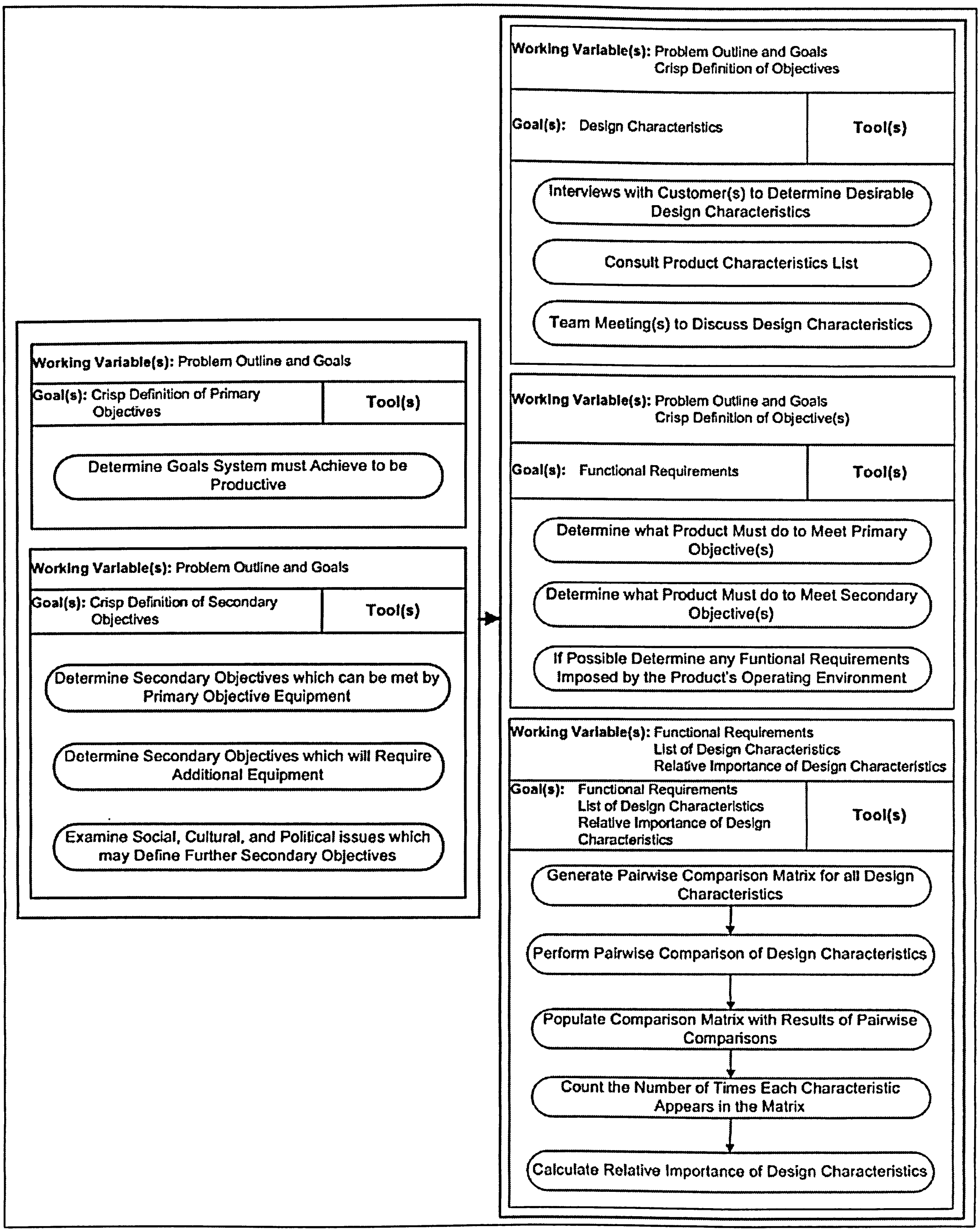


Figure 7: Expansion of Identify and Determine Modules

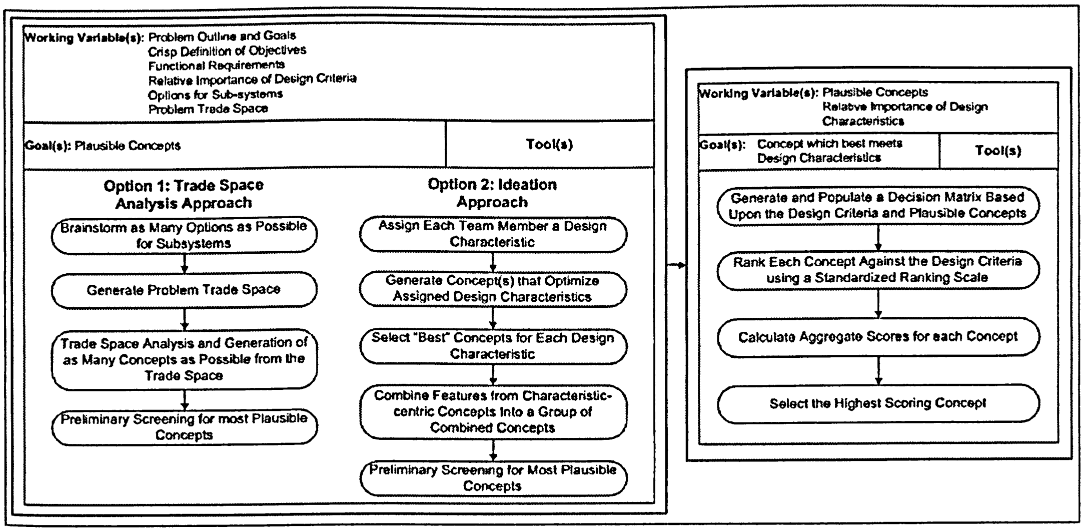


Figure 8: Expansion of Explore and Analyze Modules

3.3 The *Identify* Module

The first module is the *Identify* module. The purpose of this module is to help designers gain a better understanding of exactly what customers expect from the product. The expansion is presented in Figure 9.

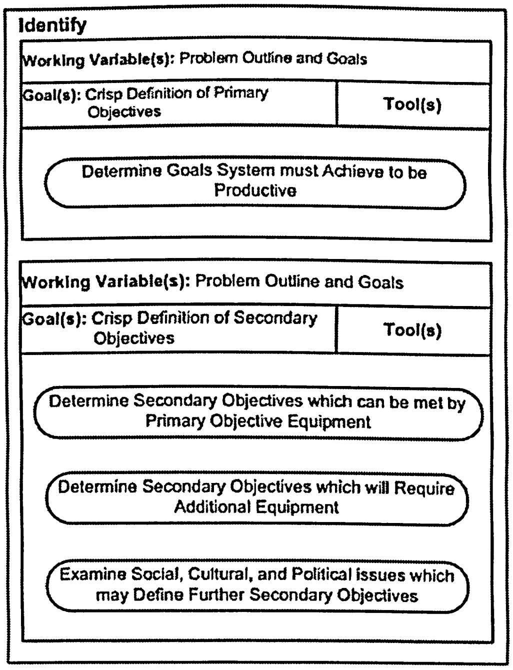


Figure 9: Identify Module Expansion

At this initial stage of concept design there is very little information available other than a preliminary statement of the problem outline and goals. While this provides managers and other stakeholders with an overall sense of about the project, it provides little techniqal information to the design team. To gain a more detailed understanding of the problem, engineers must identify two different sets of information: the *primary* and *secondary* objectives.

As can be seen in Figure 9, the primary objectives are defined as the goals that the system *must* achieve. Basically, they are what the final design must do at a minimum to be considered a success. Any concept that cannot meet these objectives is immediately removed from further consideration.

Due to the importance of the primary objectives, the design team must take extra care when defining them. Items included needlessly can make the final design too expensive and difficult to complete. It is therefore advisable to meet with all stakeholders to gain their perspective. Ideally, the Problem Outline and Goals document would not include any extraneous requirements; however, this is rarely the case. Quite often, it is drafted by individuals who are not technically literate. They see that a competing product can perform a certain function and decide that their product must do the same thing, even if those expectations are neither realistic nor required. It is the job of the engineering team to discuss the objectives of the project with all stakeholders in order to determine if all objectives are truly required. Once these issues have been clarified, the team can then brainstorm amongst themselves to generate crisp definitions of the primary objectives.

The secondary objectives add further value to the product. Where the primary objectives are the needs of the customer, the secondary objectives are their *desires*. For example, imagine that a major airline has stated that they need a new airplane that can fly from Toronto to London in seven hours, and that it would be greatly beneficial to the airline if the trip could be made in as little as six.

The primary objective here is to design an airplane that can fly from Toronto to London in a maximum of seven hours. This is what the airline *needs* in order to consider purchasing the new product. A flight time of six hours is a secondary objective in this case, because it is desirable but not required.

As with the primary objectives, the design team must take special care over what secondary objectives are included in the final design. While secondary objectives add value, they must never interfere with the successful implementation of the primary objectives. The inclusion of too many secondary objectives can lead to a phenomenon known as "feature creep" where features are added to a product until it becomes either difficult to manufacture or prohibitively expensive. Feature creep can also increase design time as concepts are constantly revised to include just one more feature. At some point, the design team must stop conceptualizing and start selecting.

Secondary objectives can be broken down into three types; (1) those that can be accomplished using equipment designated for the primary objectives, (2) those that require additional equipment, and (3) those that are externally imposed.

The airplane example noted above demonstrates a type 1 secondary objective. The secondary objective (a six-hour flight time) can be achieved through a more efficient airframe, engines, etc, which are all required to meet the primary objective.

An example of a secondary objective of type 2 might be in-flight entertainment. This would require extra equipment beyond anything required for the primary objectives.

Type 2 secondary objectives can be problematic since they typically impact the performance of primary objectives in some way. In-flight entertainment systems increase aircraft weight by some small amount. While this may only have a slightly negative impact on the performance of the aircraft, the inclusion of too many such features could have a dramatic effect on the overall weight and performance.

Type 3 secondary objectives are those that are imposed by external (usually societal and political) sources. Although many engineers rarely think of them, objectives of this type can have significant effect on the final design. For example, the design of factories has become far more difficult since the ratification of the Kyoto Accords, which specify strict pollution controls. While these pollution controls are not required to complete the primary objectives of a factory, they are nonetheless *required*. The inclusion of pollution controls not only adds to the cost of a plant, but they can also have a serious impact on its performance. Normally, a secondary objective that seriously impacts the performance of a primary objective would be discounted immediately; however, because of government regulations the pollution controls must remain. While this is an extreme example of the kind of impacts that political and societal issues can have on a design, it illustrates that designers should be cognizant of these issues and prepared to overcome them.

3.4 The *Determine* Module

The second IDEA module is the Determine module and is shown in Figure 10.

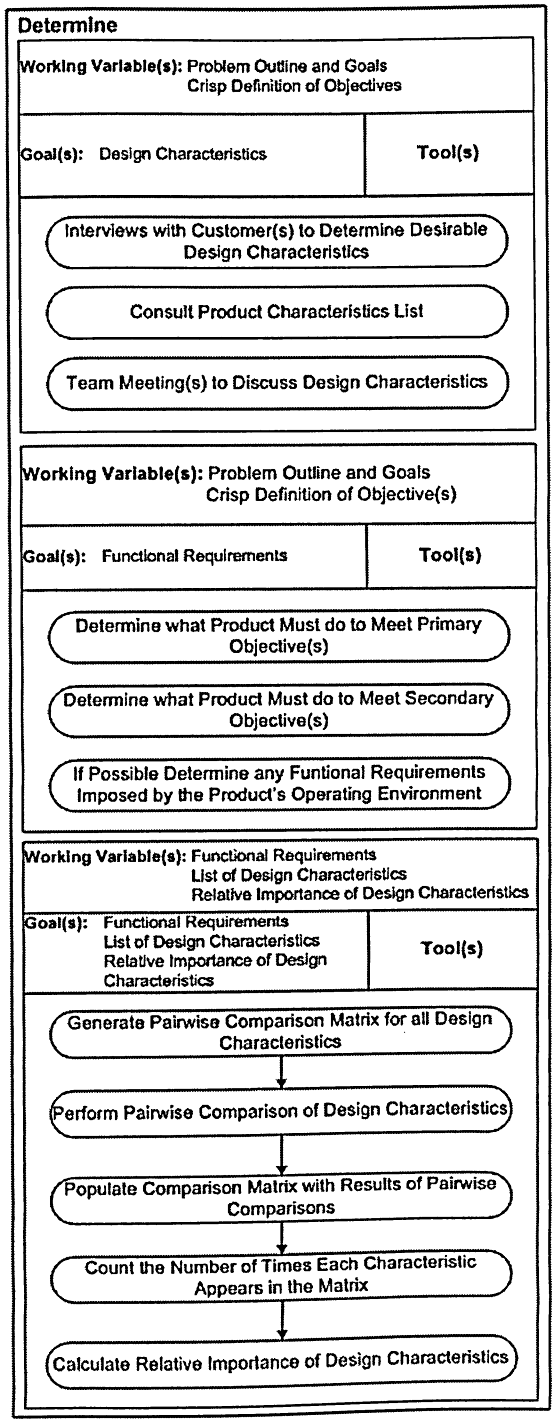


Figure 10: Determine Module Expansion

The *Determine* module is used to gain a better idea of what form the solution should take. Because of the increased level of information available at this stage, there are more tools available to the designer. Figure 10 reveals that the *Determine* module is divided into three main tasks: determination of *design characteristics*, determining *functional requirements*, and determining the *relative importance* of the design characteristics. Completing these three tasks will help the design team gain a better technical understanding of what the final product should do and what characteristics it should have.

The first task is to determine the design characteristics for the final product. Design characteristics serve a number of roles in concept design.

First, design characteristics describe what the final product has to *be*, such as "the product must be lightweight" or "the product must be safe". They define the physical and usability attributes that the product must have. It is important that the design characteristics make no mention of *how* the product will accomplish these goals since at this stage in IDEA, designers have very little hard information on which to base solutions. This work defines the boundaries of constrained trade space for the product; placing the final design on a specific functional path at this point may unnecessarily limit the variety of possible solutions.

Design characteristics may seem obvious, but they can have an extensive effect on the final design of the product [17]. For example, stating that the final product must be lightweight can influence everything from the configuration of sub-systems to the selection of materials for the product. They serve as succinct reminders to the design team of what really matters in a design.

A second role of design characteristics is to serve as criteria by which the effectiveness of concepts is assessed. Design characteristics can become drivers for the final design and assessing how well a particular concept is able to meet these characteristics can serve as a good indication of its success. Because the design characteristics for a particular design problem are constant among all concepts, they are also well suited for use as a standardized benchmark when selecting concepts. By assessing each concept against a reference design with respect to the design characteristics, it is possible to conduct meaningful comparisons between design options to determine which is the most promising.

There are several different techniques for determining the design characteristics and a combination of these methods will most likely provide the best results. One such technique, interviews with customers, was explained in Section 2.1. It is particularly useful here for several reasons.

Firstly, if a product is intended for public consumption then it is important to understand what characteristics would make that product desirable to the general public. In many cases, the average consumer does not really care what functions a product has internally, rather they are more interested with its usability and aesthetics. The iPod is an excellent example of this. Compared to many other MP3 players, the iPod has a number of deficiencies (inferior sound quality, a draconian music distribution model, a lack of additional features such as radio tuners and support for other music formats) [28]. Yet, even with these deficiencies the iPod significantly outsells its competition. One of the prime reasons for this is marketing; the iPod is an *extremely* well marketed device; many people call all MP3 players iPods. Other factors of the iPod's success are its simplicity, small size, sleek form, and wide assortment of available colours. All these factors are best represented by design characteristics.

Another advantage of determining design characteristics through interviews is that many people have limited technical knowledge. Design characteristics deal with such generalities, making them the ideal type of information to gather from general consumers.

A second method of generating design characteristics is to use a standardized list. While design characteristics may change from problem to problem, there is also a great deal of commonality. This commonality has led to some standardization. Clearly, this is of great advantage to designers since the list can be used as a starting point to generate the design characteristics for a particular problem.

Finally, while customer interviews and standardized lists are good starting points, they likely miss some important but particular characteristics that only apply to the current design problem. The generation of these final few characteristics is often best accomplished through brainstorming sessions and face-to-face meetings.

The second major task in the *Determine* module is establishing the overall functional requirements of the design. Whereas the design characteristics are designed to map out what a product must *be*, the functional requirements are meant to define exactly what the final design must *do*. Figure 10 reveals that there are three ways to generate functional requirements within IDEA.

The first of these involves thinking about what the product must do operationally to meet the primary objectives. The primary objective in the airline example is a seven hour flight time. This statement implies a number of functional requirements: the airplane must fly under its own power, sustain that performance for at least seven hours, and achieve sufficient ground speed to reach London from Toronto in that time. It must also safely lift off and land. While this list is not exhaustive, it does show how a single primary objective can define many different functional requirements.

Because these functions were defined by the primary objectives they are denoted as *primary functions*; things that the final product *must* do. A similar analysis of secondary objectives leads to secondary functions.

The third class of functional requirements arise from the environment in which the product must operate. For the airplane example, these include but are not limited to maintaining internal pressure, withstanding ground and in-flight weather, and maintaining internal and ground communications. The use of IDEA helps ensure that these items are not forgotten by the design team. A useful method for determining the three different types of functional requirements is brainstorming sessions between the entire design team.

The final task in the *Determine* module is establishing the relative importance of the design characteristics. In an ideal situation, all design characteristics would be equally important to the final design; however, this is not the case in practice. In reality, designers are often forced to trade off the performance of one part of a design for another. By determining the relative importance of the design characteristics, designers are better able to understand which tradeoffs are feasible and which are not.

IDEA uses a *Pairwise Comparison Chart (PCC)* to determine the relative importance of the design criteria. The PCC was selected for several reasons. Ranking amongst the design characteristics is an example of a single criteria decision-making problem, a situation for which the PCC is well suited. PCCs also render the selection process insensitive to violations of Arrow's IIA axiom, and are able to generate the relative weights in a single iteration. PCCs also promote participation of the all stakeholders during the critical process of assigning the relative weights.

The inclusion of all stakeholders also allows for further discussion on the relative weights after pairwise comparison is complete. As stated earlier, the relative weights generated by the comparison chart should only be used as a first estimate. Further discussion amongst all stakeholders can dramatically shift these results in order to satisfy all parties involved.

Upon completion of the *Determine* module, the design team should have a better understanding of the problem and know in crisp but overall terms how proposed solutions can be assessed.

3.5 The *Explore* Module

The purpose of the *Explore* module is to help the designers generate the trade space for the design problem, and to explore that space to find a superior solution. Unlike the previous modules, the *Explore* module offers two different approaches for this. The first is a typical trade space analysis, while the second is a method known as *ideation* that attempts to find the best overall solution by combining partial solutions. The details of these options are presented below. An overview of the module is given in Figure 11.

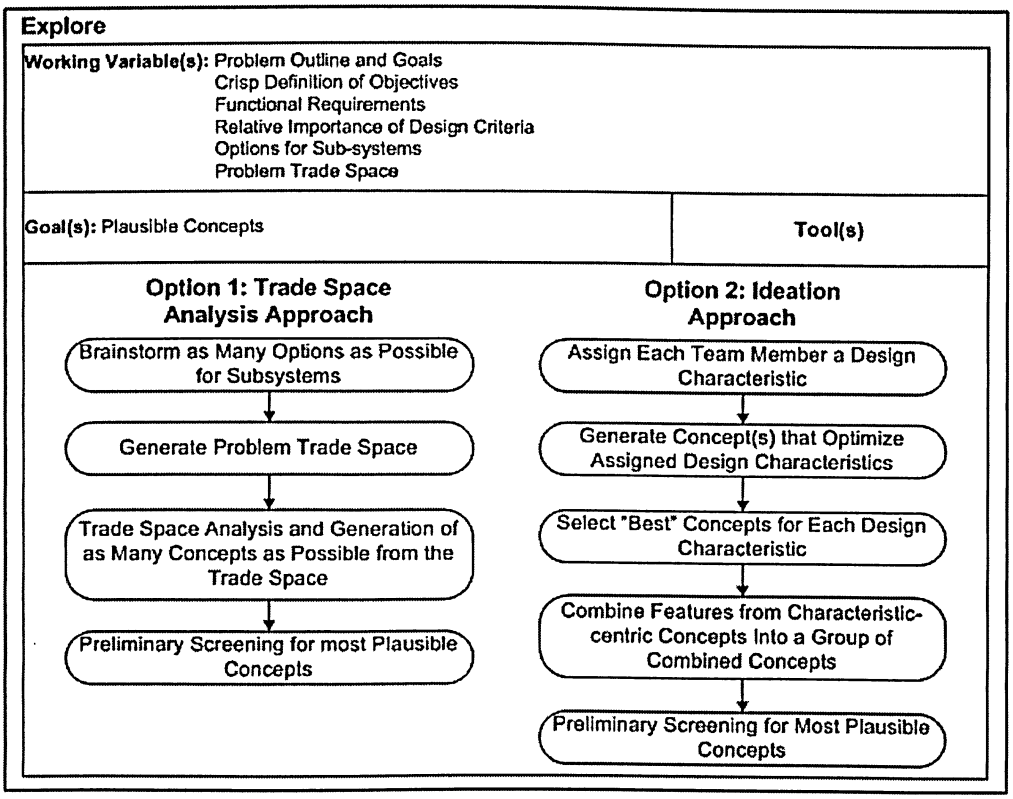


Figure 11: Explore Module Expansion

The typical trade space analysis is as outlined in section 2.2.2. First, the design team generates as many options as possible on a subsystem-by-subsystem basis. By focusing on individual subsystems rather than an entire concept, designers are forced to break the problem down into smaller and more easily managed pieces. While this can prevent certain overall solutions from being identified immediately, it is very good at controlling the inherent complexity of the task. By preventing the designer from attempting to look at the entire problem at once IDEA allows them to focus their creativity on a single aspect of the design in an attempt to optimize it as far as possible.

Once all of the options have been generated, the next step is to create a trade space diagram to outline all of the possible alternatives. Although there are many different types of diagrams, it is suggested by the author that designers utilize the form outlined earlier in this report since it is simple to create and displays large amounts of information in a very clear manner.

Once the trade space diagram is laid out, the design team can begin to generate concepts. While an engineer's first impulse when presented with a design problem is to start imagining concepts, up until this point the IDEA process has attempted to inhibit that impulse. The danger in immediately generating concepts is that any concepts created at such an early stage may have deficiencies due to a lack of knowledge about the design problem. Yet, it is these concepts that will stay with designers and bias their thinking, limiting their creativity later on. IDEA attempts to subvert this potentially dangerous bias. By utilizing the trade space diagram, designers can test different sub-system combinations as they use their creativity to explore the trade space. It is important for the design team to create as many different concepts as possible in order to fully explore the trade space. Many of these concepts will be unsuitable, but by generating a large number of them, the possibility of finding a truly novel one is increased.

Once the design team has finished generating concepts, the final step in trade space analysis is to reduce what is most likely a large list down to the five to ten most promising candidates. Because this is merely a preliminary pass at screening the concepts, structured methods are not necessary. A thorough brainstorming session and discussion between team members should be more than adequate to reduce the list of candidates. These candidates will then be subjected to a far more thorough screening process in the final module.

The second option available in the *Explore* module is an ideation-based approach. As implemented within IDEA, this technique is very simple. The first step is to assign each individual in the design team a single design characteristic. Each team member then generates concepts that are optimized only for that specific characteristic. Generating entire concepts at once is normally quite difficult while focussing on a single characteristic is much simpler. Much of the difficulty that normally occurs when generating entire concepts stems from the fact that "good" concepts typically balance many different design characteristics. It is the balancing of disparate goals that causes difficulty. By focusing on a single design characteristic this balancing acts is avoided which makes generating concepts quite easy. These partial concepts are usually called *ideas* – hence, the name of the technique. Ideas are best generated as an individual exercise of each team member, to limit the natural influences of the thinking of other designers.

While generating ideas is easier, they will be only partial solutions. To develop full concepts, ideas must be combined. This is relatively easy to do: one simply selects a set of ideas such that all the design characteristics in the requirements specification are covered. The number of concepts that can be generated this way is very large. For example, if three ideas are developed for each of three design characteristics, then one can generate 27 different concepts; for five ideas developed for each of five characteristics, there are a total of 3125 concepts possible. Blending ideas together into full concepts is an activity best done collaboratively because at this point the influences of all the participants will tend to lead solutions that are more integrated.

Obviously, many of the concepts generated this way can be discounted out of hand for one reason or another, depending on the nature of the design problem. A commonly used technique to "prune the search space" is to identify combinations of ideas that can be discounted immediately before starting to combine ideas into concepts. This is usually done "by inspection" as the concepts are generated. It is important to do this as a team exercise to take advantage of everyone's combined experience.

For example, a concept may be identified as combining materials in a way that is physically impossible or not feasible (e.g. welding aluminum to steel). Of note is that eliminating one pairing of ideas also eliminates all concepts containing that pairing. This can very quickly prune the search space.

Ideally the design team should now have a fairly long list of possible combined concepts. The final task in ideation is to analyze these concepts and select the five to ten most plausible candidates. If the team is unsure about a particular concept it should be included since it will undergo detailed analysis later. Brainstorming is a particularly good technique to be used for this task since it encourages discussion between all team members, is relatively lightweight and quick, and preserves information on why certain concepts were rejected.

3.6 The *Analyze* Module

The fourth and final module in IDEA is the *Analyze* module. The details of the module are shown in Figure 12.

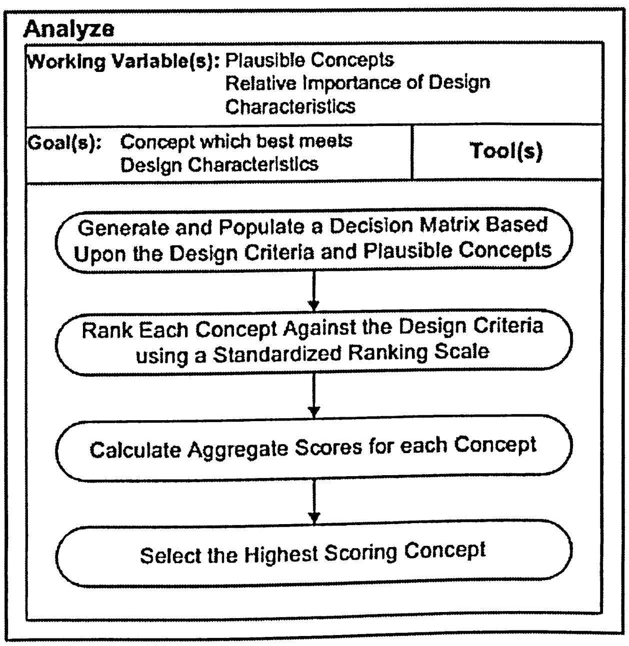


Figure 12: Analyze Module Expansion

The Analyze module helps designers determine which of the concepts that resulted from the Explore module will selected for detailed design work. A weighted decision matrix (WDM) was selected for this task for a number of reasons.

First, the WDM (as discussed in Section 2.3.3) is an ideal tool to use in cases of multi-criteria decision-making problems with multiple participants; concept evaluation and selection is one such kind of problem.

Second, a WDM helps assure traceability in the decision making process. Selecting the most suitable concept is the most important design decision, but regardless of the concept selected, there could be unanticipated problems during detailed design. In such cases, a WDM can provide engineers with a record of the decision making process, to remind them why those decisions were made.

Third, a WDM encourages the participation of the entire design team. This allows multiple points of view to be expressed in the final selection and increases the probability that that final selection is the best possible one. Additionally, including the entire team in the decision making process ensures that all members are on "the same page" and helps to foster communication within the group.

Finally, a WDM is lightweight and simple, keeping the "overhead" of its use low. IDEA also utilizes SAW to generate the aggregate scores for each concept. This simplifies the computations, while still producing results comparable with more intensive methods.

The first step in the Analyze module is to build an empty WDM. This is done using the techniques described in Section 2.3.3. The weights of the design characteristics, calculated in the Determine module, are simply inserted into the matrix.

Once the WDM has been prepared, each concept is rated. The WDM as implemented in IDEA uses the -2 to 2 ranking scale that is commonly found in the automotive industry. The use of this scale in combination with the SAW aggregation function means that the IDEA WDM does not satisfy the annihilation property and thus does not automatically discount concepts that fail in one or more design characteristics. The design team must practice due diligence in ensuring that concepts which are clearly unsuitable are eliminated.

Ranking the concepts is best done in a group setting where each individual is free to voice their opinion on how well each concept meets each characteristic. Again, this ensures that all points of view are respected, and helps to make sure that the decisions made are a representation of the preferences of the team as a whole.

Once the concept ranking is completed, SAW is used to calculate the aggregate scores. In SAW, each standardized ranking is multiplied by the relative weight of the corresponding design characteristic. Once all of these multiplications have been completed, the results are simply summed to generate the aggregate scores.

Examination of the aggregate scores will most likely reveal a few concepts that have significantly higher scores than the others. If there is only one high scoring concept the final concept selection is trivial; but if there are two or more, then the selection becomes far more difficult. In this case, the final selection can be made by simply picking one using experience and intuition, or a second iteration of the IDEA process can be conducted in an attempt to reach a decision. If the two concepts are very close in score, a third possibility is to advance both to more detailed design. While this may be time consuming, detailed design will more than likely reveal strengths and weaknesses that will let a final single selection be made. The specific route taken by a particular team will depend on a number of external factors (scheduling, milestones and gates, business processes, etc), so IDEA is intentionally generic on this point.

One final item of note is that design teams should look for opportunities to combine concepts. One of the great strengths of WDMs is that they display the strengths and weaknesses of all concepts simultaneously. While low scoring concepts may seem to be of no real value, designers should look at the WDM closely to determine if any poorly scoring concepts can be combined. The combination of two poorly scoring alternatives may address their respective weaknesses and create a much better concept. The WDM does not have to be solely a decision making tool, it can also be used to spark further creativity and innovation.

4.0 The IDEA Software Interface

This section will discuss the software interface that was created for IDEA. The actual mechanics of using the interface will not be covered here; this can be found in Appendix A. Instead, this section will focus on the motivations behind the interface and the details of its creation. This thesis also includes an installation CD that contains all of the files necessary to install Compendium and the IDEA interface. The instructions for the installation of this CD can also be found in Appendix A.

4.1 Motivation

As was discussed earlier, the overall goals of IDEA are to create a concept design process that is robust and traceable, as well as easy to implement and use. Ease of use is of critical importance because no matter how robust a process may be, if it is time consuming and difficult to use, it will not be adopted widely. Despite its many advantages, IDEA as presented in the previous section is relatively time consuming. While the individual tools are not difficult to use, the high degree of traceability built into the process does require significant bookkeeping to track the large amounts of data that are generated. This detracts from the time engineers can spend actually designing. Not only is it tedious, but the constant need to write things down gets in the way of the creative process since the designer is constantly pushed away from thinking about ways to solve the problem.

Discussions with Professor Filippo A. Salustri [18] revealed that bookkeeping was the single greatest weakness of the IDEA process. However, a computational tool could be developed to do almost all of the bookkeeping work. This would allow a user to leverage the advantages of the underlying process while still working within a much more efficient and user friendly environment.

Furthermore, if a computational tool were developed on a popular and robust platform, IDEA will become more readily available to design practitioners. Leveraging open source software wherever possible will also keep initial and operating costs controlled.

Software has also been shown to also improve brainstorming activities [19]. Petrovick found that computer supported brainstorming was more productive and more enjoyable for the participants. Since brainstorming is an important element of IDEA, we can expect to see these benefits also, at least to a degree.

4.2 Creation of the Interface

Modern software platforms offer many alternatives for the developer. At the extremes, in this case, are (1) to write the IDEA tool from "scratch" in a language like C++, and (2) to leverage existent opensource tools and applications. Developing from scratch provides better development control and (eventually) efficiency of code, but it is very time consuming. Leveraging existent code allows far faster development at the expense of performance and control.

Leveraging existent code was selected for three reasons. First, the tool is intended as a proof-of-concept prototype, so performance efficiency is not as important. Second, given the academic constraint of the current work, there would have been no time to develop an implementation from scratch. Third, writing portable graphics software, even relatively simple two-dimensional graphics, remains a particularly difficult undertaking and something in which the author has no particular training.

An appropriate platform on which an IDEA tool could be built must satisfy the following requirements.

1. The platform must be open source, to eliminate cost, licensing, and intellectual property issues that could delay development and early adoption of the tool.
2. The platform must be relatively mature and reliable.

3. The platform must be able to run on any popular hardware and operating system (i.e. MS Windows and Linux, including Mac OS).
4. The platform must be sufficiently flexible to adapt to the needs of IDEA.
5. The platform must permit linking documents generated by other means (e.g. MS Word, Excel, etc) and "auto-launch" appropriate applications for them.

From the many potential candidates available, *Compendium*, by Verizon and The Open University UK stood out as meeting these requirements best. Originally developed to support the *Compendium methodology*, Compendium has since found use in various business and technical applications [20]. One of the most well known of these is the Jet Propulsion Laboratory (JPL) at the California Institute of Technology, where Compendium has been used in many innovative ways, including the coordination of rover movement and scientific experimentation in the Mobile Agents Project for the Spirit and Opportunity rovers [20].

Compendium can also manage files linked to it. For instance, it can prepare externally generated files linked to it (for example, MS Word documents) for sharing with other users and automatically start an external application when such a link is selected by a user. It also has a WYSIWYG ("what you see is what your get") editor to customise its behaviour and create templates for specific application interfaces. This makes Compendium quite easy to use.

The IDEA interface was written in Compendium, and designed to have a consistent layout and interface between the different IDEA modules. Interface elements that are uniform across all screens are always positioned in the same place, so users will more quickly become accustomed to them. The "home screen" that appears when the IDEA tool is first started is particularly important since users would typically spend more time using it than any other page.

The home screen, shown in Figure 13, reflects the IDEA process itself (see Figure 6).

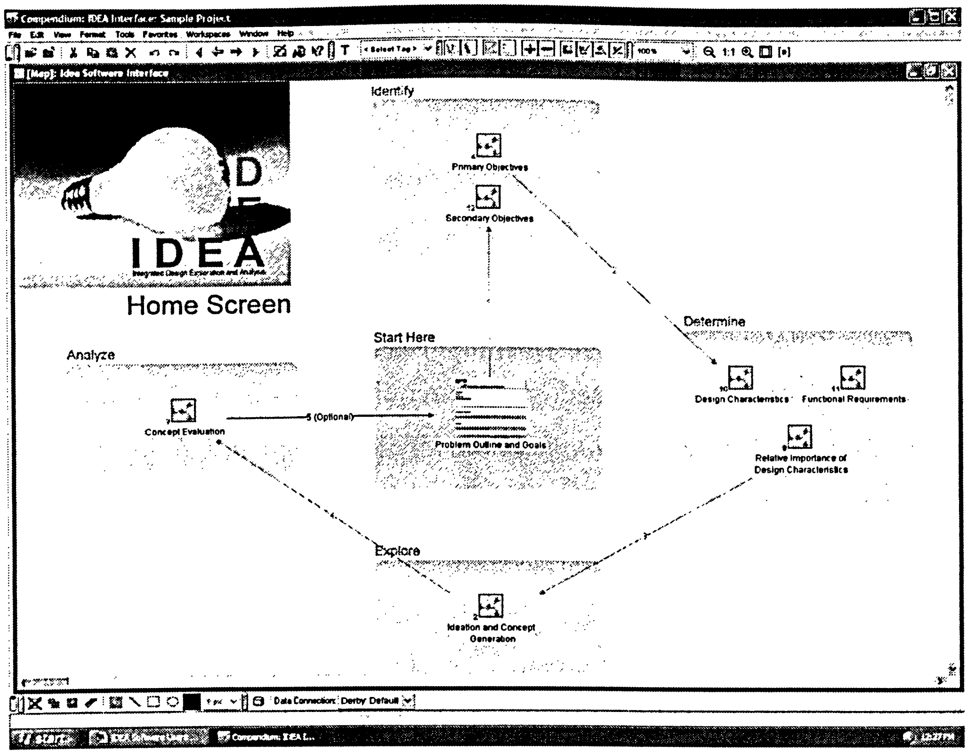


Figure 13: IDEA Interface Home Screen

The user starts in the box at the center of the screen and then follows the arrows in a clockwise fashion until they return to the starting point where they have the option of performing another iteration of the loop, or proceeding onto more detailed design. Red was selected for the starting point since people are naturally drawn to the colour, and a new user will instinctively look here. Further examination by the user will then reveal the overall workflow and they should quickly find themselves comfortable with the mechanics of navigating the home screen.

Figure 14 shows the screen produced by double-clicking on the Problem Outline and Goals icon. Note that actually Microsoft Word is running in Figure 14. While the capacity to start other programs is typical of many software packages, there is one important disadvantage. Since Compendium cannot know a-priori what programs might be started in this way, it cannot accommodate information integration with those other programs. The developers of Compendium have promised this capability is some future version. To make up for this deficiency, the IDEA tool incorporates a large number of template files. While this is a suboptimal solution, it does help lessen the data entry work that users have to perform.

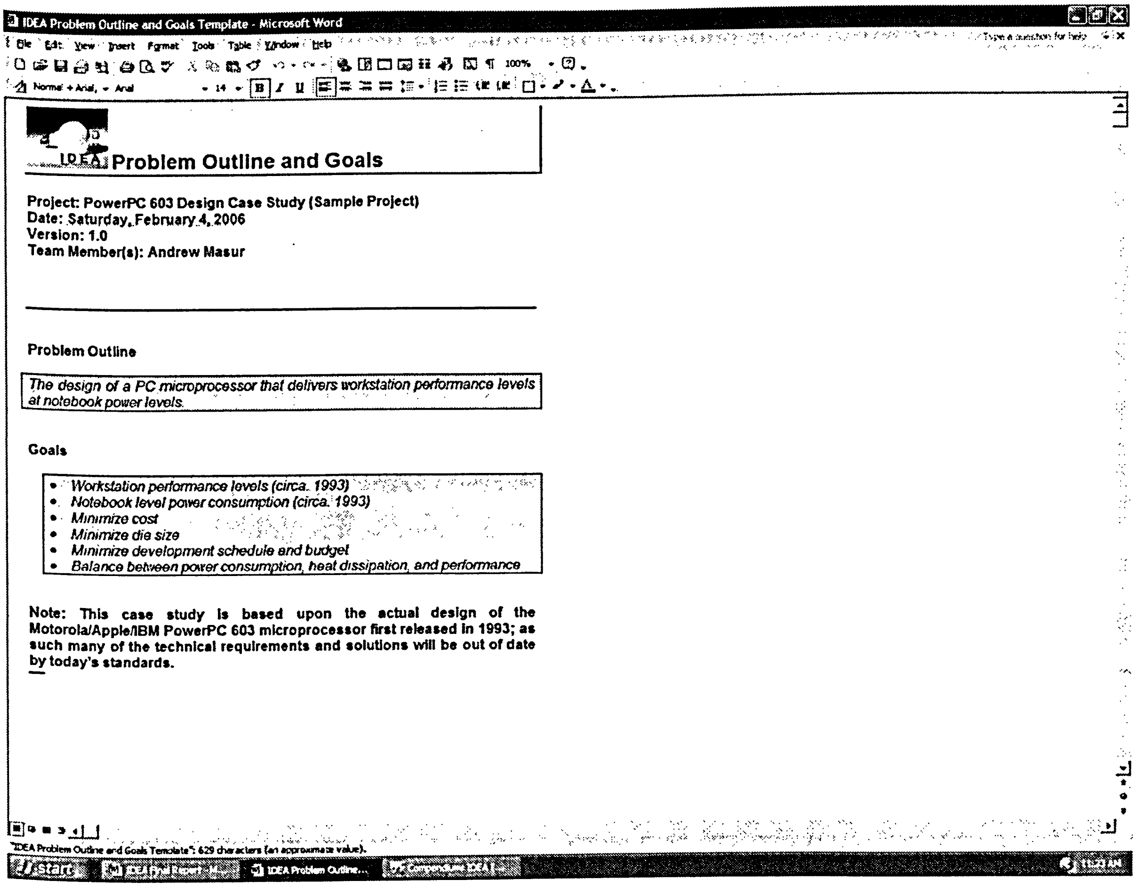


Figure 14: Problem Outline and Goals Document

The most serious problem is that users are now required to have a variety of application programs in addition to Compendium. Fortunately, Microsoft Office sees widespread use in the industry. There is also an open source alternative, OpenOffice, which is nearly identical to Microsoft Office and can be used by the IDEA tool. On Unix-based operating systems, for which Compendium is also available, OpenOffice is required.

4.3 Overview of the Interface

This section will provide an overview of the IDEA Software Interface and several of its notable features; a full description of all functions can be found in the IDEA Software Interface Users Guide contained in Appendix A.

As described above, the home screen of the interface is designed to resemble the IDEA process itself. After recording the outline and goals for the current problem, the user proceeds in a clockwise fashion through the various IDEA modules represented by the light-blue boxes on the home screen. Each box can contain several icons that represent the tasks to be completed for that module.

Double-clicking on one of these icons opens a window that contains tools that are specific to that task. For example, double-clicking on the Design Characteristics icon in the light-blue *Determine* box opens the following window displayed in Figure 15.

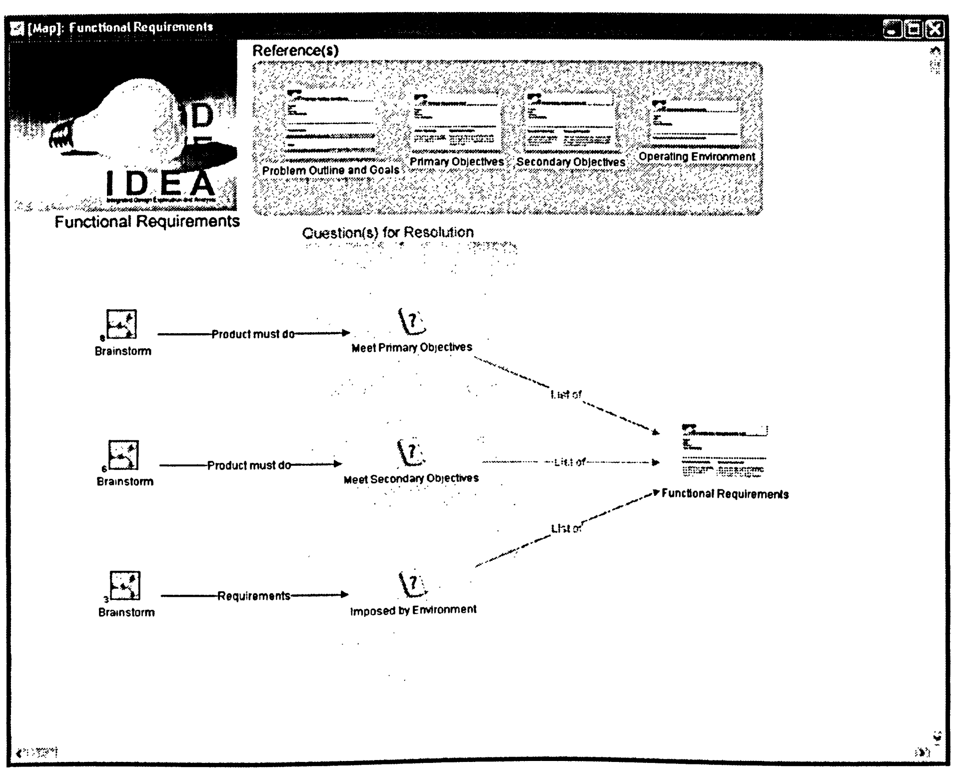


Figure 15: Functional Requirements Window

Clearly this window has been modified in order to determine the Functional requirements; however, the basic layout is common amongst all task windows in the interface. This helps to increase the comfort level of users since they will quickly learn where to find the tools and information they need to complete a particular task.

Figure 15 is arranged as follows. The top of the window contains a large grey box labelled "References" and contains relevant information such as data that was generated earlier in the IDEA process, links to relevant internet sites, etc. This area is common to all task windows, improving user comfort since relevant information can always be found in the same place.

The lower portion of the window is dominated by a centred blue column. The icon(s) in this column represent questions that must be answered by the designers in order to complete the task. The icon(s) to the left represent tools that can be used to answer these questions, while the icon(s) to the right represent where the answers to these questions are recorded, in this case a list of functional requirements for the design.

The diagram can be read from left to right in order to determine exactly what it is that the design team must do. For example, the top row of icons can be read as "Brainstorm what the product must do to meet the primary objectives and record this in the list of functional requirements". The structure of the diagram allows complex instructions to be presented in a very compact and understandable form.

The IDEA interface also includes templates that can be used to run brainstorming sessions. The template is the same for any IDEA brainstorming session. Again, this increases comfort with the interface and allows designers to spend more time designing. Double-clicking on a brainstorming icon opens the window in Figure 16.

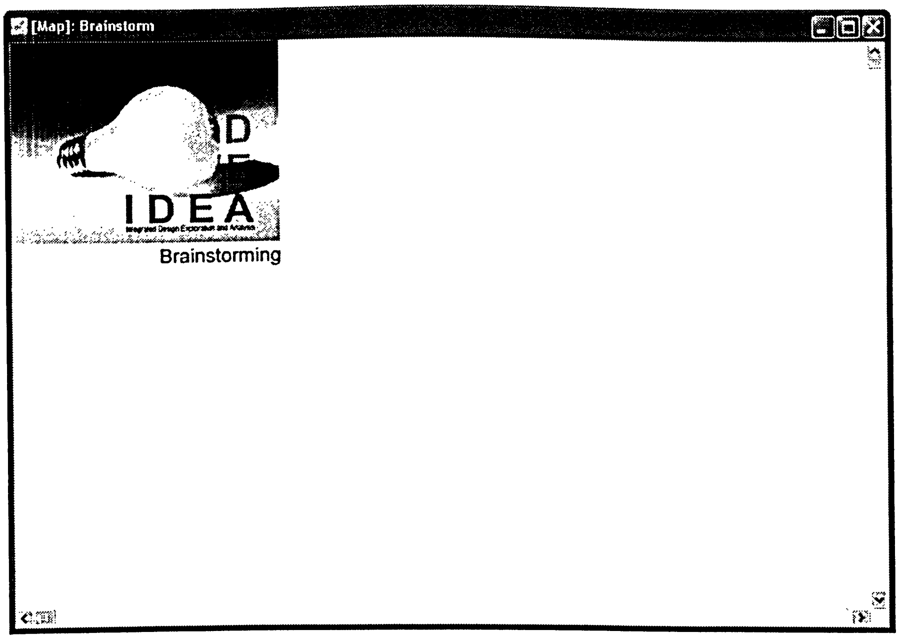


Figure 16: Empty Brainstorming Window

While the this template may simply seem to be an empty window, it is in fact perfectly suited for conducting brainstorming in Compendium itself; that is, it is as if the user had started a new Compendium brainstorming session.

Compendium was originally designed for recording the results of meetings like brainstorming sessions. This allows it to create the nodes and links that are required for a brainstorming diagram with the press of a single key on the keyboard and a single click of the mouse. Because of this, a brainstorming template was not required, and the lack of any implied structure can actually be of great benefit to designers since they are free to use whatever brainstorming method is most comfortable. Compendium is also flexible enough to help users create almost any type of diagram. The mechanics of creating these diagrams are covered in Appendix A; however, an example of one possible completed brainstorming diagram is presented in Figure 17.

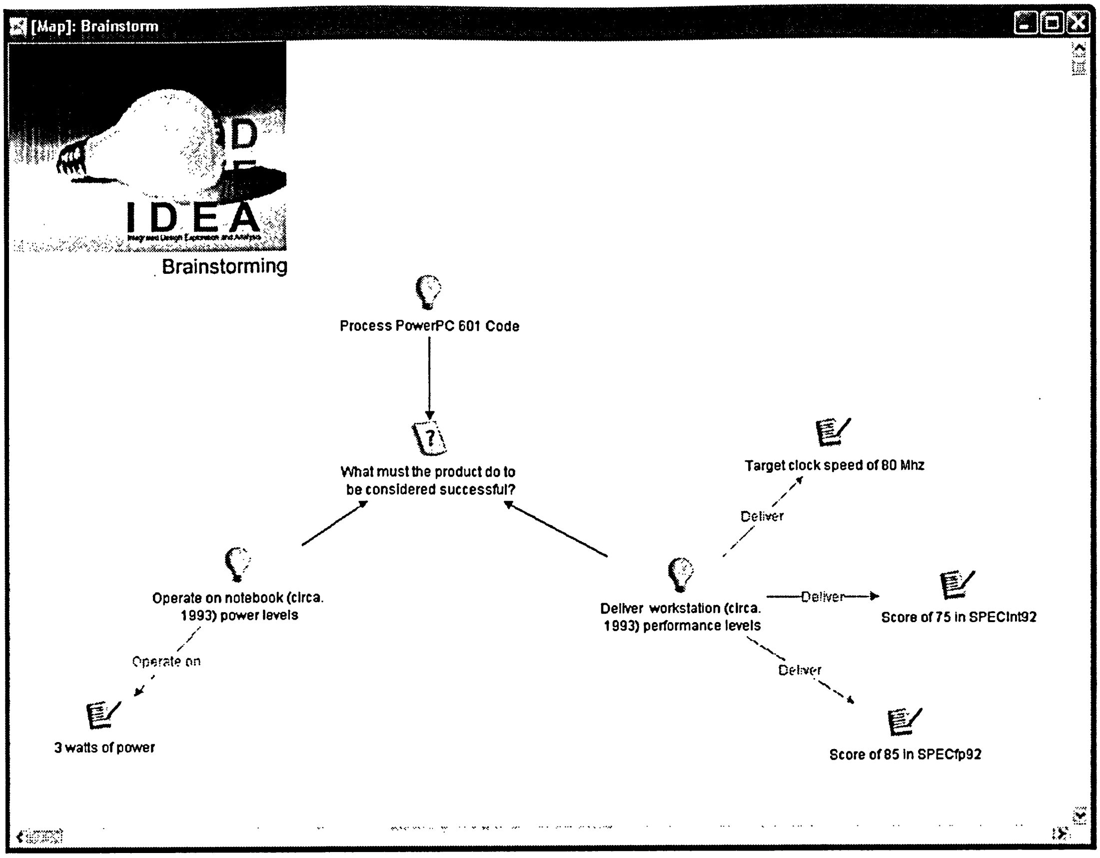


Figure 17: Completed Brainstorming Diagram

The above diagram closely adheres to the IBIS form specified in section 2.2.1, but conceivably any form of brainstorming diagram can easily be created through Compendium.

Figure 15 also reveals that the results of all brainstorming sessions are recorded in the functional requirements list. Ideally, this list would be automatically updated with the contents of the brainstorming diagrams, but this feature has not yet been implemented in Compendium. Instead, the list is stored as a Microsoft Word format file and designers must manually enter the contents of the brainstorming diagrams into this file.

While this may seem an unnecessary burden on the users, it actually has several advantages. First, by placing all of the items in a single list, designers only have to look at a single document to access all of their results. Second, the templates for recording brainstorming results also require designers to list the reasons why they have included that item on the list. By asking users to do this, the interface forces them to provide a rationale for their decisions, which is vital for downstream accountability. This increases the probability of catching design problems early, which increases the effectiveness of the design process.

Third, forcing designers to make manual records makes them think differently. For the most part, brainstorming templates are used as a graphical tool to record the results of brainstorming sessions. This causes designers to think *graphically*, skewing their creativity in one direction. Forcing designers to write down why they have included certain items causes them to change to a verbal thinking mode, which provides a different perspective. Not only does this increase the possibility of catching errors early in the design process, but it can also improve creativity and lead to more innovative designs.

Another feature of interest is the IDEA Concept Evaluation Workbook. The IDEA process uses both pairwise comparison and decision matrices to evaluate a collection of design concepts. Though these are conceptually simple techniques, they require a great deal of bookkeeping. The purpose of the workbook is to automate many of these tasks in order to reduce errors and increase the speed of concept design.

The workbook is in Microsoft Excel format and consists of three separate areas, each devoted to a different area of concept evaluation. The first section of the workbook is shown in Figure 18.

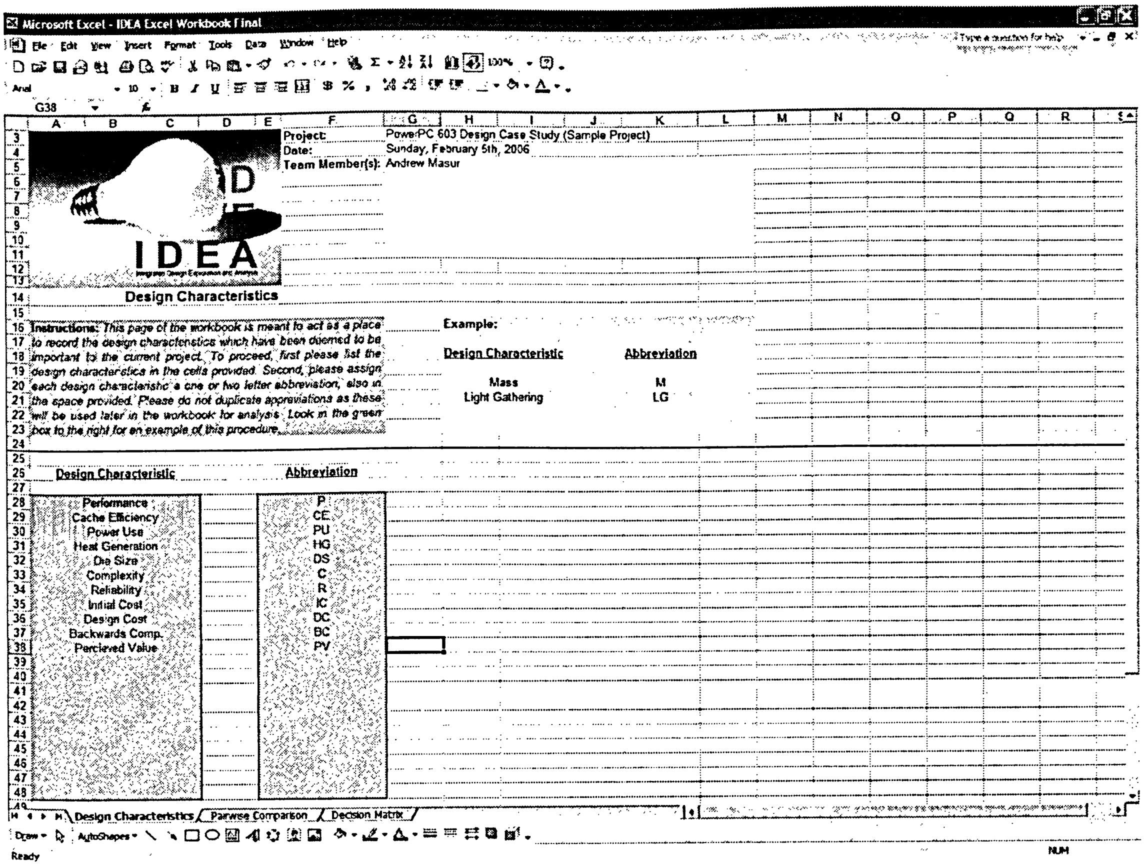


Figure 18: Workbook Section One

The goals for the workbook interface were the same as for the IDEA interface itself, namely, the interface should reduce bookkeeping, be easy to understand, and help to spur creativity. In order to improve clarity the workbook has a number of standardized features that are meant to reduce confusion for users. Each section includes a light-blue box along the top that contains instructions for that section along with a light-green box to the right with an example of the required procedure. Since these features are always in the same location, the user instinctively knows where to look if they become lost.

To complete the first section of the workbook, designers enter the design characteristics that they have determined in previous steps, and assign each of them a unique abbreviation. These abbreviations are then used in the pairwise comparison section as seen in Figure 19.

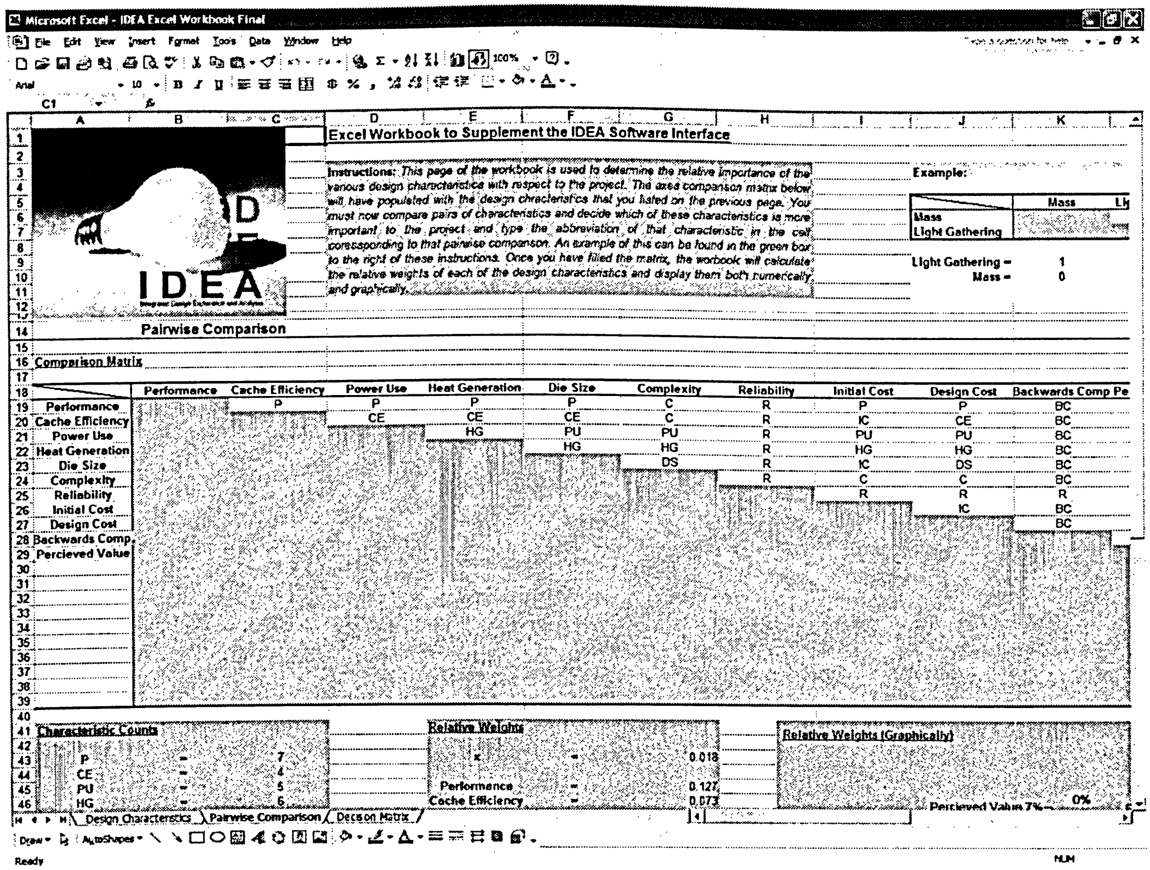


Figure 19: Workbook Section Two

Figure 19 demonstrates one of the key advantages of using the Concept Evaluation Workbook. Normally, the pairwise comparison chart would have to be generated by hand; however, the workbook automatically generates the chart based on the design characteristics. In addition, the workbook ensures that only valid comparisons are made by greying out the cells of the chart that have no meaning. This entire structure is updated in real time, and if the user were to return to the first page and add or remove design characteristics, then the pairwise comparison chart will automatically update to reflect those changes. This automation frees the design team to spend more time and resources actually conducting the pairwise comparisons rather than wasting it with routine bookkeeping tasks.

One weakness of the current version of the pairwise comparison chart is that it does not allow for two design characteristics to be tied. Various fixes were attempted to remedy this problem; however, as of this writing, no suitable solution has been found. As it currently stands, the design team must always select a winner from amongst each pairing of design characteristics.

Another advantage of using the workbook can be seen along the bottom of Figure 19 and is shown in detail in Figure 20.

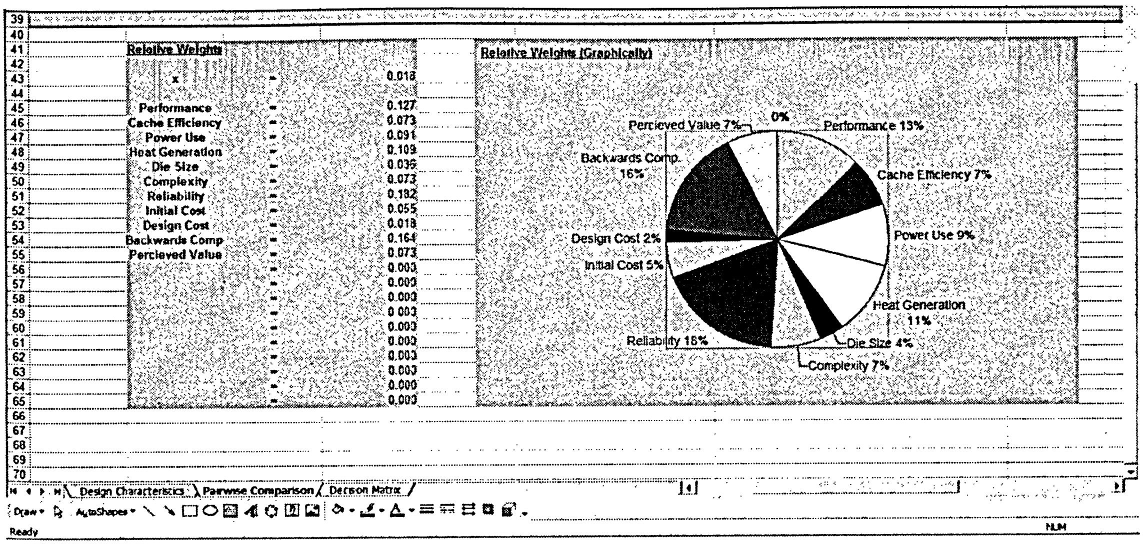


Figure 20: Workbook Section Two Lower Half

The workbook not only automatically generates the empty pairwise comparison chart, but also performs all of the calculations required to determine the relative importance of the design characteristics. This not only increases the speed of this process, but also eliminates mistakes from erroneous calculations. Additionally, the workbook provides an immediate visual representation of the relative importance of the design characteristics in the form of a pie chart. This allows designers to know at a glance which of the design characteristics is the most important to the design and whether the distribution of the relative weights makes sense. Both the calculations and the pie chart are updated in real time. This is particularly useful when multiple stakeholders are discussing exactly what the relative weights should be since the proposed changes can be seen immediately.

The third section of the workbook is dedicated to the WDM used to select the "winning" concept. Typically this matrix would be created by hand; however, the Concept Evaluation Workbook automatically creates a matrix for up to ten concepts using data from the previous two sections. The design team merely has to fill in the appropriate sections of the matrix using the techniques described in Section 2.3.3. An example of a completed WDM is presented in Figure 21.

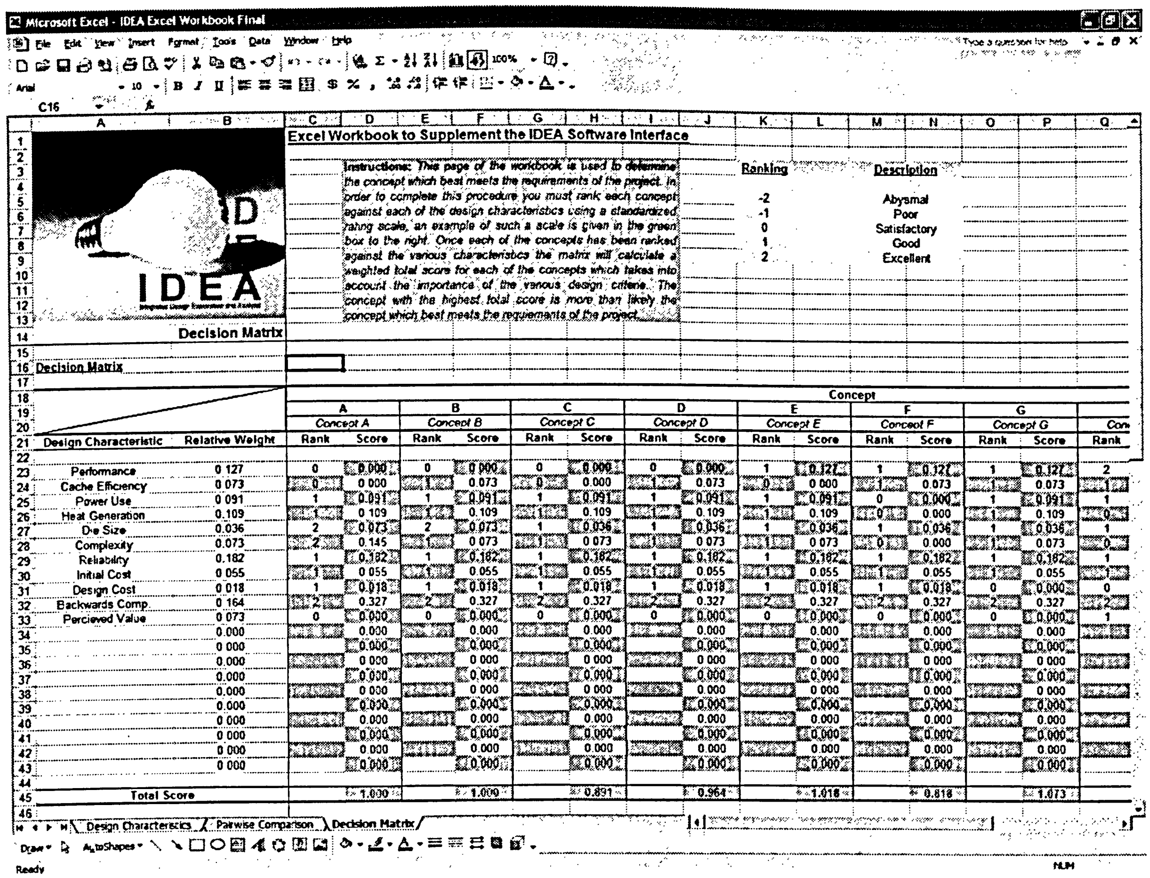
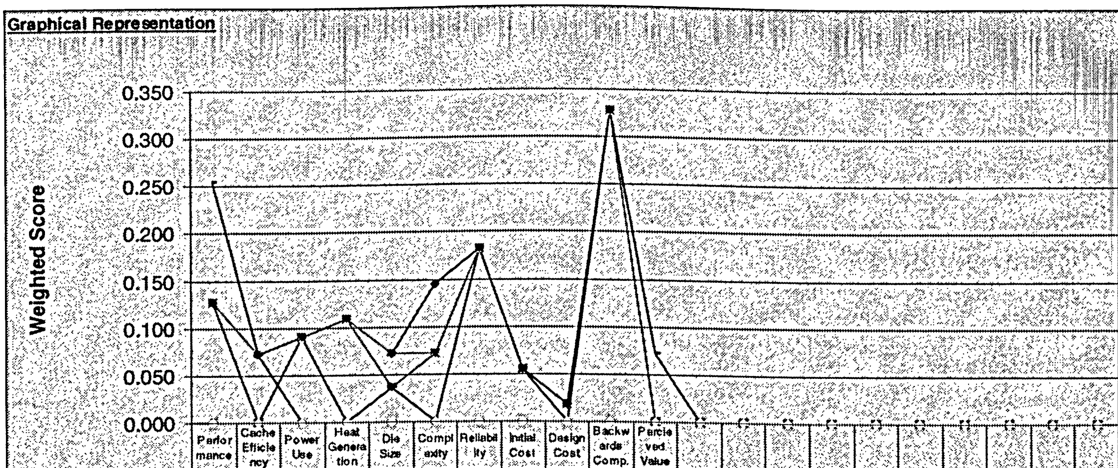


Figure 21: Workbook Section Three

In addition to automatically generating the weighted decision matrix, the workbook is also able to perform all of the calculation required to generate the aggregate scores for the concepts. While the calculations are not difficult, having the computer perform them saves time and ensures that they are performed correctly. This is especially important in this step since the results of these calculations will determine which of the concepts moves on to the more detailed design phase of the engineering process. A mistake made here could lead to the selection of a suboptimal concept that would have a dramatic impact on the success of the project.

These values are constantly updated in real time so any changes made in the first two sections of the workbook are immediately evident here. This can be of great advantage in a situation where a company has a standard set of components or designs that they use to solve similar problems. Instead of redoing the whole analysis every time they have to choose between these components, the components and their ratings could be left in the matrix. In most problems, only the relative weights of the design characteristics would change due to the design problem. Because the WDM reflects changes made in the first two sections, the ratings of each standard component would change depending upon the weights of the design characteristics for the current problem. Designers could then consult the matrix to see which of the standard components is the most suitable for the current problem without having to redo the whole analysis. This saves a great deal of time and suggests how the IDEA process can seamlessly be integrated in to existing engineering practices at a company.

One additional feature of the Concept Evaluation Workbook is its ability to represent the results of the WDM graphically. This is demonstrated in Figure 22.



	Performance	Cache Efficiency	Power Use	Heat Generation	Die Size	Complexity	Reliability	Initial Cost	Design Cost	Backwards Comp.	Percieved Value													
Concept A	0.000	0.000	0.091	0.109	0.073	0.145	0.182	0.055	0.018	0.327	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Concept B	0.000	0.073	0.091	0.109	0.073	0.073	0.182	0.055	0.018	0.327	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Concept C	0.000	0.000	0.091	0.109	0.036	0.073	0.182	0.055	0.018	0.327	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Concept D	0.000	0.073	0.091	0.109	0.036	0.073	0.182	0.055	0.018	0.327	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Concept E	0.127	0.000	0.091	0.109	0.036	0.073	0.182	0.055	0.018	0.327	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Concept F	0.127	0.073	0.000	0.000	0.036	0.000	0.182	0.055	0.018	0.327	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Concept G	0.127	0.073	0.091	0.109	0.036	0.073	0.182	0.055	0.000	0.327	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Concept H	0.256	0.073	0.091	0.000	0.036	0.000	0.182	0.055	0.000	0.327	0.073	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Enter name here	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Enter name here	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Figure 22: Graphical Representation of a Decision Matrix

This graph is located below the decision matrix and is constantly updated as the contents of the matrix change. Each of the lines represents the strengths and weaknesses of a particular design. While a WDM is good at showing this information, a graphical form provides a different perspective. With the graph it is possible to tell at a glance how balanced a particular concept is and whether or not it has any significant strengths or weaknesses.

For example, a concept may get the highest total score in the WDM, but it may do so by scoring well in the two most important design characteristics while being seriously deficient in all the rest. The WDM would show this information; however, it can become very easy for designers to overlook if there are a large number of concepts or design characteristics represented. The clarity of the visual presentation lets designers see these problems right away, and might sway them into choosing a lower scoring, but ultimately more balanced solution.

A third feature of note in the IDEA interface is the Trade Space Analysis tool. Located in the *trade space analysis* option of the *Explore* module, the tool helps designers create trade space diagrams. While the details of the tool's operation can be found in appendix A, the user would normally proceed by clicking on the Enter Design Options button. This opens the design option entry window as shown in Figure 23.

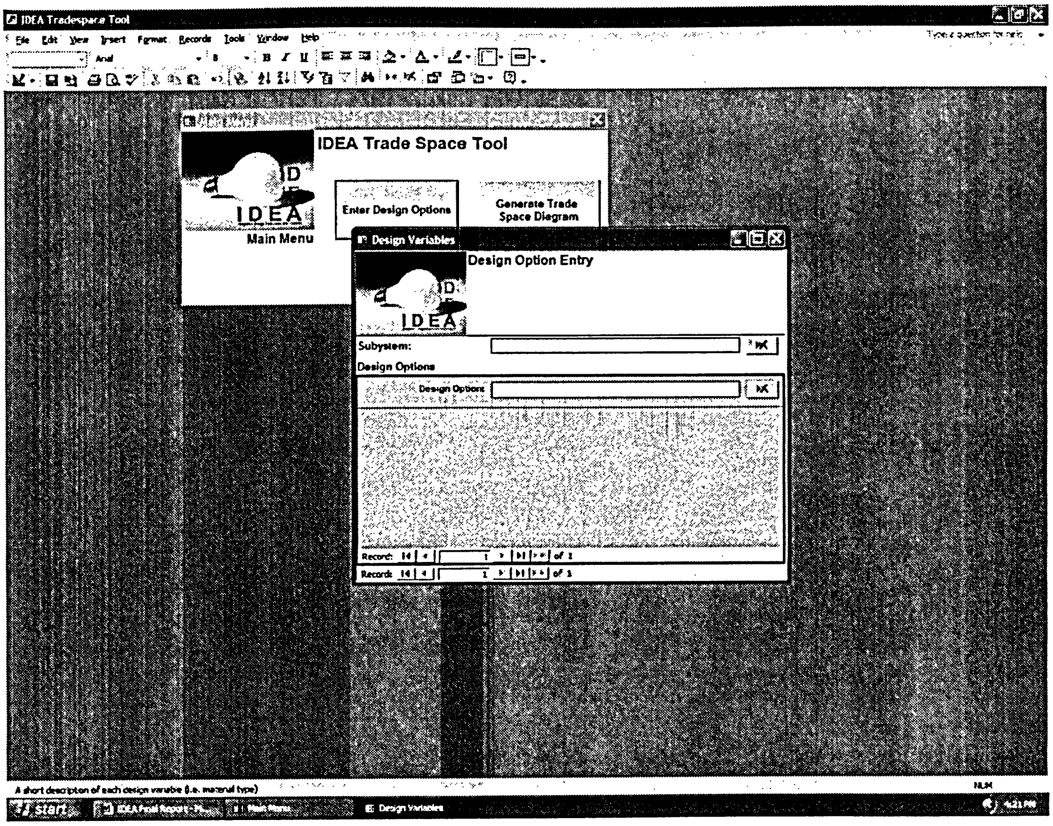


Figure 23: Trade Space Tool Option Entry

The design team can use the tool not only to generate a trade space diagram but also to keep track of their discussions during trade space analysis. First, the team enters the name of the sub-system of current interest in the appropriate box of the design option entry form. They can then record all the options for that particular sub-system in the light blue section in the lower portion of the entry form. Once all of the options a sub-system have been exhausted, the team moves onto the next sub-system and repeats the procedure.

There are several advantages to using the tool to record this data. First, designers can spend more time thinking about the design because all the necessary templates are provided.. Second, filling in an entry amounts to simultaneously preparing the data to generate a trade space diagram. Typically, the design team would need to generate the trade space diagram manually; however, if designers use the tool during their discussions, it can generate the diagram automatically. This saves a great deal of time and effort.

The trade space diagram is generated by selecting the Generate Trade Space Diagram button on the main IDEA menu. The tool then uses the data in the form to generate the diagram. An example of trade space a diagram generated by this tool is in Figure 24.

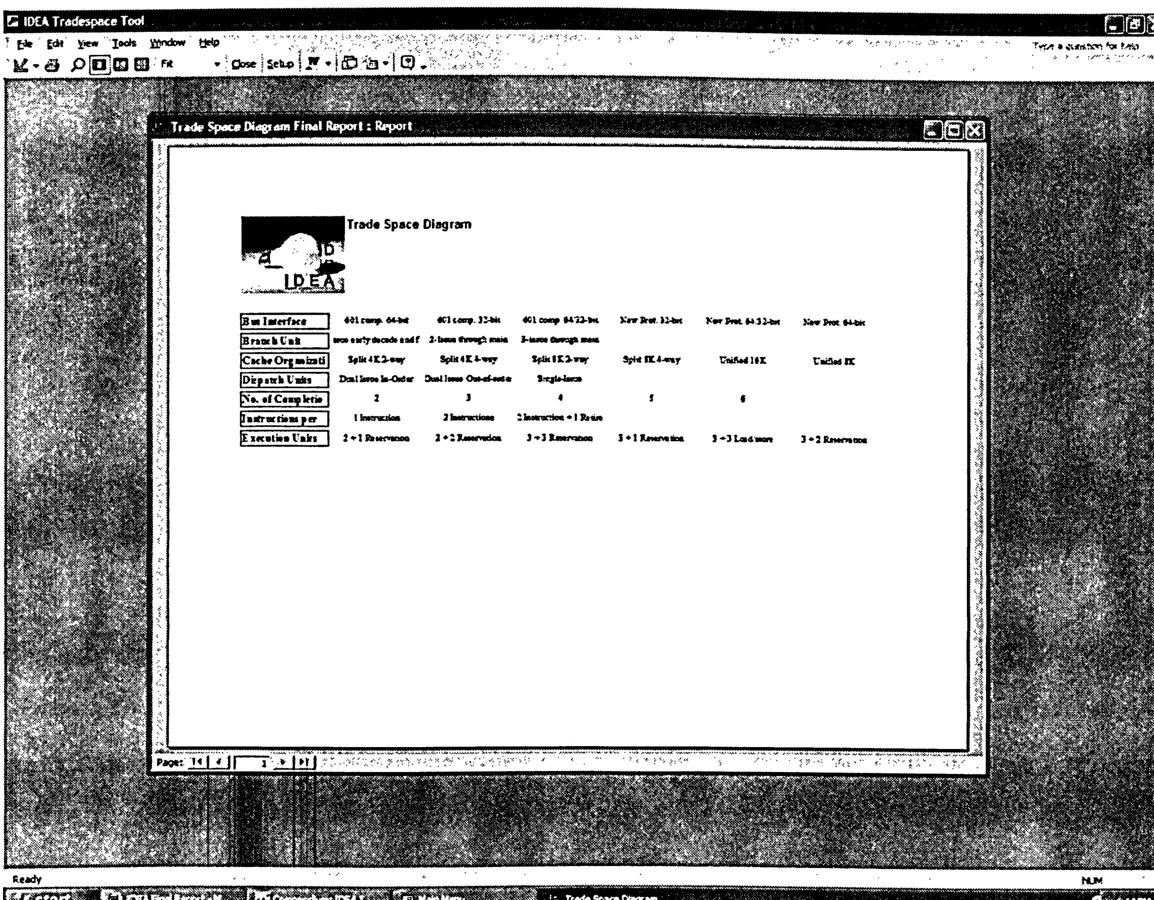


Figure 24: Trade Space Diagram from the Tool

The generated diagram is very similar to the one presented in Section 2.2.2; however, the effort required creating it was minimal. This underscores the key purpose of the IDEA interface. It has taken a task that would normally require substantial, tedious bookkeeping and has made it a far more efficient.

5.0 Case Studies

The following three sections will outline three case studies that were conducted in order to test the effectiveness of the IDEA process. Each case study was conducted using the IDEA Software Interface to test its functionality and to look for errors in the program. Each case study was conducted on a problem to which the final answer was already known, to determine whether IDEA can generate reasonable results. The first case study will be presented in significant detail, while the others will simply present the results and any other interesting features to reduce the length of this document. The save files for each of these case studies can be found on the included installation CD.

5.1 Case Study 1: Martian Sample Processing Unit (SPU)

5.1.1 Problem Overview

The first case study is based on the author's undergraduate thesis and deals with the design of a Sample Processing Unit for the Mars Science Laboratory (MSL) mission. Built upon many of the technologies pioneered in the current Martian rovers, Spirit and Opportunity, the MSL rover will use a small nuclear power source to increase its range and expected lifetime. The rover will contain a comprehensive array of scientific instruments that will be able to analyze samples of the Martian soil in far more detail than any previous surface mission.

One of the key components of these scientific systems is the Sample Processing Unit; or SPU for short. The purpose of the SPU is to accept either solid core or loose surface samples from the Martian surface and process these samples down into a fine particulate suitable for further analysis by the rover's scientific instruments. While this would be a simple exercise on Earth, when implemented on a Martian rover there are many challenges that such a device must overcome.

As with most hardware intended for use in space, there are significant limits on the mass of the final product. This is particularly true for interplanetary missions like the Mars Science Laboratory. Because any interplanetary spacecraft must attain sufficient velocity to overcome Earth's gravity, there are harsh limits placed on its allowable mass. These limits are in place to avoid the use of an overly large launch vehicle that would increase the cost of the mission, or if a large enough vehicle cannot be found, preclude it all together.

Limits on spacecraft size are often directly coupled to mass constraints. Launch vehicles commonly used for interplanetary spacecraft not only have strict mass requirements, but have limited volume as well. This places strict constraints on the overall volume and configuration of the final design in order to allow it to fit in the launch vehicle.

Another common challenge for space hardware is the limited amount of electrical power available. Many interplanetary spacecraft use the Sun as their primary energy source. This is a particular challenge for Mars surface craft since the dust in the Martian atmosphere will quickly coat solar panels, drastically reducing the amount of power available. While the MSL will not suffer from this problem due to its nuclear power source, the amount of power available is still limited which imposes a significant power constraint on the final design of the SPU.

While these constraints are the main drivers of the final design there are many more which affect the SPU. When the project was originally undertaken, assistance was imparted by engineers at MDA Space Systems who provided many pages of requirements for the final design. These requirements will be covered later in this case study.

5.1.2 Reasons for Selection

There were a number of reasons which lead to the selection of this problem as one of the case studies. The first of these was a familiarity with the problem. As outlined before, this same problem was the basis of the author's undergraduate thesis. Because of this there was a great deal of familiarity with the subject material. This would allow the focus of the case study to be on how well the IDEA process worked and not the problem itself.

Because this case study was based upon a previous thesis [21], there was a great deal of material available on not only the problem, but also the final solution. This would allow for a comparison between the original and IDEA generated results. The original solution was deemed quite innovative by MDA and IDEA would have to either produce the same solution or one that was equally innovative [8].

The third reason for the selection of this problem was the simple fact that the earliest inspiration for the IDEA process came from the original undergraduate thesis. As stated earlier, on many visits to MDA the author observed concept design in use at the company. The most notable deficiency with the concept design process in use at the time was a perceived lack of any kind of mechanism to assist with the selection of concepts. One of the goals of the IDEA process was to address this deficiency and the results of the case study would help to prove whether or not that had been accomplished.

5.1.3 Discussion of IDEA Process Steps

This section will outline the results of the various steps in the IDEA process as they pertain to the Martian SPU case study. It is only meant to serve as an overview of the results and a more detailed discussion about the actual steps can be found in either Section 3 or in Appendix A.

The first step in the IDEA process is to generate a Problem Outline and Goals document for the current design problem. Using the information contained in section 5.1.1 along with thinking about other factors which might influence the design resulted in the following outline and goals for the SPU.

Problem Outline

The design of a Sample Processing Unit (SPU) for the Mars Science Laboratory (MSL) mission which will accept either a solid core or loose surface regolith sample and reduce it to a fine particulate for further analysis by NASA supplied scientific instruments.

Goals

- *Low power use*
- *Low mass*
- *As small a volume as possible*
- *High reliability*
- *Able to resist the Martian environment*
- *Able to handle a wide array of sample types*
- *Maintain sample integrity*
- *Ability to dump waste samples*
- *Maintain low cost*
- *Ability to break down large samples into smaller portions*
- *The ability to monitor several parameters during sample processing*
- *Minimize contamination of both Earth and Martian environments*

Clearly there are a number of goals here which were not immediately evident in the problem overview. Some of these goals were included in the original design requirements document supplied by MDA while others were determined by simply thinking about issues which might affect the problem.

Some of the goals, like high reliability, are very self evident. With current technology it would be very difficult, if not impossible, to send a human to fix a serious problem with the rover. This requires that the final design not only prevents problems from occurring in the first place, but to also allows for automated and remote recovery from all but the most serious of errors.

Other goals, such as maintaining sample integrity, are not as evident to the casual observer. To the layperson, processing one sample is much the same as processing another; however, to a scientist this is definitely not the case. For example, if scientists are attempting to determine the differences in composition between the soils at two different sites it is imperative that the results from one location do not impact the results from another. In most cases it would be preferable to return no results at all rather than erroneous ones. Erroneous results could lead to the wrong conclusions which may have a detrimentat impact on the scientific findings of the mission. For this reason it is critical that the SPU prevent cross-sample contamination, and if there is any suspicion of possible contamination it should be able to dump the sample in question as waste.

The next step in the IDEA process was to determine the primary and secondary objectives for the Martian SPU. To review, the primary objectives are things that the final design must do in order to be considered successful while the secondary objectives are things that should be included in order to add value for the customer. The primary and secondary objectives identified for the SPU are presented in Tables 8 and 9.

Primary Objective(s)	Reason(s) for Inclusion
Must process loose and solid regolith according to required parameters	*At the core this is what the SPU is. Everything else is designed to support this function, if it does not do this, then it has to reason to exist.*
Must prevent cross-sample contamination	*This is critical since cross-sample contamination could lead to erroneous results. It would be better to return no results rather than incorrect ones.*
Must prevent contamination from Earth	*Again, any contamination from either residue from Earth or the SPU itself must be avoided to prevent the return of erroneous results to Earth.*

Table 8: SPU Primary Objectives

Secondary Objective(s)	Reason(s) for Inclusion
Ability to monitor sample parameters during processing	*This ability was specifically stated in the problem outline provided by MDA. There is a possibility that this ability could be added using primary objective hardware*
High reliability	*This is very desirable since it will directly lead to a longer potential mission life. Again, more reliable primary objective equipment will lead to higher reliability overall*
Low mass/volume	*This is important because the SPU will be mounted on a rover and space will be very limited, thus lowering the mass and volume will be very desirable.*
Ability to divide samples into smaller portions	*This ability could be useful because it will minimize waste if a particularly large sample is processed because the sample can be scanned in manageable chunks. This ability will also lower overall power consumption since the crusher can process a larger sample all at once and then remain idle while the tests are carried out.*
Ability to dump waste samples	*This requirement was explicitly stated in the problem outline provided by MDA. The ability to dispose of a sample if there is a chance that it has become contaminated could be quite useful.*
Ability to transfer raw regolith to the crusher	*While at first this may seem like a primary objective it was deemed secondary since this function could perhaps be better served by another sub-system on the rover*

Ability to transfer processed samples to the sensors	*Again, this may seem like a primary objective, but it is possible that this function could be relegated to another subsystem, hence its inclusion in the secondary objectives.*

Table 9: SPU Secondary Objectives

From the above tables it is clear that not only are there a large number of objectives, but all of the objectives help to fulfill at least one of the design goals.

It is also interesting to note how much clearer the problem has become after the generation of the primary and secondary objectives. Through these there is now a much crisper understanding of exactly what kind of characteristics the final design should have. During the original undergraduate work there was no formal analysis of the primary and secondary objectives. This made it much more difficult to determine exactly what capabilities the final design should have. An example of this can be seen in the secondary requirement for transferring processed samples to the sensors.

In the original analysis, it was simply assumed that the SPU would have to perform this task. During the final design there were great difficulties in integrating this function and in the end it was simply ignored due to time constraints. A more thorough analysis utilizing the IDEA process reveals that perhaps this function did not have to be performed by the SPU. It is likely that this function could be passed off to another sub-system, thus simplifying the design of the SPU.

Once the primary and secondary objectives were identified, the next step was to determine the design characteristics that would drive the design and be used as the benchmark to measure the effectiveness of the concepts. These characteristics were generated by using the standardized design characteristic list and the brainstorming tools built into the IDEA interface. There was no need to use the electronic interview form since this case study was an academic exercise performed essentially in a vacuum. The results of the brainstorming session are presented in Figure 25.

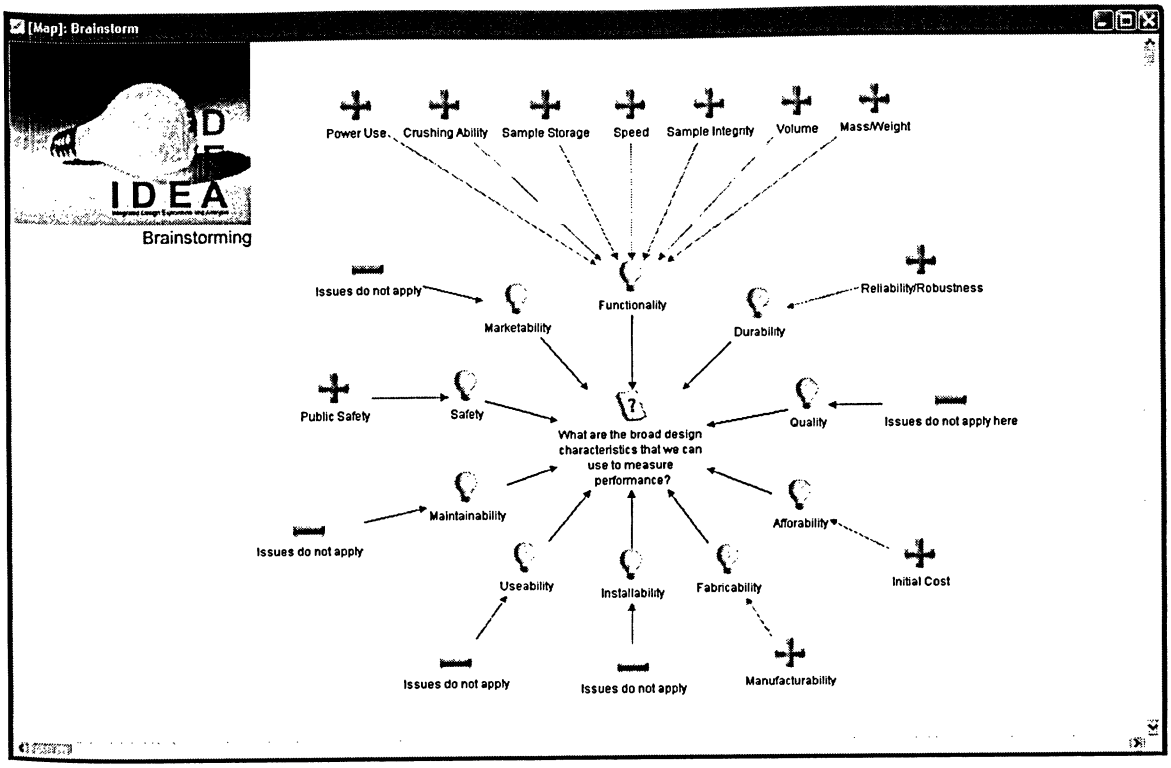


Figure 25: SPU Design Characteristics

Figure 25 demonstrates that there are no fewer than eleven different design characteristics which were deemed important to the design problem. This helped to ensure that many different aspects of the design were studied prior to making the final selection. These design characteristics were then entered into the first section of the IDEA Concept Evaluation workbook so that they could be used later in the concept design process.

The next step in the IDEA process was to determine the functional requirements for the SPU. While the design characteristics provide a better idea of the physical characteristics for the final design, the functional requirements outline exactly what the SPU has to do.

This process was initiated by researching the environment that the SPU will operate in. Typically, people do not think of environmental factors when faced with a design problem, but when a device must operate in an environment as extreme as the surface of Mars the conditions there can be a dramatic effect on the final design.

Using data from the undergraduate thesis [21] the following data was used to model the Martian environment.

Atmospheric Pressure: 11 millibars
Maximum Temperature: 20 ^{o}C
Minimum Temperature: -130 ^{o}C
Atmospheric Constituents: 95.3% Carbon Dioxide
2.7% Nitrogen
1.6% Argon
0.13% Oxygen
Gravity: 38% of Earth's; 3.72 m/s^2
Other Factors: Fine dust (0.02mm particle diameter) suspended in the Martian atmosphere. Winds can accelerate this dust to several hundred kilometres per hour.

Clearly, such an extreme environment would have an effect on the final design. In the case of the Martian SPU example these effects can be seen in the latter three functional requirements as outlined in Table 10.

In addition to the requirements imposed by the environment, the primary and secondary objectives also resulted in a number of functional requirements. These requirements were determined through the use brainstorming techniques and the results of these sessions are presented in Table 10.

Functional Requirements	Reason(s) for Inclusion
Process raw regolith	*Quite simply, this is what the SPU has to do. If it does not do this then it is not a success.*
Prevent cross-sample contamination	*This is also critically important*

	since no results are better than returning erroneous results.
Transfer raw regolith to the crusher	*This is also an important function of the design, but it may be possible to integrate this functionality into one of the other sub-systems*
Divide samples into smaller portions	*This would be a desirable function to have since it would save processing time and power*
Monitor samples during processing	*This is a requirement that was handed down by MDA. One can see its importance in sensing blockages or other such problems with the crusher which could possibly help with finding a solution to these problems*
Dump waste samples	*Again, this function was handed down by MDA. One can see the advantage of this function if for some reason scientists back on Earth thought that the current sample had become contaminated in some way.*
Transfer processed samples to the sensors	*This is an important function because if the samples cannot be studied there is no real reason to process them. This is another function which may have the possibility of being transferred to another sub-system.*
Resist large temperature swings	*This function is imposed by the Martian environment which is subject to temperatures that range from 20 to -130 degrees Celsius*

	every day
Resist contamination by airborne dust	*Again this is imposed by the Martian environment whose atmosphere is laced with fine airborne dust with particles as small as 0.02mm*
Able to operate under low air pressure	*This functional requirement stems from the fact that Mars has a very thin atmosphere*

Table 10: SPU Functional Requirements

Examination of the functional requirements reveals that they are split into two very distinct categories. The first of these, known in IDEA as active functional requirements, are things that the final product must actively do in order to be considered a successful design. For example, processing raw regolith is something that the SPU must actively do in this case.

The second group of functional requirements, known in IDEA as passive functional requirements, are also things that the final design must do, but they can be thought of as going on behind the scenes. For example, looking at the functional requirements for the SPU we see that the SPU must resist contamination by airborne dust. We can infer from the wording that the SPU is performing the act of resisting contamination; however this is likely to be done in a passive fashion. The resolution of this functional requirement will more than likely be a product of the structure of the SPU rather than an active system.

The next task, determination of the relative importance of the design criteria, was accomplished through the use of the pairwise comparison section of the concept evaluation workbook. The pairwise comparison chart had been automatically generated using the data from the first section and the only task now was actually performing the pairwise comparisons. This was accomplished using the techniques described earlier in this report and the results can be seen Table 11.

	Power Use	Crushing Ability	Sample Storage	Speed	Sample Integrity	Volume	Mass/Weight	Reliability	Initial Cost	Manufacturability	Public Safety
Power Use		CR	P	P	I	P	W	R	P	P	P
Crushing Ability			CR	CR	I	CR	W	R	CR	CR	CR
Sample Storage				SS	I	V	W	R	SS	SS	SS
Speed					I	V	W	R	S	S	S
Sample Integrity						I	I	I	I	I	I
Volume							W	R	V	V	V
Mass/Weight								R	W	W	W
Reliability									R	R	R
Initial Cost										C	C
Manufacturability											M
Public Safety											

Table 11: SPU Pairwise Comparison Chart

Using this data, the workbook automatically calculated the relative importance of the design characteristics. The results of these calculations are shown in Figure 26.

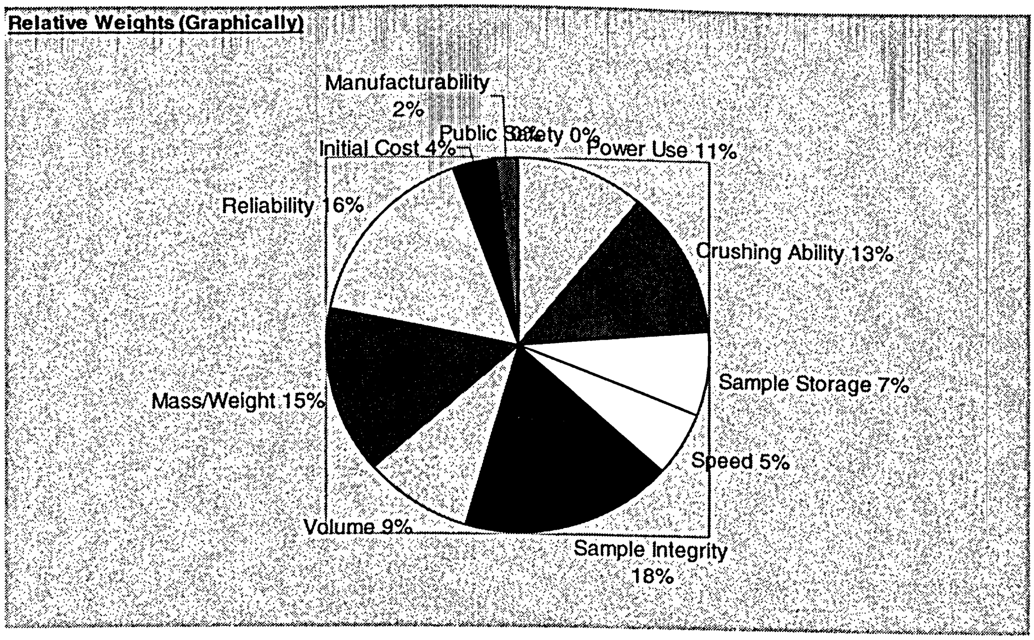


Figure 26: Relative Weights of the SPU

Figure 26 clearly demonstrates that sample integrity is the most important characteristic followed closely by reliability and mass/weight. The decision was made early on to return no results at all rather than erroneous ones. Thus, in order to ensure mission success, it was imperative that sample integrity be maintained during all stages of processing, even if it meant a reduction in performance with regards to other design characteristics. These results were consistent with those that had been obtained during the analysis for the undergraduate thesis. Since those were considered to be "good" numbers, this result is strong evidence for the validity of the IDEA process up until this point.

Armed with a better understanding of the specifics of the design problem and a clear definition of the characteristics and functions of the final design, the task now became actually generating concepts for the design problem. Trade space analysis was selected for this case study since it was being conducted by only a single individual and a number of different options had already been generated for the original study.

The first step was to determine options for each of the functional requirements. These options were entered into the IDEA Trade Space Tool and the trade space diagram in Figure 27 was generated.

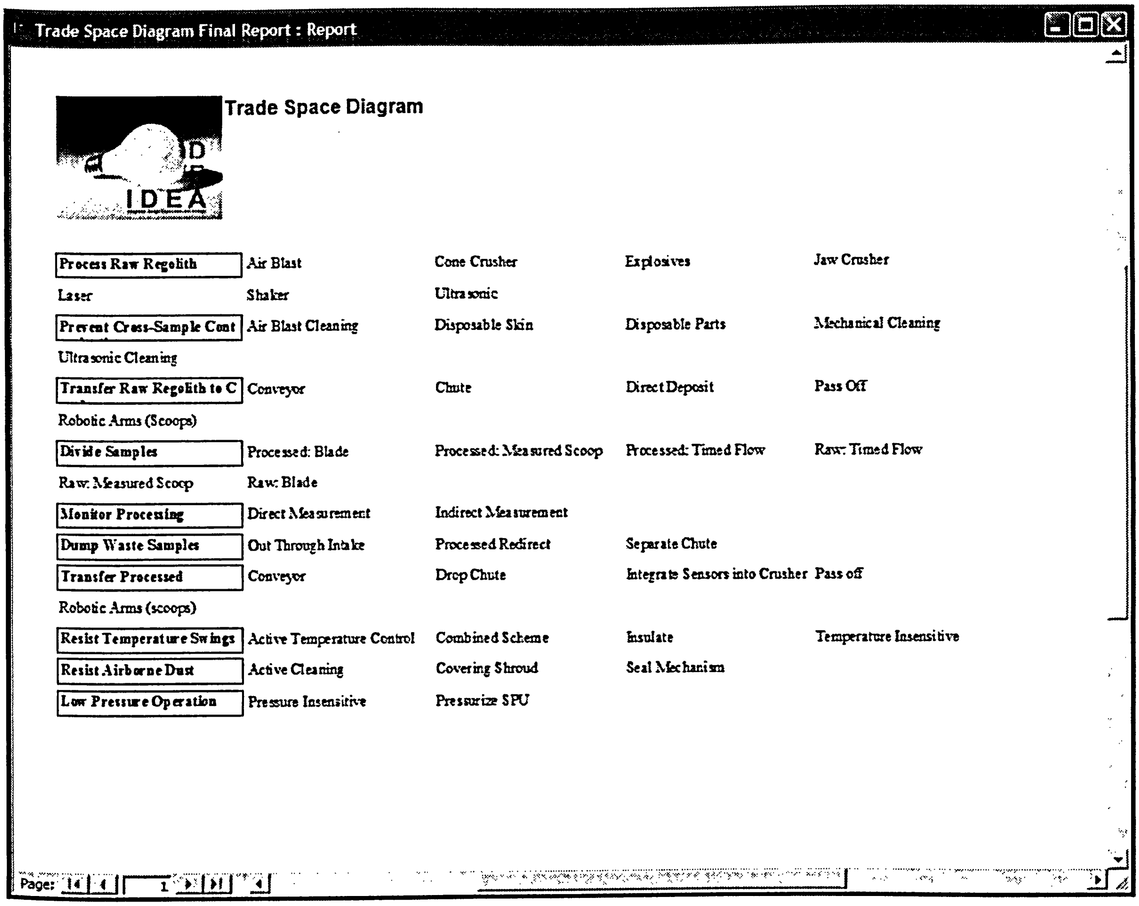


Figure 27: SPU Trade Space Diagram

From the diagram above, we see that there are many design options generated for each functional requirement. Concepts are generated by combining these options in different ways. An analysis of the diagram revels that there are no less than 756,000 different possible combinations, and each one of these combinations represents a different concept. While many of these concepts will have significant weaknesses, the sheer number indicates that there has been a thorough exploration of the trade space. Clearly, analyzing

756,000 different concepts in any detail is unrealistic; however, it is better to have far too many concepts available rather than too few.

With the trade space diagram complete, the next step was to generate a smaller sub-set of concepts to undergo further suitability analysis. This was accomplished by stepping through the trade space diagram and trying different combinations of design options. In order to simplify this task it was decided to make sample processing the deciding factor for the design since the performance of the crushing sub-system would have the greatest impact on the overall performance of the concepts. Figure 28 shows the six most plausible concepts.

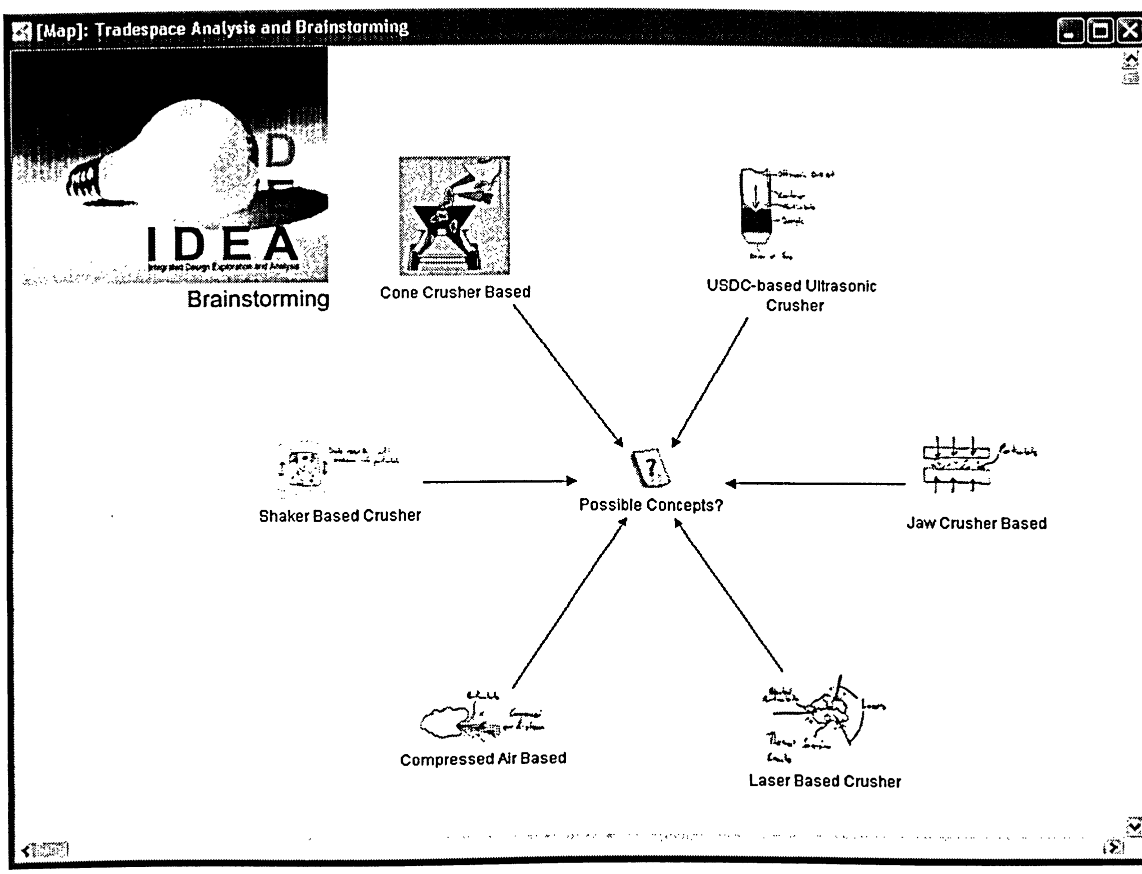


Figure 28: SPU Plausible Concepts

Clearly this is a far more manageable set than the one represented by the original diagram.

It is interesting to note that in the original undergraduate thesis there were a total of only three concepts generated, while the IDEA process has helped to select the best six out of more than three-quarters of a million different possibilities. What is also interesting is that the concepts themselves demonstrate a wider variety of designs when compared to the concepts in the original work. Both of these facts demonstrate not only a larger sampling of the problem trade space, but a far more diverse one as well. This wide range of solutions, coupled with the fact that the number of plausible concepts has doubled, increases the possibility of generating the "best" solution for the final design.

The final step of the trade space analysis process is to once again screen the plausible concepts in order to determine if any of them can be removed. Typically this is accomplished through brainstorming and discussion; however, designers should take extra care when discounting concepts at this stage. If there is any doubt as to whether or not a concept should be removed, it is better to err on the side of caution and leave that concept for more formal analysis.

The final step in the IDEA process is to select the "best" concept in order to proceed to the more detailed design stage. This is accomplished using the Weighted Decision Matrix (WDM) that has been implemented in the Concept Evaluation Workbook.

The six plausible concepts from the previous trade space analysis phase were reduced down to the three most likely candidates and inserted into the WDM. Each of the concepts was ranked against the various design characteristics. The workbook then automatically calculated the aggregate scores using the SAW method along with the relative weights generated earlier in the case study. The completed WDM and the results of the rankings are presented in Table 12 and Figure 29.

		Concept					
		A		B		C	
		Cone Crusher		*Ultrasonic Crusher*		*Laser based Crusher*	
Design Characteristic	**Relative Weight**	**Rank**	**Score**	**Rank**	**Score**	**Rank**	**Score**
Power Use	0.109	0	0.000	2	0.218	-1	-0.109
Crushing Ability	0.127	1	0.127	1	0.127	0	0.000
Sample Storage	0.073	1	0.073	-2	-0.145	0	0.000
Speed	0.055	1	0.055	1	0.055	0	0.000
Sample Integrity	0.182	1	0.182	1	0.182	1	0.182
Volume	0.091	-1	-0.091	1	0.091	2	0.182
Mass/Weight	0.145	-1	-0.145	1	0.145	2	0.291
Reliability	0.164	0	0.000	1	0.164	-2	-0.327
Initial Cost	0.036	1	0.036	0	0.000	-1	-0.036
Manufacturability	0.018	2	0.036	1	0.018	-1	-0.018
Public Safety	0.000	2	0.000	2	0.000	2	0.000
Total Score			**0.273**		**0.855**		**0.164**

Table 12: SPU Decision Matrix

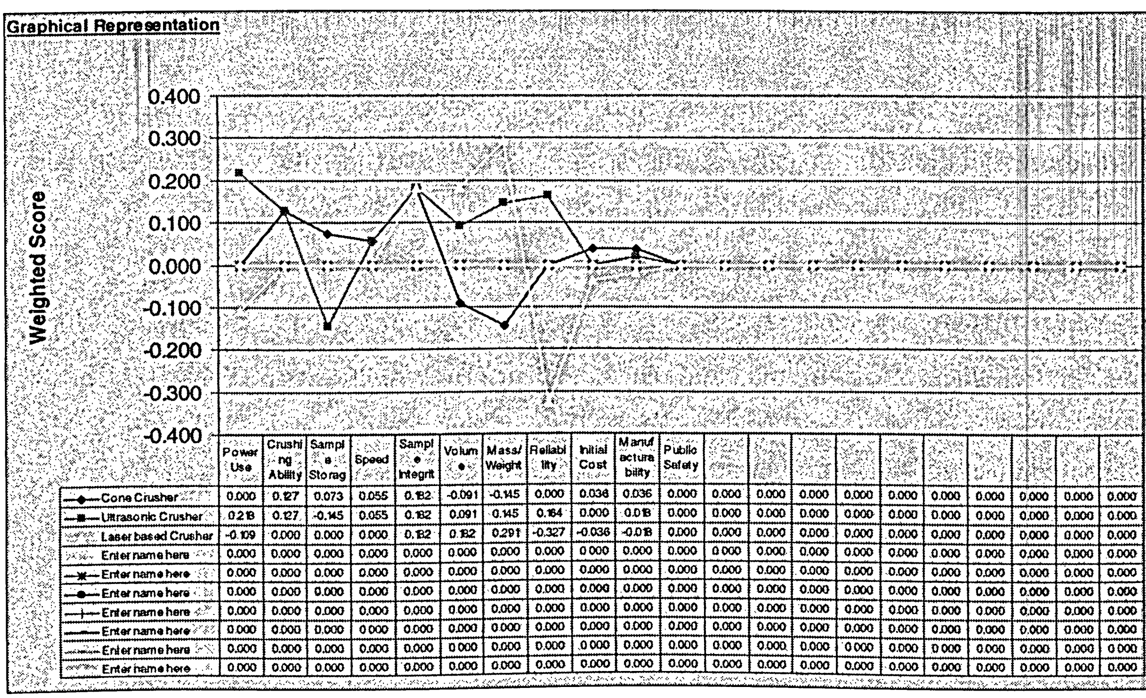


Figure 29: SPU Concept Scores (Graphical)

We can see from the above results that the Ultrasonic Crusher concept is ranked the highest with a score of 0.855. Examination of the graph in Figure 29 revels that while the other two concepts scored higher in certain areas, the Ultrasonic design was by far the most balanced of the three concepts. While it did not do everything exceptionally well, it also did not have many weaknesses, hence the higher score. This is the same result that was originally obtained in the undergraduate thesis, thus validating the results from the IDEA process.

5.1.4 Examination of Results

While the selection of the Ultrasonic Crusher helps to validate the results of the IDEA process somewhat, two other aspects of the case study are actually far more interesting.

The first item of interest is the fact that the use of the IDEA process allowed for a better understanding of the design problem when compared to the analysis undertaken for the undergraduate thesis. During the original work there was no real analysis of the design characteristics and functional requirements. This lead to a sense of confusion as to exactly what the final design would have to do and what its physical limitations were. The use of the IDEA process helped to clarify this, which not only reduced confusion but also provided a clearer definition of the design space.

The second item of note was the greatly increased number of concepts generated by using IDEA. The undergraduate thesis generated a total of three concepts, while the IDEA process identified at total of 756,000. Clearly, this is a far better exploration of the trade space than the previous method provided. This helps increase the likelihood of generating the "best" solution to a design problem.

Another advantage of a larger trade space can be seen in the six plausible concepts. Not only was the number of plausible concepts doubled over the previous results, but the diversity of the concepts was also increased. In the original study, the three concepts involved the use of an ultrasonic crusher, a laser-based system, and the mechanical cone crusher. In addition to these three alternatives, IDEA proposed concepts employing compressed air, a mechanical jaw, and a rapidly vibrating container. While these additional concepts were ultimately removed, their presence indicates that the IDEA process has done a better job of exploring the design space.

It is possible that a second iteration of the IDEA process would reveal even more alternatives or perhaps a new way of combining existing ones. The use of IDEA and the software interface has not only ensured a thorough examination of the design problem, but has also helped to increase creativity. Designers can spend more time designing and less time worrying about the process.

5.2 Case Study 2: The Speluncean Explorers

5.2.1 Problem Overview

The topic for the second case study was a problem that is often given to law students at Harvard University in order to demonstrate the ambiguity of law. The problem deals with an appeal in the murder trial of four speluncean explorers.

Set in May of 4299 in the imaginary country of Newgarth, the case deals with five men who were members of the Speluncean Society. During the exploration of a limestone cavern a landslide occurred trapping all five men in the cave. They had left their location with the society and when they failed to return a massive rescue effort was organized. The location of the cave was quite remote and the rescue effort required a massive mobilization of manpower and machinery which quickly bankrupted the society. Additional money to continue the effort was provided by the government of Newgarth. The rescue efforts were delayed several times by additional landslides and, during one of these, ten workers were killed while trying to clear the entrance of the cavern. One of the chief fears of the rescuers was that because the five men had taken relatively few provisions with them they would starve to death before they could be rescued.

On the twentieth day after the men become trapped contact was established through the use of a two-way radio that the men had taken with them. They asked how long it would take to rescue them and were informed by the foreman in charge of the operation that it would take a minimum of ten additional days, assuming that no new landslides occurred. Upon hearing this, the five men asked to speak to a doctor to whom they described their physical condition and what provisions they had remaining. They then asked the doctor for their chances of surviving an additional ten days and were informed these were almost zero. The radio then went silent for eight hours and when contact was re-established one of the men, Roger Whetmore, asked the doctors whether consuming one of the five men would allow the other four to survive. The doctors reluctantly answered that it would. Whetmore then asked if the doctors had any advice on how to make the choice and none were willing to answer. The same question was asked of a government official and a priest and both were not willing to answer. Soon after all radio contact ceased and the rescuers mistakenly believed that the batteries in the men's radio had died.

When the men were rescued twelve days later, it was discovered the Roger Whetmore was no longer present. After the men had passed out of radio contact, Whetmore had continued to push the idea of consuming one of their number so that the others could survive. Eventually the other men agreed and discussions began on how best to make the selection. After much deliberating the men decided to use a set of dice that Mr. Whetmore brought along and what all five considered a "fair" scoring system.

Immediately before beginning the rolls, Whetmore declared that he now had serious doubts as to the wisdom of the plan and was withdrawing from the arrangement. The other four explorers would not accept this and proceeded to cast the dice anyway. When it came time for Whetmore to roll, another of the men cast the dice on his behalf. Whetmore was asked if he was satisfied with the fairness of the roll and he answered yes. The roll went against him and he was killed and consumed by the remaining four members which enabled them to survive the remaining twelve days until their eventual rescue.

After retrieval from the cavern the four remaining men were taken to hospital to recover from the effects of starvation. Upon their release they were charged with murder and the subsequent court case found them guilty of this charge. According to the laws of the country of Newgarth the only acceptable sentence was death unless the verdict was overruled by the Chief Executive of the court system. An appeal was placed before the Chief Executive to find the men guilty of murder, but, because of extenuating circumstances, to commute the death sentence and replace it with imprisonment for six months. The Chief Executive now had to make his choice based upon the facts of the case.

5.2.2 Reasons for Selection

While this may seem like an unlikely choice for a case study in an engineering paper, there were several reasons for its selection. The first of these is *because* it was not an engineering problem. While the IDEA process was created to assist with engineering decision making problems, an example which had absolutely nothing to do with engineering would help to demonstrate the process' versatility. This could possibly lead to more widespread use of the IDEA process not only in the engineering industry, but in any area where structured decisions have to be made.

The second reason for selecting this problem was personal interest. While engineering is a fascinating subject with varied applications and problems, many of these problems come from the same vein. An engineer recognizes a problem and then uses scientific methods in an attempt to solve it. Invariably these types of problems are reduced down to mathematical calculations and other similar forms of analysis. It can be very beneficial to occasionally branch out and look at different areas of study. Not only does this help one to become a better-rounded individual, but it can also offer a welcome relief from doing the same types of problems over and over again.

5.2.3 Discussion of IDEA Process Steps

While the exploration of moral and ethical grey areas is somewhat analogous to the exploration of a trade space, there are obvious differences. These differences do not allow the use of all the tools included in the IDEA process. However, the solution does follow the basic steps of an engineering problem. A more detailed description of these steps can be found in Section 3 and in the previous case study.

The first phase in an engineering problem is to draft the Problem Outline and Goals, and this is beneficial in this case study as well. The completed outline and goals for the problem are presented below.

Problem Outline

To provide a verdict in the case of the State of Newgarth vs. The Speluncean Explorers given the facts as presented.

Goals

The overarching goal of this problem is to not only uphold the laws of the State of Newgarth, but also to provide some semblance of justice for the defendants. The difficulty in this case is that justice for the defendants and upholding the laws of the state are at odds with each other.

The brief reveals that the goals for the Speluncean Explorers problem are far less precise than those seen in an engineering problem. This is to be expected since this problem deals with moral and ethical issues which are very difficult to capture in precise statements. Nonetheless the outline and goals have provided an overall direction for the problem and suggest a form for the final answer.

The next step was to determine the primary and secondary objectives for the problem. While at first this may seem counterintuitive since we are not really designing anything, it is actually quite beneficial. Not only does it allow for identification of all of the stakeholders in the final verdict, but it also allowed for an analysis of the positives and negatives if each of the primary objectives were achieved. The results of these analyses are presented in Tables 13 and 14.

Primary Objective(s)	Reason(s) for Inclusion
Uphold the Statutes of the State of Newgarth	*This was included as a primary objective because laws are what make existence in a complex civilization possible and if we start to make exceptions for one set of circumstances it may open a Pandora's box of other exceptions until the statute itself no longer does what it was intended to do*
Provide Justice for the Defendants	*This is also extremely important to this case, hence its inclusion here. While upholding the letter of the law is important to society it is equally important that those laws provide a sense of justice to the populace, because it is this sense of fairness and justice that gives people confidence in their laws and their government, and without this it is very difficult to maintain a civilized society.*

Table 13: Speluncean Explorers Primary Objectives

Secondary Objective(s)	Reason(s) for Inclusion
Justice for the dead workers	*This was included here because simply executing the people that 10 workers had given their lives to save hardly seems to do justice to the sacrifice that those 10 people made. It makes their deaths meaningless*
Justice for families of the dead workers	*If the defendants are simply*

	executed the families of the dead workers will have lost their loved ones for no reason.
Justice for Speluncean Society	*They have exhausted their treasury saving these people, to have them executed will have made that a waste of resources*
Society in general wants to see the men pardoned	*Polls of random citizens in Newgarth show that an overwhelming majority of them (90%) want to see the men pardoned*
Justice for families of the defendants	*The families of the defendants just got their loved-ones back after thinking that they were irretrievably lost, now they are going to be taken away again*
Keep the spirit of the law	*It is always a difficult line to tread, but it is possible to not break the exact words of a law while breaking its spirit. Taking into account* ***why*** *a law was created is just as important as its exact wording*

Table 14: Speluncean Explorers Secondary Objectives

With the primary and secondary objectives mapped out the next task was to determine the functional requirements for a verdict. Again, this may seem strange, but doing so will allow for a better understanding of what exactly a verdict must accomplish and how it will accomplish those goals. In the case of the Speluncean Explorers problem, both the primary and secondary functions are the same since a verdict must do the same thing in order to meet both the primary and secondary objectives. The functional requirements presented in Table 15.

Functional Requirements	Reason(s) for Inclusion
A guilty or not guilty decision	*This is essentially the heart of the matter. For any verdict to be at all useful it must include a decision on whether or not the defendants are guilty*
A sentence	*If the defendants are found to be guilty then the verdict must also include a sentence for it to be deemed useful*

Table 15: Speluncean Explorers Functional Requirements

The tasks that any verdict must accomplish are to render a guilty or not guilty decision and then to assign some kind of punishment or payment depending upon that decision. Any verdict *must* accomplish both of these tasks, so the potential solutions have to meet these criteria.

Since this case study does not use all of the tools in the IDEA process, it was possible to skip directly to the concept generation step. The nature of the problem did not lend itself to a trade space analysis approach, so the ideation option was used instead. Because there was only a single individual conducting the analysis and there were no design characteristics to speak of, there was little point in generating concepts optimized for a single characteristic. The only tool of any real use in the ideation option was the brainstorming template at the end of the process. This was used to generate the three options presented in Figure 30

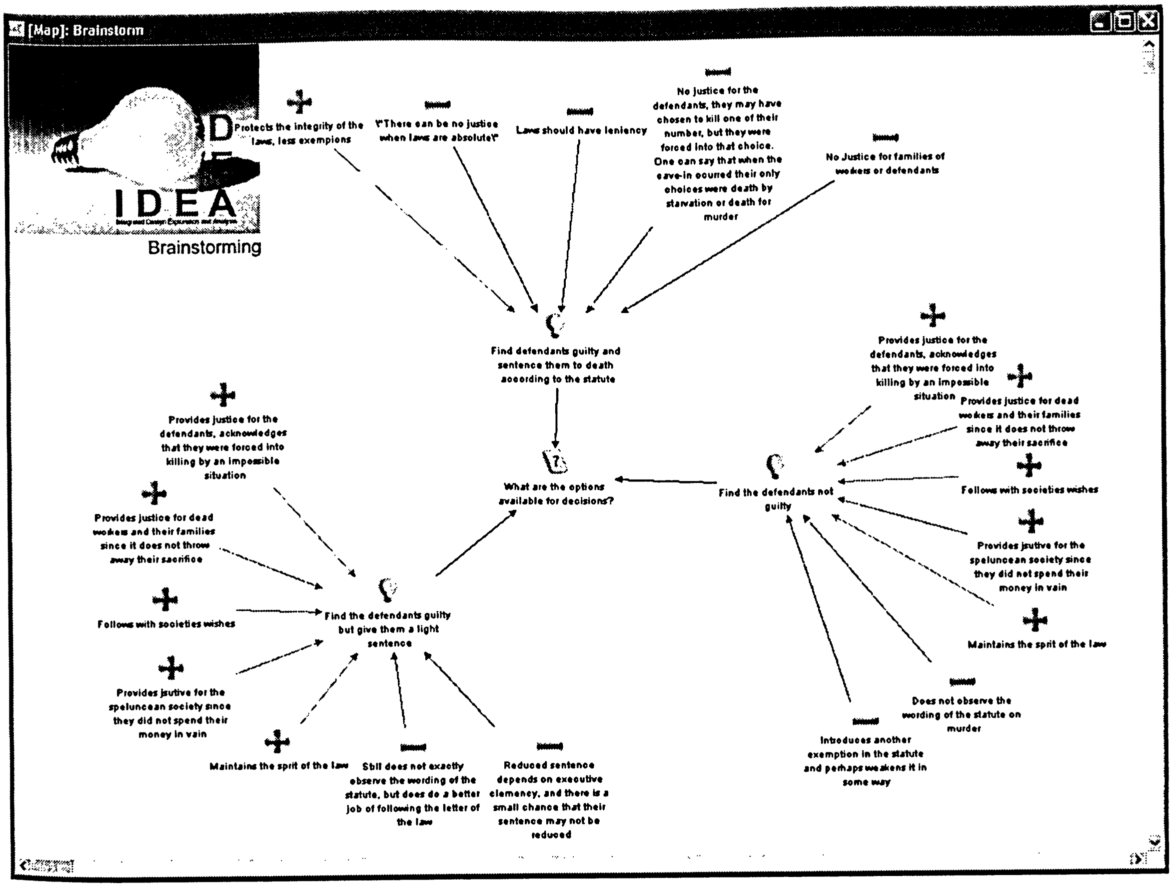


Figure 30: Speluncean Explorers Possible Verdicts

The first option is to simply uphold the original verdict as determined by the court. While this verdict does uphold the letter of the law, executing the men lacks the compassion that many individuals dcsire from the government. It does not recognize the fact that the men did not kill their companion out of free choice, nor does it recognize the wishes of over 90% of the citizens of Newgarth [22]. The basic role of government and courts is to abide by the wishes of their population, and the execution of the men would go against that trust.

The second choice would be to summarily dismiss the case and find the men innocent of all wrongdoing. While this would please the general public, it would also violate the statute upon which the law is based. The fear is that this case could be interpreted as yet another exception to the rule, thus weakening statute and raising questions as to its validity.

The final choice is a compromise between the other extremes. In this verdict the men would still be found guilty of murder, but instead of being executed their sentence is reduced to six months in prison or something similar. This choice is attractive because it tends to make everybody happy. The general population will approve because the court has recognized that the men really had no choice in killing their companion, while the court will be satisfied since it upholds the letter of the law while at the same time acknowledging its spirit.

Normally there would be a formal selection process using a decision matrix to select the "winning" verdict; however, it was not implemented in this problem for several reasons. The first is that there were no design characteristics specified for the problem and therefore no standard baseline to compare the verdicts against. The second issue with the use of a decision matrix is more of a moral dilemma. The decision on what verdict to render is literally a life and death decision and such a decision cannot be judged on a mathematical basis. Instead, it must rely on human intuition and understanding. In the end, was decided that the third option was the best choice since the compromises satisfy all parties and allow the men to live.

5.2.4 Examination of Results

One of the key reasons that this problem was selected as a case study was to demonstrate the versatility of the IDEA process. Even through all of the tools were not used in finding a verdict, IDEA assisted with organizing all of the available information. In addition, because the IDEA process forces decision makers to look at a problem in a piecewise fashion it was easier to identify all of the stakeholders in the final verdict.

At first glance, it appears that only the four men on trial have any real interest in the final outcome; however, the use of the IDEA process identifies several other groups which may not have received any attention in a more cursory analysis. In some ways the reaction of these groups to the verdict is more important than that of the accused individuals since it would be these groups who would feel the lasting repercussions of any decision. In the case of a death sentence the men would be executed and their part in this case would then be over. Many of these other groups however, would continue to feel the repercussions of that loss for many years. The use of IDEA has ensured that not only are these groups identified, but that their points of view are respected and reflected in the final decision as well. While IDEA is definitely geared towards solving engineering and design problems, it can find use in almost any situation where organized decisions are required.

5.3 Case Study 3: Design of the PowerPC 603 Processor

5.3.1 Problem Overview

The third case study deals with the design of the PowerPC 603 microprocessor. Released in late 1993, the PowerPC 603 was jointly developed by Motorola, IBM, and Apple and was designed to build upon the technologies which had debuted in the PowerPC 601 a year earlier. The microprocessor market of the early 1990's was a far different landscape than it is today. While today's market is characterised by fierce competition between Intel and AMD, the market at this time was a battle between Intel/Microsoft and Apple.

In 1993 Intel had just released the original Pentium microprocessor. This processor offered vastly improved performance over its predecessors, however even at this juncture Intel was already encountering design difficulties due to its use of the decade-old x86 architecture. The original Pentium was notoriously power-hungry and users of many of the first Pentium-based systems complained of overheating problems.

On the other side of the market, Apple Computer had seen its market-share steadily eroded since the release of the IBM PC in 1981. In an attempt to reverse this trend the executives at Apple decided that they would create a new home computer which would couple the ease of use of a Macintosh with the highest performance ever offered in a home computer. To achieve these goals Apple contracted Motorola and IBM to design a totally new processor for the new range of computers. By forgoing support for legacy applications, the designers believed that they could avoid many of the problems that plagued existing processors while still achieving industry-leading levels of performance.

The result of this collaboration was the PowerPC 601 processor, the first processor to implement what was known as the PowerPC Architecture. While the PowerPC 601 met with great critical acclaim, it did have a number of problems. One of the most pressing was that from the outset the PowerPC was designed to be the highest performing processor available. Yet at best the 601 only equalled the performance of the Pentium. The caused many potential costumers to forgo purchasing the new computers since it would force them to buy entirely new software and offered no tangible performance advantages in return.

Additionally, while the PowerPC 601 did indeed use less power than the Pentium, the power draw still precluded the use of the new processor in notebook computers. Further limiting potential markets for the new chip. Thus the main design targets design of the PowerPC 603 were set; it would have to achieve workstation performance (a SPECInt92 value of 75 [23]) while operating at notebook power levels.

5.3.2 Reasons for Selection

There were a number of reasons for the selection of this problem. One of the most important was that there was a great amount of information available on the design of this processor. Much of the design of the PowerPC 603 was accomplished using an in-house microprocessor simulation program known as BRAT [24]. The design team could enter potential parameters for the processor into the simulator and it would return data on the theoretical performance along with a host of additional data such as cache efficiency and hit rates.

Much of this data has been tabulated in the many papers written about the design of the PowerPC 603, allowing for accurate performance estimation of different processor configurations. This performance data, when coupled with results that the actual design team obtained for various processor configurations, allows IDEA to perform a meaningful suitability analysis for a variety of concepts. In addition, the reasons behind the original design decisions were also available making it possible to compare those decisions with those mandated by the IDEA process; certainly an intriguing proposition and another reason to select this problem for the third case study.

Finally, the PowerPC 603 design problem was selected because it could be used as a further demonstration of IDEA's versatility. The first case study dealt with a problem which focused mostly on mechanical engineering while the second case study did not deal with engineering at all. The successful completion of the PowerPC 603 case study would demonstrate that the IDEA process was not limited to mechanical engineering but could theoretically be used in all engineering disciplines.

5.3.3 Discussion of IDEA Process Steps

Because the IDEA process steps have previously been covered in detail, this section will simply present the results of this case study along with some discussion of more interesting observations. A detailed discussion of this case study is contained in chapters four through eight of the IDEA Software Interface Users Guide which can be found in Appendix A.

As usual, the IDEA process starts with the definition of the Problem Outline and Goals for the project. These are presented below.

Problem Outline

The design of a PC microprocessor that delivers workstation performance levels at notebook power levels.

Goals

- *Workstation performance levels (circa. 1993)*
- *Notebook level power consumption (circa. 1993)*
- *Minimize cost*
- *Minimize die size*
- *Minimize development schedule and budget*
- *Balance between power consumption, heat dissipation, and performance*

The next step was to determine the primary and secondary objectives for the design. This was accomplished in the usual manner and the results are shown in Tables 16 and 17.

Primary Objective(s)	Reason(s) for Inclusion
Deliver workstation performance levels	*This objective was included on this list because it helps to make the processor very desirable from a customer standpoint. It is never possible to have enough computing power and creating a powerful processor makes it a far more versatile part which can be used in many different situations. Western culture also perceives things that are more powerful as better and a powerful processor will increase its marketing appeal.*
Operate on notebook power levels	*Even as early at the early 1990's designers were becoming very aware of the power and heat problems that current CPUs were creating, not to mention the problems that future CPUs were predicted to create (history has proven that their estimates were not excessive, in fact if anything they were too conservative). Thus, if we put ourselves in the role of a designer in the early 90's operating on notebook power levels would be a primary concern because it would not*

	only increase the types of machines that the processor could be used in, but it would also go a long way to combating the power and heat problems which were beginning to rear their heads. Also any advances made on power saving techniques could be used in future processor lines.
Maintain backwards compatibility with PowerPC 601 code base	*This is an important requirement since it allows customers to use their existing applications and operating systems on the new processor. This prevents the added expense of having to buy entirely new software when the new processor is released. Customers can continue to do what they are doing, only it will now happen much faster; adding value to the product.*

Table 16: PowerPC 603 Primary Objectives

Secondary Objective(s)	Reason(s) for Inclusion
Low cost	*This is actually a fairly self-explanatory objective in that almost all consumer product aim to reduce costs in order to have a competitive price when the product comes to market. By having a lower cost the product carries far more appeal on the market than a product which is prohibitively expensive*
Minimize development time and costs	*This objective directly links to reducing product costs. By minimizing development time and budget we directly lower the costs of the entire project and these savings can be passed on to the consumer. At the same time having a faster development timeline also creates the possibility of*

	beating competition to the market and gaining an advantage.
Environmentally friendly	*This requirement not only deals with the actual product itself, but also with the methods used in its manufacture. Not only is being environmentally friendly important from a public relations standpoint, it is also an important aim in and of itself.*

Table 17: PowerPC 603 Secondary Objectives

The next task was to determine the design characteristics for the problem. While this case study used the standardized design characteristics list, it also made use of the electronic interview form built into the IDEA interface. This form allows designers to interview potential customers and determine what they feel would be the most important aspects for the product in question. In this case the author interviewed an individual who has used computers for several years but has little technical knowledge of how exactly a computer works. This is the perfect type of candidate to be interviewed since they know enough to tell designers what they do and do not like, but lack the technical knowledge which might sway their responses. The results of this interview are presented below.

Electronic Interview Form

Project: PowerPC 603 Case Study
Date: January 27th, 2006
Name of Interviewer: Andrew Masur
Individual Interviewed: Alexander Masur

Stage 1: Initial Seed Questions

1.) Please list what you feel would be the most important characteristics of the proposed product?

Performance, reliability, heat generated, size, fans needed, power use

2.) Which of the above characteristics first comes to mind as the most important and why?

Reliability, because it's better to have a slow but reliable processor rather than one that gives constant problems.

3.) Why is each of the above characteristics important to the design of the product?

Performance --> Does not have to be bleeding-edge, as long as the speed is adequate to do what I need it to do in a reasonable amount of time.
Heat Generated --> Lower heat increases the reliability of the processor.
Size --> The smaller the size of the processor, the smaller the overall size of the system. This increases the portability of the unit.
No. of Fans Needed --> Fewer fans are more desireable since there is less parts that can fail. It can also be quieter.
Power Use --> Lower power use brings energy savings in terms of dollars and in terms of non-renewable resources.

4.) How could similar products that you use today be improved?

I don't know.

Stage 2: Follow-up Questions

1.) If you were the design of a processor which characteristic would you sacrifice first, ie. which is the least important to you?

I would sacrifice size since in the home market the size of the processor really does not matter within limits.

The interview has provided some useful information on what design characteristics are important to the final product. In particular, the issue raised about the number of fans required to cool the processor was quite novel since the brainstorming sessions did not generate this characteristic. The information from the interview was added to the characteristics generated using brainstorming and the standardized list to generate the following overall list.

- Performance
- Cache Efficiency

- Power Use
- Heat Generation
- Die Size
- Complexity
- Reliability
- Initial Cost
- Design Cost
- Backwards Comp.
- Perceived Value

The functional requirements were generated in the usual manner and are listed in Table 18 below.

Functional Requirements	Reason(s) for Inclusion
Process PowerPC 601 Code	*This is an important requirement because it will help increase the speed of adoption of the new processor. Because the 603 will be able to process the same software as the 601 potential customers will be more willing to use it since their existing applications will continue to work, only faster.*
Deliver workstation (circa. 1993) performance levels	*Improving performance levels not only improves the value to the customer, but also increases the desirability of the processor.*
Operate on notebook (circa. 1993) power levels	*The ability to operate at low power levels not only increases the environmental friendliness of the processor, but also lowers it overall heat output which allows it to be used in more situations and many different types of computers.*
Work with existing infrastructure	*This requirement covers everything from being able to use existing types of memory, to the*

	ability to be socketed into existing motherboards, to the ability to use existing compilers. The ability to work with existing infrastructure will lower the cost of adopting the processor and may lead to greater adoption by consumers.
Sell for as low a cost as possible	*Again, this increases the desirability of the processor from a consumer standpoint.*
Use existing technology where feasible	*This requirement will help to lower the design cost of the processor. Instead of reinventing the wheel, designers can look at what worked in the design of the PowerPC 601 and determine ways to leverage that technology to improve the current product.*
Generate a minimum of heat	*As before, this not only makes the processor more environmentally friendly but also allows it to be used in a wider range of situations.*
Use as few raw materials as possible	*This will not only make the production of the processor more environmentally friendly, but will also make production cheaper and more efficient, lowering the cost of the final silicon.*
Ability to operate under increased temperatures	*This is a requirement for most IC's that are used in a computer environment. Because of the increased temperatures found within a computer components used within it must be able to tolerate increased temperature.*

Not affected by RFI emissions	*The interior of a computer is subject to RFI emissions from most of the components. To successfully operate in such an environment the individual components must not be affected by these emissions.*

Table 18: PowerPC 603 Functional Requirements

Calculation of the relative importance of the design characteristics was accomplished in the usual manner through the use of the concept evaluation workbook. The relative weights of the characteristics are shown in Figure 31.

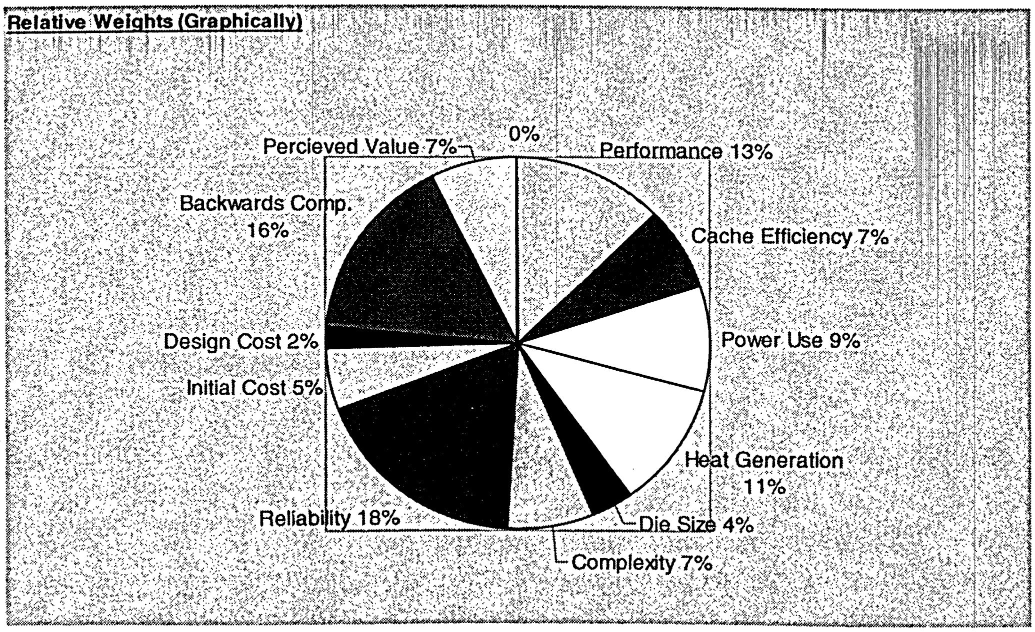


Figure 31: PowerPC 603 Relative Importance

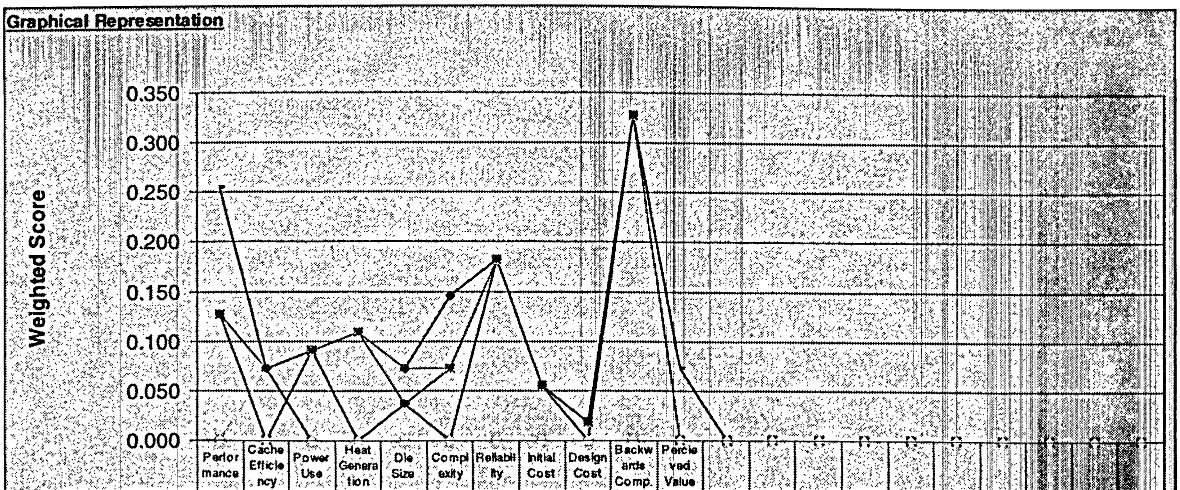
Concept generation was carried out using trade space analysis. The plausible concepts at the conclusion of this process are shown in Table 19.

Brief Description of Concept	Image (If available)
Concept A: *601 Compatible 64/32-bit + Split 8K 2-way Cache + 4 Completion Units + 2 Instructions per Cycle + 2 Execution Units/1 Reservation Station*	*N/A*
Concept B: *601 Compatible 64/32-bit + Split 8K 4-way Cache + 4 Completion Units + 2 Instructions per Cycle + 2 Execution Units/1 Reservation Station*	*N/A*
Concept C: *601 Compatible 64/32-bit + Split 8K 2-way Cache + 5 Completion Units + 2 Instructions per Cycle + 2 Execution Units/1 Reservation Station*	*N/A*
Concept D: *601 Compatible 64/32-bit + Split 8K 4-way Cache + 5 Completion Units + 2 Instructions per Cycle + 2 Execution Units/1 Reservation Station*	*N/A*
Concept E: *601 Compatible 64/32-bit + Split 8K 2-way Cache + 6 Completion Units + 2 Instructions per Cycle + 2 Execution Units/1 Reservation Station*	*N/A*
Concept F: *601 Compatible 64/32-bit + Split 8K 4-way Cache + 6 Completion Units + 2 Instructions per Cycle + 2 Execution Units/1 Reservation Station*	*N/A*
Concept G: *601 Compatible 64/32-bit + Split 8K 4-way Cache + 5 Completion Units + 2 Instructions per Cycle/1 Retire + 2 Execution Units/1 Reservation Station*	*N/A*
Concept H: *601 Compatible 64/32-bit +*	*N/A*

Split 8K 4-way Cache + 5 Completion Units + 2 Instructions per Cycle/1 Retire + 3 Execution Units/1 Reservation Station

Table 19: PowerPC 603 Plausible Concepts

These concepts were ranked using the weighted decision matrix in the Concept Evaluation Workbook. Concept H was found to have the highest combined score at 1.091. This concept is very similar to the final design of the actual PowerPC 603 processor and is just ahead of concept G which would also have been a suitable candidate if die space was at a premium. The graphical representation of all of the concept scores is presented in Figure 32.



	Performance	Cache Efficiency	Power Use	Heat Generation	Die Size	Complexity	Reliability	Initial Cost	Design Cost	Backwards Comp.	Percieved Value											
Concept A	0.000	0.000	0.091	0.109	0.073	0.145	0.182	0.055	0.018	0.327	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Concept B	0.000	0.073	0.091	0.109	0.073	0.073	0.182	0.055	0.018	0.327	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Concept C	0.000	0.000	0.091	0.109	0.036	0.073	0.182	0.055	0.018	0.327	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Concept D	0.000	0.073	0.091	0.109	0.036	0.073	0.182	0.055	0.018	0.327	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Concept E	0.127	0.000	0.091	0.109	0.036	0.073	0.182	0.055	0.018	0.327	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Concept F	0.127	0.073	0.000	0.000	0.036	0.000	0.182	0.055	0.018	0.327	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Concept G	0.127	0.073	0.091	0.109	0.036	0.073	0.182	0.055	0.000	0.327	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Concept H	0.255	0.073	0.091	0.000	0.036	0.000	0.182	0.055	0.000	0.327	0.073	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Enter name here	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Enter name here	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Figure 32: PowerPC 603 Concept Scores

5.3.4 Examination of Results

While this case study was very similar in structure to the Martian SPU example, it did contain a number of interesting items. The first was that this case study was the first to make use of the electronic interview form. The use of this form not only confirmed many of the findings of the original design team, but also showed that customer interviews can result in unexpected information. In the sample interview the individual who was being questioned mentioned the fact that the number of fans required to cool the processor was important. They felt that by reducing the number of fans there would be a corresponding decrease in the noise level of the computer which they found desirable. This observation had not been generated in any of the brainstorming sessions and reinforced the fact that reducing power use could also lead to quieter computers which would be a very desirable characteristic.

The second item of note was the fact that it was performed on what is essentially an electrical engineering problem. Up until this point, many of the engineering examples in this report were based upon mechanical engineering. The use of an electrical engineering problem in the final case study reinforces the fact that the IDEA process is a multi-disciplinary engineering tool. This case study could just as easily have involved a problem in chemical or industrial engineering and the basic steps of the process would remain the same. This is important since the entire point of IDEA is to create a standardized concept design process that almost anyone can use in a wide variety of situations.

The third and most interesting item of note can be found in the conclusions derived from this study. Concept H, the concept which was eventually selected as the "winner", has a very similar design to the actual PowerPC 603. This helps to validate the results generated by IDEA since that design was deemed satisfactory enough to be released into the marketplace. Of even greater interest is the creation of concept G. While the aggregate score for this concept was not as high as that for concept H, it was quite close. The configuration of this design was quite different than concept H, it was optimized for low power use while concept H was optimized more for performance, however the fact that it scored very close to the "winning" concept means that it could possibly be a legitimate alternative. At the very least this type of result would merit further study of both concepts by the design team either with IDEA or through more detailed procedures.

6.0 Conclusions

6.1 IDEA and the Three Pillars of Concept design

The beginning of this report described the three pillars of a concept design process:

1.) Accurately determine the needs and desires of the customer.
2.) Thoroughly explore all possible options for the final design in order to maximize the possibility of finding the "best design".
3.) Select the "best" concept in a rational manner which engenders confidence in the results.

The question now becomes "Does the IDEA process meet these three pillars?" To answer this let us look at each of these goals individually.

Determining exactly what potential customers want from a product is one of the most important tasks in engineering. No matter how well designed a product may be, if customers do not want it they will not buy it. This is the harsh reality of the marketplace, so determining exactly what customers want is of prime importance. In order to achieve this, the IDEA process has several tools that are used to explicitly determine customer needs.

The early modules of the IDEA process are used to determine project objectives using team meetings and brainstorming sessions. One of the best ways of determining exactly what customers want is to include them in these sessions. This ensures that their needs are reflected in these decisions. Additionally, the IDEA interface includes an electronic form that can be used to conduct interviews with potential customers. Because the form is stored electronically it is a relatively easy to involve potentially hundreds of customers in the concept design process. With such a large sample, the needs of the general public become clear, and, as was seem in the PowerPC case study, electronic interviews can sometimes provide information that the design team would otherwise overlook. In both cases, the IDEA process includes tools that ensure that it meets this first pillar of concept design.

The second goal of any concept design process is to conduct a thorough examination of the trade space in order to maximize the possibility of finding the "best" concept. Not only does the IDEA process include tools to explore trade spaces, but it also includes many tools to help define them. A study of the primary and secondary objectives, as well as the design characteristics and functional requirements, helps designers to find the boundaries of the trade space for a particular problem. This helps designers gain a better understanding of what may and may not work for that problem. All of these things help to reduce the size of the trade space, which allows designers to explore it more thoroughly and increases the chances of finding the "best" design.

When the time comes to actually explore the trade space, the IDEA process includes two different options for doing so. The trade space analysis option is a very straightforward, almost brute force approach to exploring the trade space. Designers simply generate as many concepts as they can and try to determine which the best are. While this can be time consuming, it is almost guaranteed to work due to the sheer number of concepts that are generated, as can be seen in both the first and third case studies.

The second option provided by the IDEA process is known as ideation. Unlike trade space analysis, ideation is less mechanical and more intuitive. Each designer attempts to create "ideas" that are tailored to one particular design characteristic. The designers then come together and try to fuse their individual ideas into superior combined concepts. This is not always possible, and while the ideation process can result in truly innovative concepts, potential incompatibilities between individual concepts mean that there is always a chance that no useful concepts will be found at all. In both cases, the IDEA process helps designers thoroughly explore the trade space in an attempt to locate the "best" concept.

A concept design process that generates hundreds of concepts is laudable but is essentially useless unless there is some to select rationally amongst them. The third pillar of concept design is meant to address this problem by ensuring that such a system is in place. In order to meet this third pillar the IDEA process includes several measures to help with making rational decisions.

The main tool in IDEA for selecting amongst different concepts is the weighted decision matrix (WDM). A WDM ranks concepts against a baseline design on a common set of criteria and ensures that these comparisons are meaningful and repeatable. In addition, the use of a WDM ensures that the relative importance between different design characteristics is reflected in the final selection. A well-balanced design will usually win out over one that is only good at certain tasks, no matter how exceptional at those tasks it may be. This not only ensures that the decisions of the design team are logical, but also prevents concepts with obvious weaknesses from moving on to more detailed design.

Finally, a WDM also acts as a record of the decision making process. With any engineering project, invariably the question of whether or not the design team made the right choice arises. A WDM not only records the decisions that the design team made, but also why they made them. This can be of great use when trying to justify the selection of a concept to a manager who is displeased with the progress of a project. In the rare case that a project is terminated, the weighted decision matrix can also act as a record of what went wrong so that the same mistakes are not repeated. All of these features combine to ensure that the IDEA process meets the third pillar of concept design.

6.2 Deliverables

While all of this talk about the theoretical performance of the IDEA process is necessary, the real question is "What has actually been accomplished with this thesis?" This list of deliverables briefly outlines everything that this thesis provides.

- **The IDEA process itself:** This is the real heart of the thesis and where much of the time and effort was expended.
- **This report:** For any project as large as this a report detailing the theory behind the final project is invaluable for judging its merit. In addition the report provides several case studies which can be seen as "proof" of IDEA's validity.
- **The IDEA Software Interface:** This was created to make the IDEA process easier to use which should help to increase adoption.
- **The IDEA Users Guide:** This document can be found in Appendix A and serves not only as a guide on how to install and use the interface, but also as a concise step-by-step explanation on how to conduct concept design using IDEA.

6.3 Future Work

Most engineering work is an evolutionary process; flaws are found with an existing design and are fixed in the next revision. The IDEA process itself is no different. While the current form of IDEA is quite robust, there is always room for improvement. In particular, there were three areas identified during work on this thesis that stand out as items that could be improved in future versions.

The first of these is the modelling of uncertainty in the IDEA process. Uncertainty is a fact of life for an engineer; and while IDEA is designed to help minimize uncertainty, it by no means eliminates it completely. Because uncertainty will always be present in concept design, it would be of great interest to model it in some fashion. One of the most promising avenues discovered during research for this project was the use of fuzzy numbers to model the uncertainty in design. In his paper *Imprecision in Engineering Design,* Antonsson postulates that fuzzy numbers are ideal for modelling uncertainty [25].

At its most basic, a fuzzy number is nothing more than another method for expressing a range of values. In concept design, a higher fuzzy number could be used to indicate a lower level of uncertainty while a lower fuzzy number would indicate a higher level of uncertainty. While this may seem counterintuitive at first, this type of system might be useful in a WDM. Concepts in which the design team has confidence would have a higher chance of succeeding and would thus receive a higher score. Conversely, concepts in which the design team has little confidence would receive a lower score. If these numbers could somehow be integrated into the overall score then the relative certainty of a concept would become another variable to trade off during the concept design process. A design team would now have a numerical measure of exactly how risky a more cutting edge concept would be. If the risks were not worth the gain, then the design team might ultimately choose to go with a more traditional but safer design.

The other two areas of improvement in the IDEA process both deal with the software interface. As stated earlier, the reliance on external applications to store data is one of the greatest weaknesses of the interface. One problem is that data entered into Compendium is not automatically transferred to the documents used by the external applications. This requires members of the design team to manually enter this information, increasing the bookkeeping required to use the interface. A future improvement might include a way to automatically transfer this information to the external documents.

A further improvement to this system would be to remove the reliance on external applications all together. This would not only help to ease deployment, but would also simplify the saving and loading of documents. Therefore, an extremely beneficial improvement would be to integrate these storage documents directly into Compendium and the interface itself. Removing the reliance on external applications would also help to eliminate the requirement that a user's computer have these applications installed. Many of these applications are not free, and their elimination would make the IDEA interface free to implement, increasing its appeal. All of these improvements would help to make the IDEA process far more robust and desirable. If one is going to use IDEA to improve products, it should be just as open to improvement itself. That is progress after all.

References

[1] Salustri, Dr. Filippo A.; Ryerson University; What is Design?; http://deseng.ryerson.ca/~fil/t/whatisdesign.html; (March 18th, 2006)

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

[3] NASA; Mars Exploration Rover Mission; http://mars.jpl.nasa.gov/mer (March 18th, 2006)

[4] Keeney, Ralph L.; "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

[5] Colwell, Bob; "Brainstorming, Influence, and Icebergs"; *Computer Magazine*; April 2004

[6] Westcombe, M.; Pidd, M.; Mackenzie, A.; Warren, I.; Sommerville, I.; *Problem Solving Dialogue: Cognitive Mapping and IBIS*; Lancaster University Management School; 2002

[7] Smith, Jeffery H.; *Low Cost Robotics for Space Exploration: A Probabilistic Trade Space for Biomorphic Exploration Devices*; IEEE; 2000

[8] Vargas-Hernandez, Noe; Shah, Jami J.; Smith, Steve M.; "Metrics for Measuring Ideation Effectiveness"; *Design Studies*; No. 24; 2003

[9] Price, H.W.; Carraway, S.E.; Matousek, S.E.; Staehle, R.J.; Terrile, R.J.; Wyatt, E.J.; "Pluto Express Sciencecraft System Design"; *Acta Astronautica*; Vol. 39; No. 1-4; 1996

[10] Scott, Michael J.; Antonsson, Erik K.; "Arrow's Theorem and Engineering Decision Making"; *Research in Engineering Design*; 1999

[11] Hazelrigg, G.A.; "The Implecations of Arrow's Impossibility Theorem on Approaches of Optimal Engineering Design"; *ASME Journal of Mechanical Design*; Vol. 118; 1996

[12] Dym, Clive L.; Wood, William H.; Scott, Michael J.; "Rank Ordering Engineering Designs: Pairwise Comparison Charts and Borda Counts"; *Research in Engineering Design*; No. 13; 2002

[13] Salustri, Dr. Filippo A.; Ryerson University; Pairwise Comparison; http://deed.ryerson.ca/mec222/2005/L/pwcomp.html; (March 18th, 2006)

[14] Anderson, Dale O.; *Making Engineering Design Decisions*; Louisiana Tech University; 2000

[15] Salustri, Dr. Filippo A.; Ryerson University; Decision Matrices for Concept Evaluation; http://deed.ryerson.ca/mec222/2005/L/dm.html; (March 18th, 2006)

[16] Zanakis, Stelios H. et. al.; "Multi-attribute Decision Making: A Simulation Comparison of Select Methods"; *European Journal of Operational Research*; Issue 107; 1998

[17] Salustri, Dr. Filippo A.; Ryerson University; Product Characteristic; http://deseng.ryerson.ca/xiki/View/Learning/ProductCharacteristic; (April 14th, 2006)

[18] Masur, Andrew; Salustri Dr. F. A.; Private Correspondance; February 9th 2006

[19] Petrovick, Otto; Krickl, Otto; "Traditionally Moderated versus Computer Supported Brainstorming: A Comparative Study"; *Information & Management*; Issue 27; 1994

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

[21] Masur, Andrew; *Mars Science Laboratory Sample Processing Unit Design*; Ryerson University; 2004

[22] Fuller, Lon L.; "The Case of the Speluncean Explorers"; *Harvard Law Review*; Harvard University; Vol. 62; 1949

[23] Burgess, Brad; et. al.; "The PowerPC 603 Microprocessor"; *The Making of the Power PC*; Communications of the ACM; Vol. 37; no. 6; 1994

[24] Burgess, Brad; et. al.; *The PowerPC 603 Microprocessor: Performance Analysis and Design Trade-Offs*; Motorola and IBM; 1994

[25] Antonsson, Erik K.; Otto, Kevin N.; "Imprecision in Engineering Design"; *ASME Journal of Mechanical Design*; Vol. 117(B); 1995

[26] NASA; Mars Exploration Rover Mission: Spirit Struggles to Survive the Martian Winter; http://marsrovers.jpl.nasa.gov/spotlight/20060404.html (September 2nd, 2006)

[27] NASA; Mars Exploration Rover Mission: Rock Abrasion Tool Exhibits the Deep Red Pigment of Mars; http://marsrovers.jpl.nasa.gov/gallery/press/spirit/20060721a.html (September 2nd, 2006)

[28] Uncreative Separatists; "Yet Another Boring iPod Review"; http://www.uglx.org/ipod-review; (September 3rd, 2006)

Glossary

Unconstrained Trade Space: An imaginary construct which contains all possible solutions to all possible design problems. Theoretically the unconstrained trade space is infinite in size with an infinite number of solutions.

Constrained Trade Space: An imaginary construct which contains all of the possible solutions to the current design problem. The constrained trade space is a sub-set of the unconstrained trade space whose boundaries are defined by the current design problem. Because the constrained trade space is finite in size it contains a finite number of solutions, although that number can still be quite large.

Trade Space Diagram: A simple diagram which contains all of the solutions currently discovered in a constrained trade space.

Arrows Impossibility Theorem: Also know simply as Arrow's Theorem; this theorem, originally postulated by Kenneth J. Arrow, outlines five axioms which must be followed in order to ensure a fair decision making process. However, Arrow and others have shown that it is impossible for a practical decision making process to heed all of these axioms, hence at least one must be broken for problems of reasonable size. The consequences of breaking any of these axioms vary from problem to problem.

Pairwise Comparison: A decision making technique which breaks a complex problem down into a series of simpler decisions. Decision makers can then tackle these simpler decisions one pair at a time until the more complex problem has been solved.

Drop and Revote Method: A form of pairwise comparison which discards the losing item from each decision pair. This deletion of information makes this method very sensitive to the effects of Arrow's Theorem.

Borda Count: An alternative method of pairwise comparison which does not discard information like the drop and revote method. Instead it maintains the rankings of all items that have been voted on. This makes this method less sensitive to the effects of Arrow's Theorem.

Pairwise Comparison Chart (PCC): A graphical method of conducting a Borda Count. This method retains all of the advantages of the Borda Count while at the same time being easier to use.

Weak Order: A simple ordinal ranking of alternatives. A weak order gives the order of preference between items, but contains no information as to the level of preference.

Value Function: A map of the level of preference between alternatives in a weak order. A value function not only shows the order of preference, but also shows how much one alternative is preferred over another.

Weighted Decision Matrix (WDM): A graphical decision making technique which uses a table of values to make complex engineering decisions. This method is particularly useful when making decisions amongst alternatives with multiple criteria.

Aggregation Function: A mathematical function that is used in consort with the WDM to determine the rankings amongst the alternatives contained in the WDM.

Primary Objectives: Goals that a design must achieve in order to be considered successful. Primary objectives outline the needs of the customer.

Secondary Objectives: Goals that are meant to add value to a design for potential customers. While secondary objectives are not required, they are desirable goals to try to achieve. Secondary objectives outline the desires of the customer.

Design Characteristics: These are things which describe what the final product has to be, whether it has to be light or heavy, fast or slow, etc.

Functional Requirements: These are meant to define what a design has to do in order to meet the primary and secondary objectives.

Ideation: A method of generating concepts which attempts to combine several partial concepts known as *ideas* into superior combined concepts. Each idea is a partial concept that has been optimized for a single design characteristic.

Appendix A: IDEA Software User's Guide

This Appendix contains a copy of the User's Guide that was written for the IDEA software interface. This is intended to be a separate document, and was written for general users. Because of this the document is written in a different style than an academic paper, and the numbering has been restarted to reflect the table of contents contained herein.

Users Guide

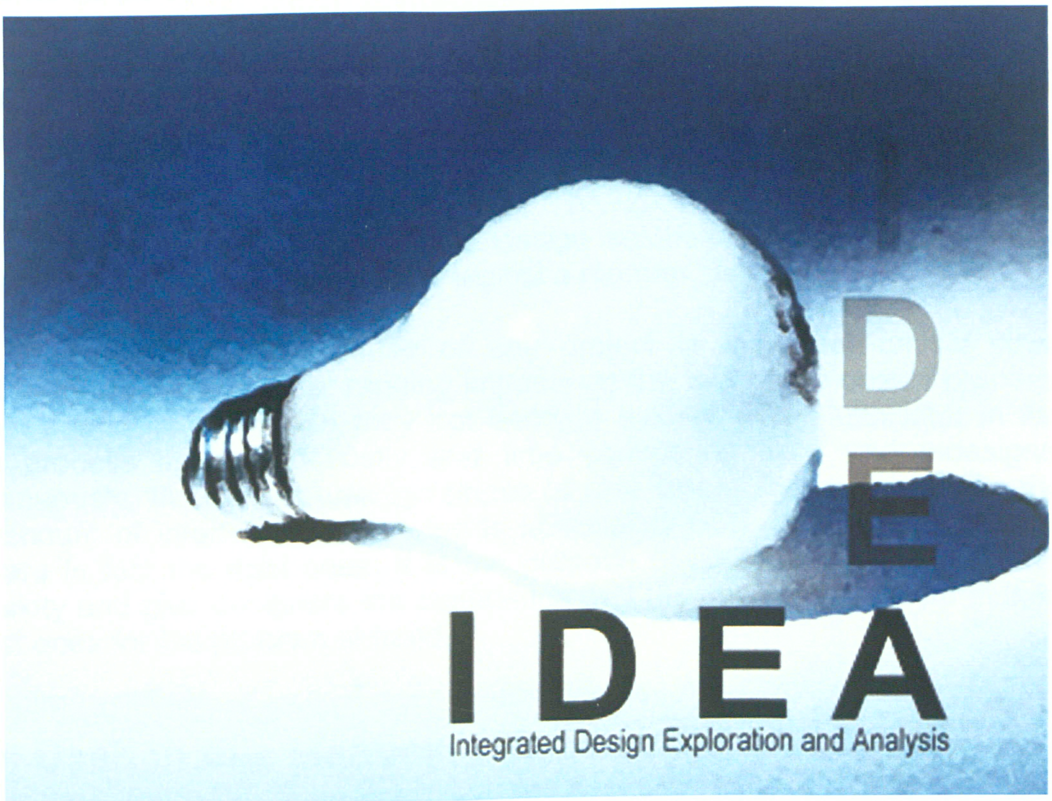
Installation, Basic Use and Maintenance
IDEA Interface Tool
Version 1.0

Andrew Masur, 2006

Table of Contents

Part 1

Installation and Setup



Chapter 1

Welcome

Welcome to the Integrated Design Exploration and Analysis (IDEA) Interface Tool. This software can streamline concept design by augmenting human creativity and decision-making using the IDEA concept design process. This is done by identifying and automating the mechanical parts of this process leaving designs to do the kind of design work that computers just cannot do.

What is IDEA?

Simply put, IDEA is a system designed to help people make decisions by following a series of logical and standardized steps. The fact that the steps are standardized instills a high degree of traceability in the solutions that IDEA provides. It is this logical framework and traceability that makes IDEA well suited to concept design problems in engineering. Through the use of IDEA designers are able to better map out alternative design solutions and are then able to choose amongst these solutions in as logical a manner as possible.

The concept design phase of any project is the most critical since decisions made here have far ranging impacts on the rest of the design process. Mistakes made in this phase may not become evident until much later in the design process leading to costly and time consuming fixes and redesigns. Simultaneously, the concept design phase is also where the designer has the least amount of information which lends uncertainty over whether the choices made are in fact the right ones. It is the purpose of IDEA to help remove this uncertainty and give designers the confidence that the choices made are indeed the best ones for the problem at hand.

Icons Used in this Manual

Throughout this manual you will find boxes with several icons denoting important information, the following table summarizes their meanings.

Icon	Description
	**Frequently Asked Questions:** When you see this icon you'll find answers to commonly asked questions or expansions on subjects covered in the main text.
	**Watch Out:** This icon warns that the box contains important information or points out common mistakes. If you see this pay special attention to the information contained in that box to stop potential problems before they happen.
	**Tips:** These boxes contain optional information for streamlining the use of the software, techniques to stimulate thinking, examples to clarify certain concepts, or other pertinent information.
	**Further Information:** Links and references to further information about a particular subject.

Chapter 2

Installation

This chapter will outline the procedure for installing the IDEA Interface Tool and configuring it for use on your computer.

System Requirements

The IDEA Interface Tool (IIT) is built upon Compendium by the Compendium Institute, an open source application designed to assist with recording the information generated during meetings. The system requirements for Compendium are as follows.

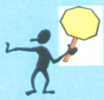
CPU	Intel Pentium III 750MHz AMD Athlon 500MHz
Memory	256MB
Operating System	Microsoft Windows 2000/XP; X Mac OSX 10.3
Other	Java Runtime Environment 1.5 (included with the Compendium installer); JRE 1.4.1_02 (for OSX)

Further information about Compendium can be found at www.compendiuminstitute.org

IDEA was made with Compendium 1.4 on a Windows XP platform. While it is likely that IDEA will work with future versions of Compendium it is not guaranteed. There are also versions of Compendium available for Mac OSX 10.3 and Linux, but the use of the IIT with these versions is not officially supported although proper operation is possible.

Because of certain limitations in the current version of Compendium it was necessary to use several external applications in order to save documentation in the IDEA templates. It is possible that future versions of IDEA will overcome this requirement. This, coupled with the highly graphical nature of the IDEA interface imposes the following system requirements.

Display	1280x1024 with 256 colours
Hard Disk Space	60MB (includes Compendium install)
Support Applications	Microsoft Word (2000/XP/Mac); OpenOffice Writer 2 Microsoft Excel (2000/XP/Mac); OpenOffice Calc 2 Microsoft Access (2000/XP/Mac); OpenOffice Base 2



Although IDEA will work without the support applications, much of its functionality will be disabled, therefore it is **HIGHLY** recommended that these applications be installed.

IDEA may be run at a resolution lower than 1280x1024, however more scrolling will be required when viewing the interface.

Installation

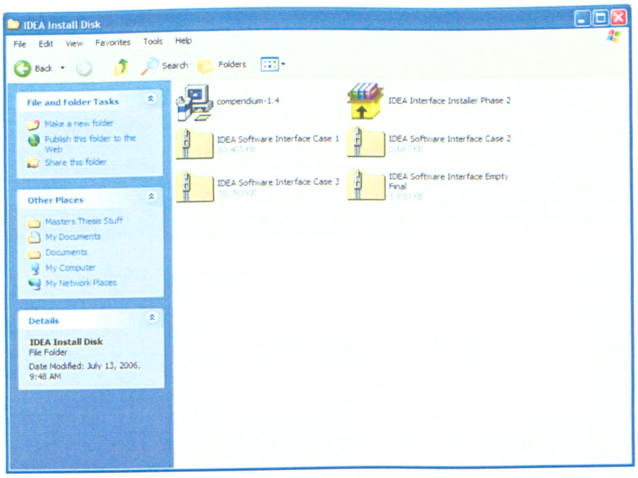
The installation for IDEA can be split into three distinct phases: the installation of Compendium, followed by the installation of IDEA itself, and finally importing the interface into Compendium.

Phase 1: Installation of Compendium

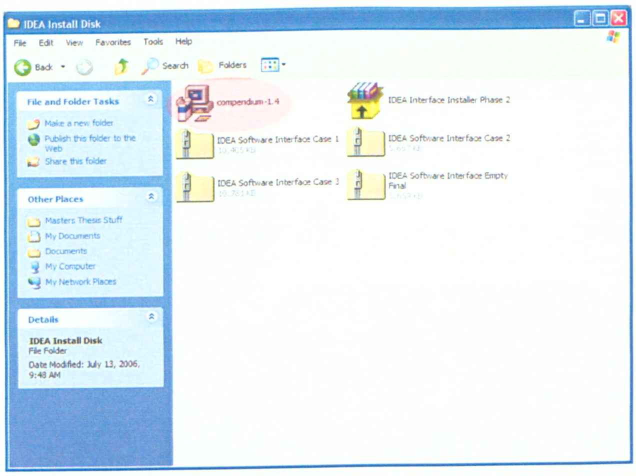
Further installation information and troubleshooting for Compendium can be found at www.compendiuminstitute.org

1.) Insert the IDEA Installation CD in your CD-ROM drive.

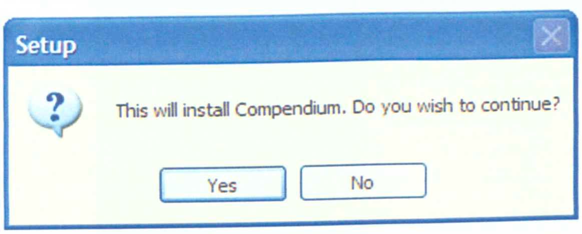
2.) Open **My Computer** and double-click on the icon representing the CD-ROM drive that you inserted the Installation CD into, the following window should appear.



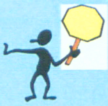
3.) Double-click on the icon labelled **compendium-1.4**

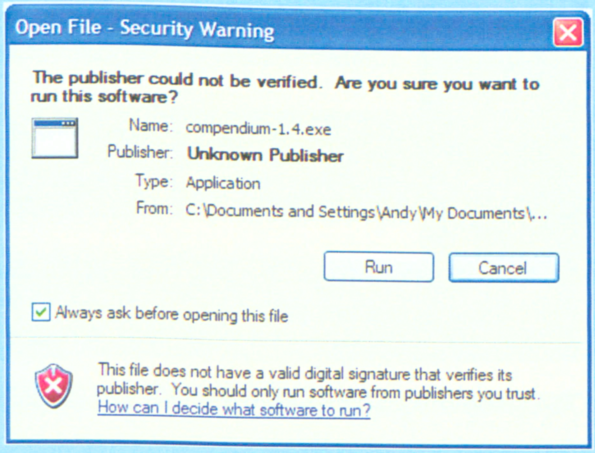


4.) A message will appear asking if you would like to continue installing Compendium, click **Yes**.



Users of Windows XP with Service Pack 2 installed may receive the following **Security Warning**, if so click on **Run** to continue the installation.

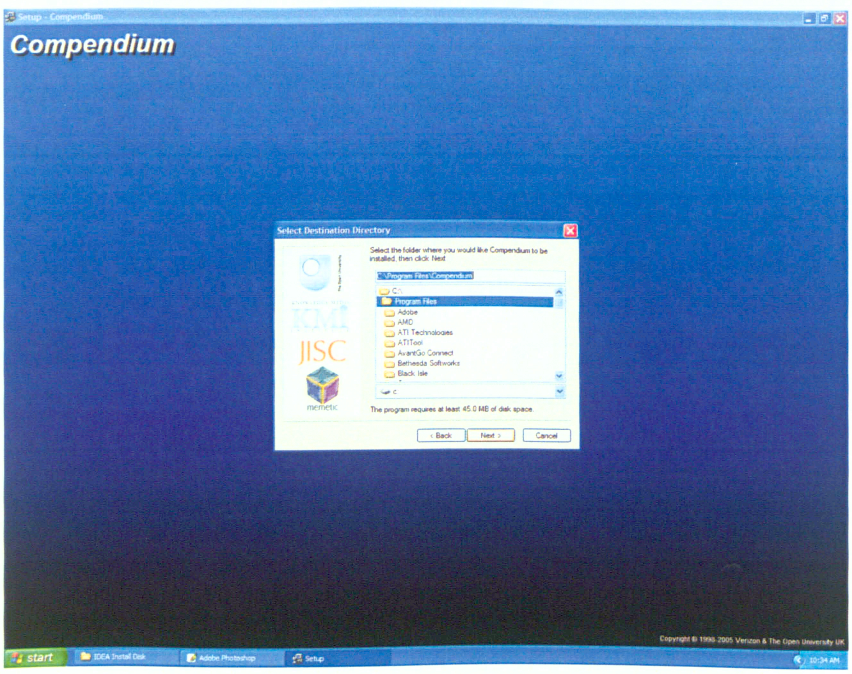


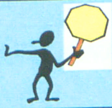


5.) A welcome screen will appear, click on **Next** to continue the installation.

6.) Click on **Yes** to accept the license agreement and continue the installation.

7.) The installer will now ask you for the installation path for Compendium; it is recommended that you accept the default. Click on **Next** to continue.

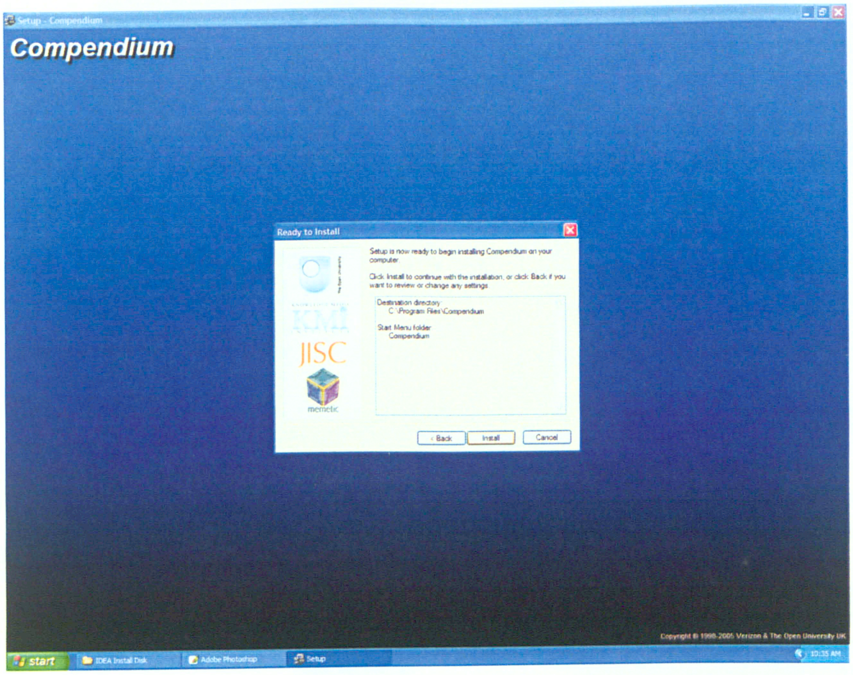




If you choose to install Compendium to a drive other than the default be sure to make note of it for the second phase of the installation.

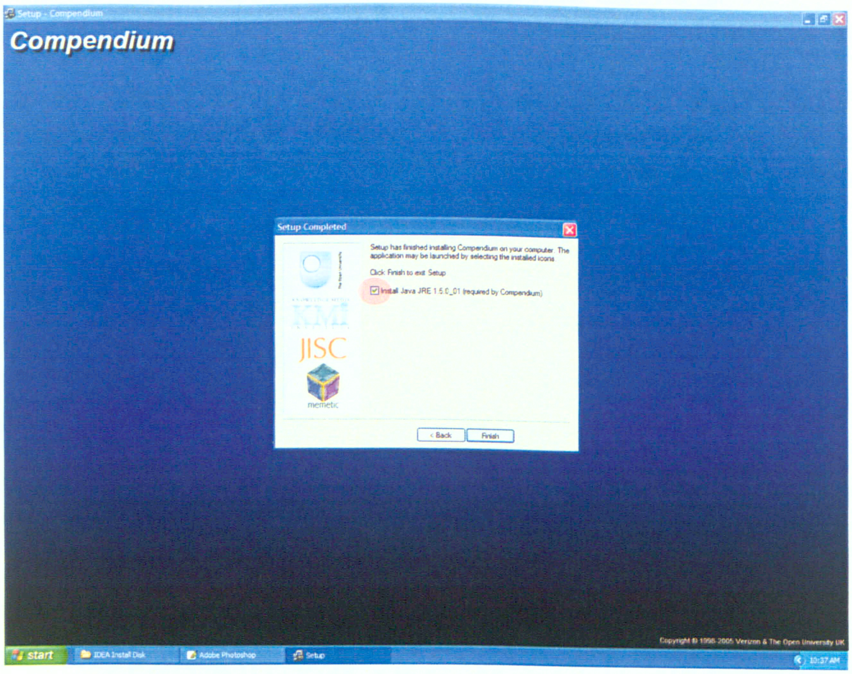
8.) The next screen will ask where you would like the icons for Compendium created, accept the default location and click **Next** to continue.

9.) The installer will now present a summary of the selected installation options. If these are correct click on **Install** to begin installation, if not click on **Back** to change these options.

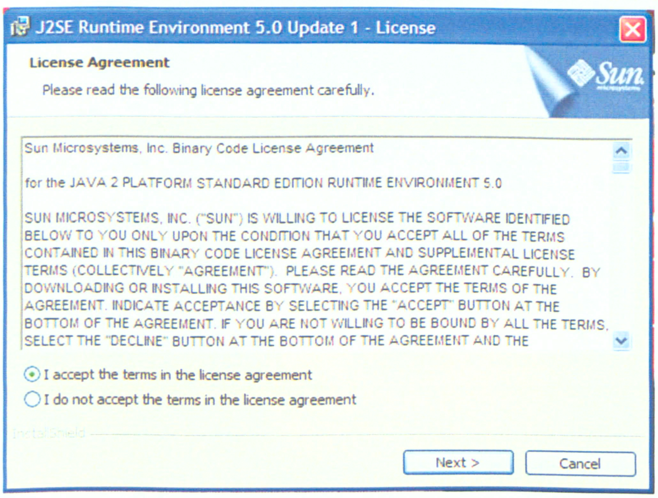


10.) After the installer has finished copying files it will present you with information about the installation of Java, click **Next** to skip past this screen.

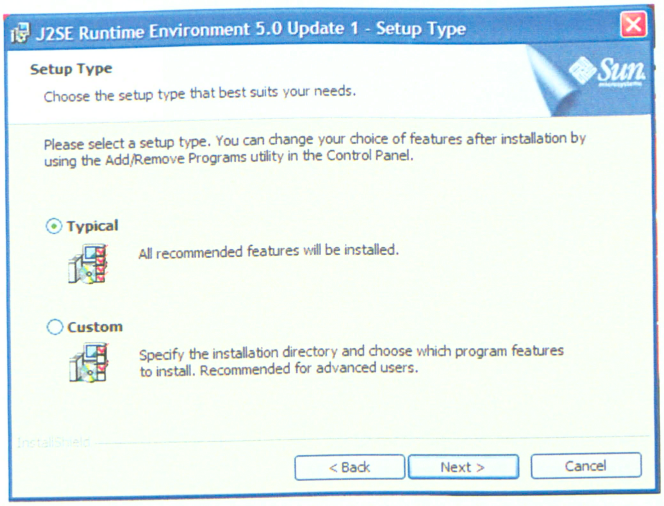
11.) You will now be presented with the final screen of the Compendium installer. Make sure that the Java installation checkbox has a checkmark in it and click on **Finish** to complete the installation of Compendium and start the installation of Java.

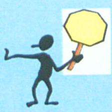


12.) You will now be presented will the license agreement screen for Java, **accept** the agreement and click on **Next**.



13.) On the next screen select **Typical** and click on **Next** to complete installation of Java.



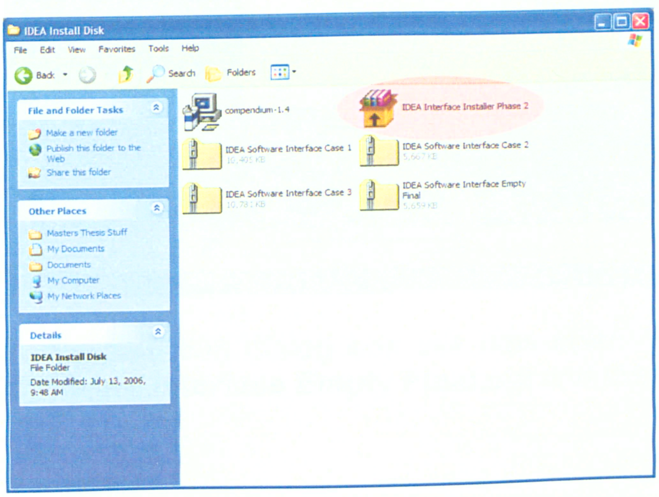


If you have installed Compendium to a drive other than the default, select **Custom** and change the drive in the installation path to match the drive which you installed Compendium to. Leave the rest of the items at their default values, and click on **Next** to move through the rest of the installation.

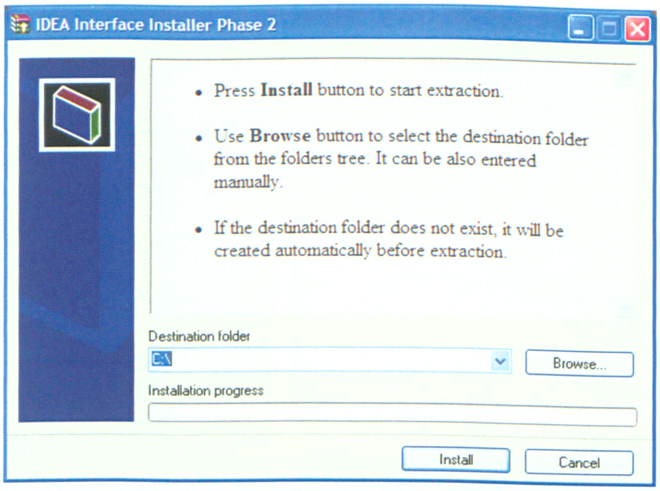
Phase 2: Installation of IDEA

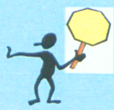
1.) Return to the IDEA Installation CD window.

2.) Double-click on the icon labelled **IDEA Interface Installer Phase 2**.



3.) The installer for the IDEA Interface will now appear asking you to which drive you would like to install the interface to. It is highly recommended that you keep the default drive. Simply click on **Install** to complete the installation of the interface.



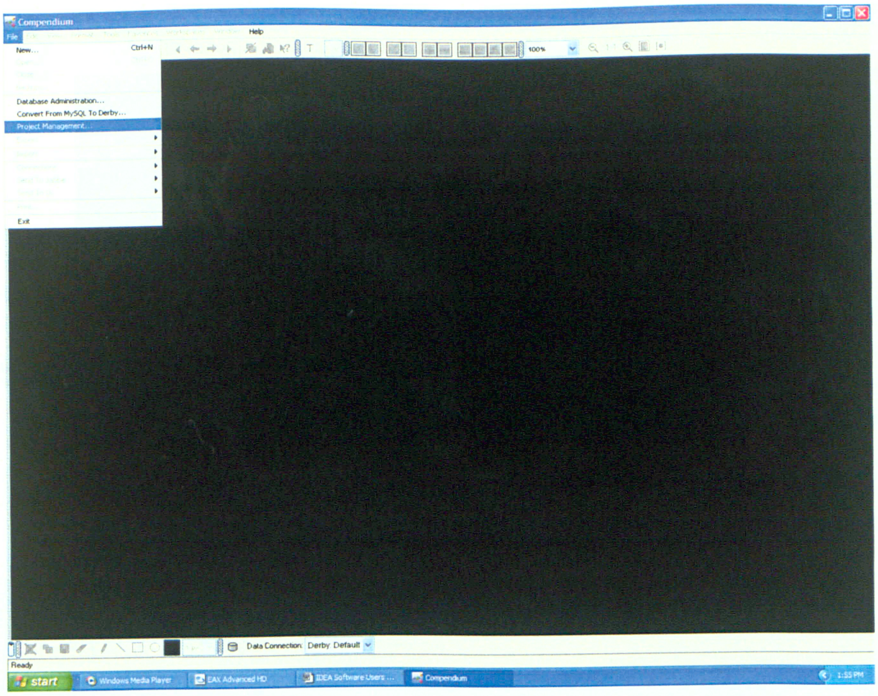


Installation of the IDEA Interface to the default drive (C:) is **highly** recommended, however if you have installed Compendium to a different drive, install the interface to same drive that you made note of in Phase 1.

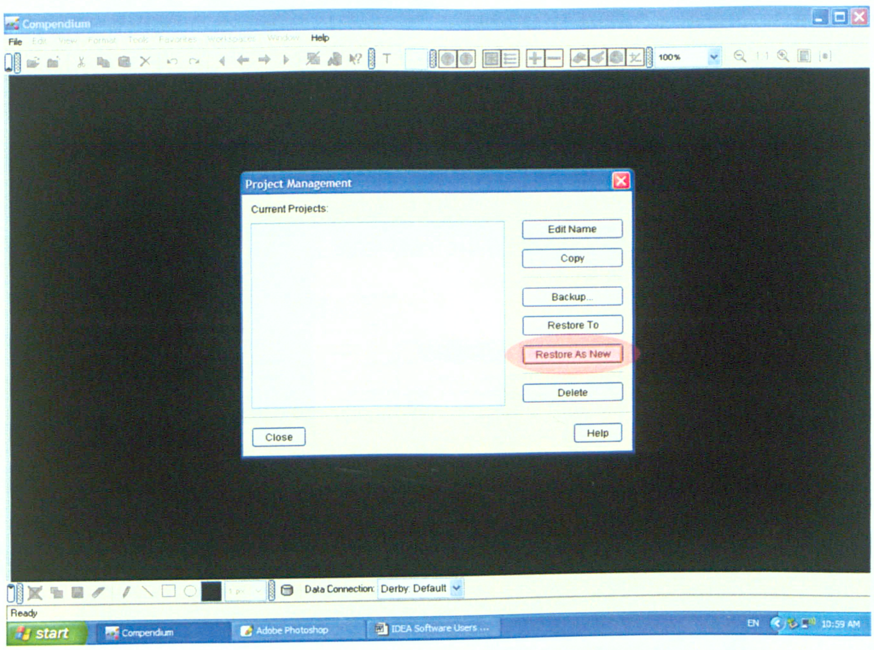
Phase 3: Import the Interface into Compendium

1.) Start Compendium by clicking on the start button and selecting the Compendium icon in the program group specified during the installation. The default path is **Start → All Programs → Compendium → Compendium**.

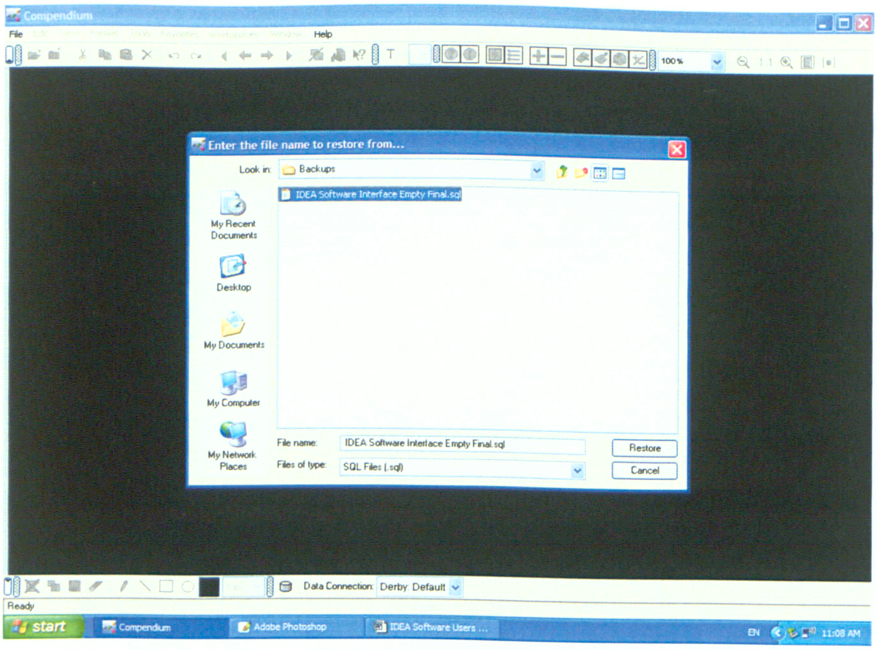
2.) Next click on **File → Project Management**.



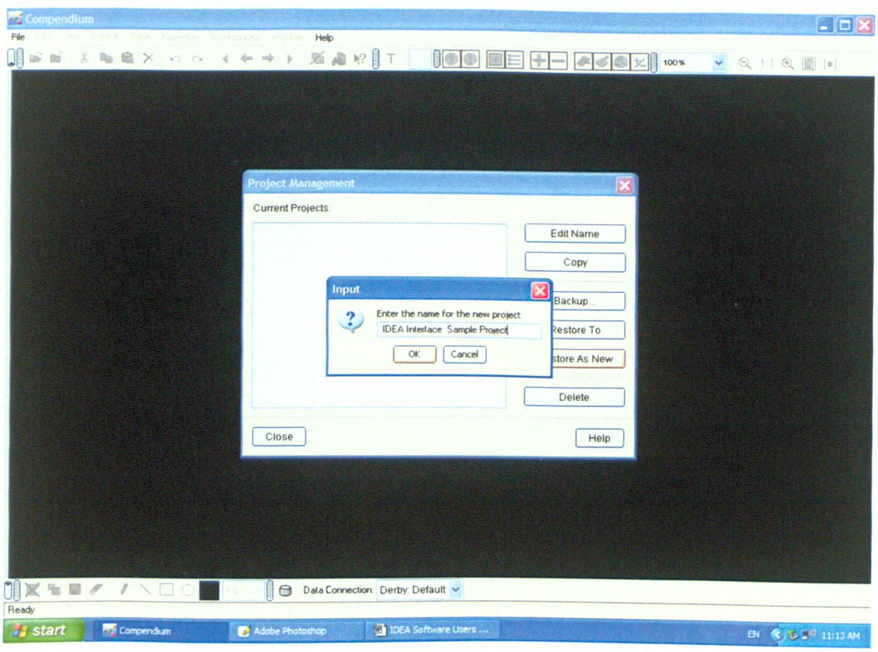
3.) In the dialog box that opens click on **Restore as New**.



4.) A file selection dialog box will now open. Select the file named **IDEA Software Interface Empty Final.sql** and then click on **Restore**.



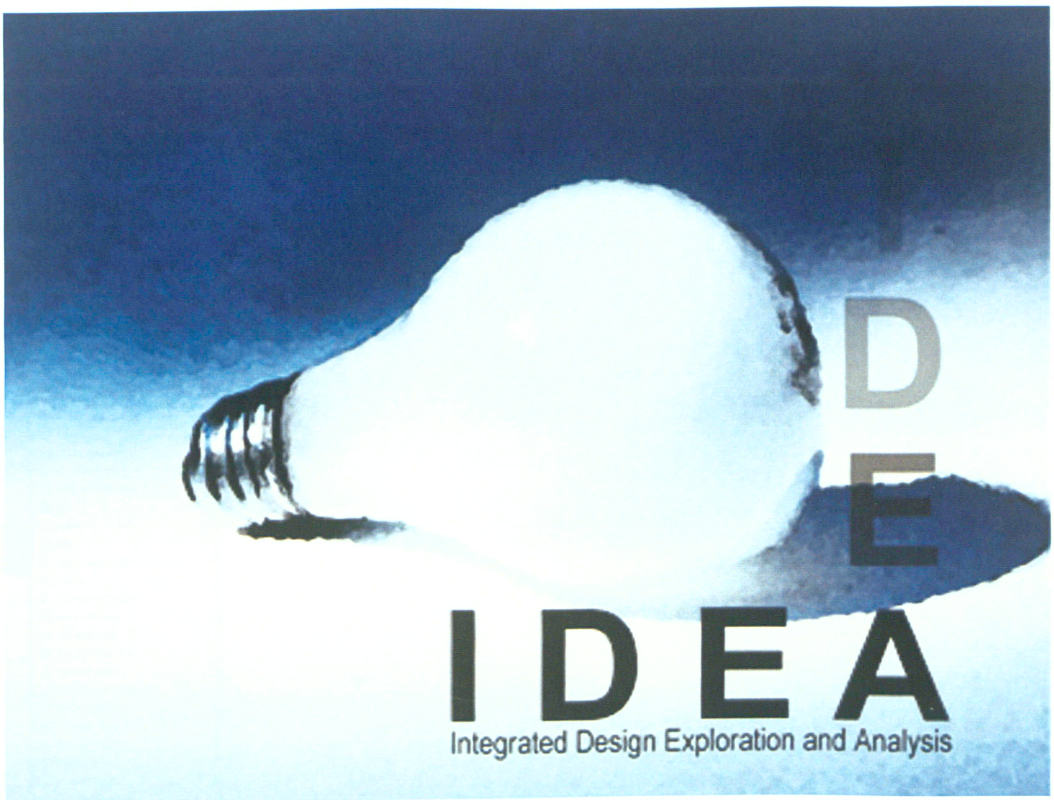
5.) Compendium will now ask you to name this instance of the interface. For the purposes of this manual we will call the interface **IDEA Interface: Sample Project**, but you can name the interface whatever you wish. Click on **Ok** to begin the import process.



6.) Once Compendium has finished importing the interface, click on **Close** to close the project management dialog box. The IDEA Interface is now ready for use.

Part 2

Use of the Interface



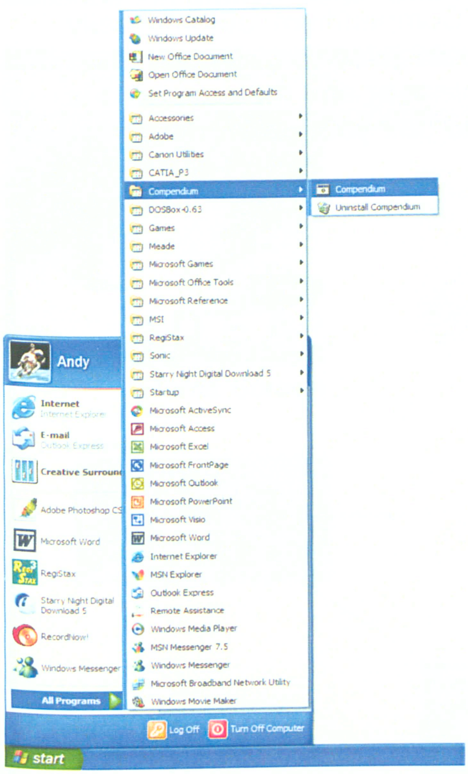
Chapter 3

IDEA: The Basics

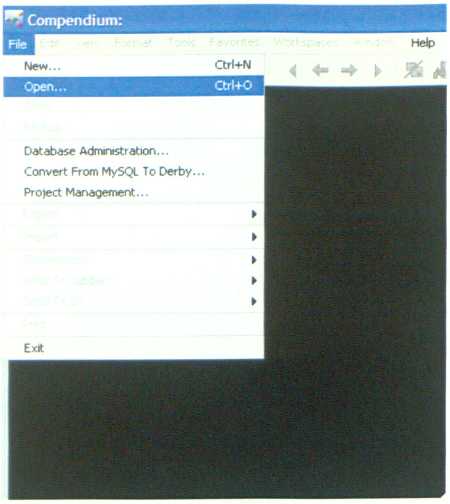
This chapter will introduce you to some of the basics for getting around the IDEA interface as well as demonstrating the technique for saving and loading projects in IDEA.

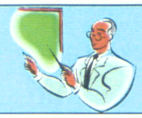
Starting IDEA

Because IDEA is an add-on to Compendium there is no separate start-up icon for IDEA, instead one must start Compendium and then load the interface within that program. Assuming you used the default installation options Compendium can be started using **Start → All Programs → Compendium → Compendium.**



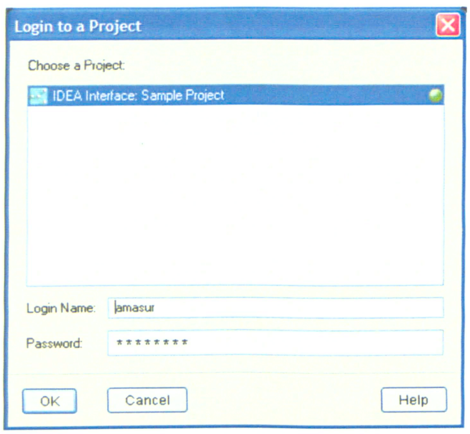
Once Compendium has started we now have to load the interface. This is accomplished by clicking on **File → Open** as shown below.



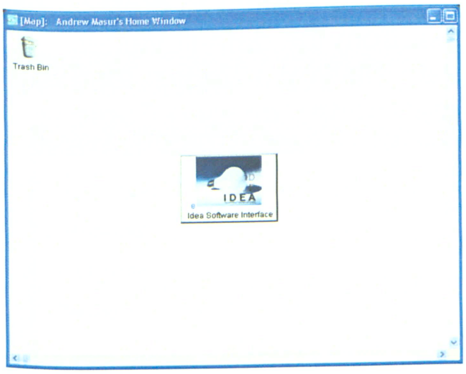


Alternatively you can also use **Ctrl+O** as a quick key combination for opening files.

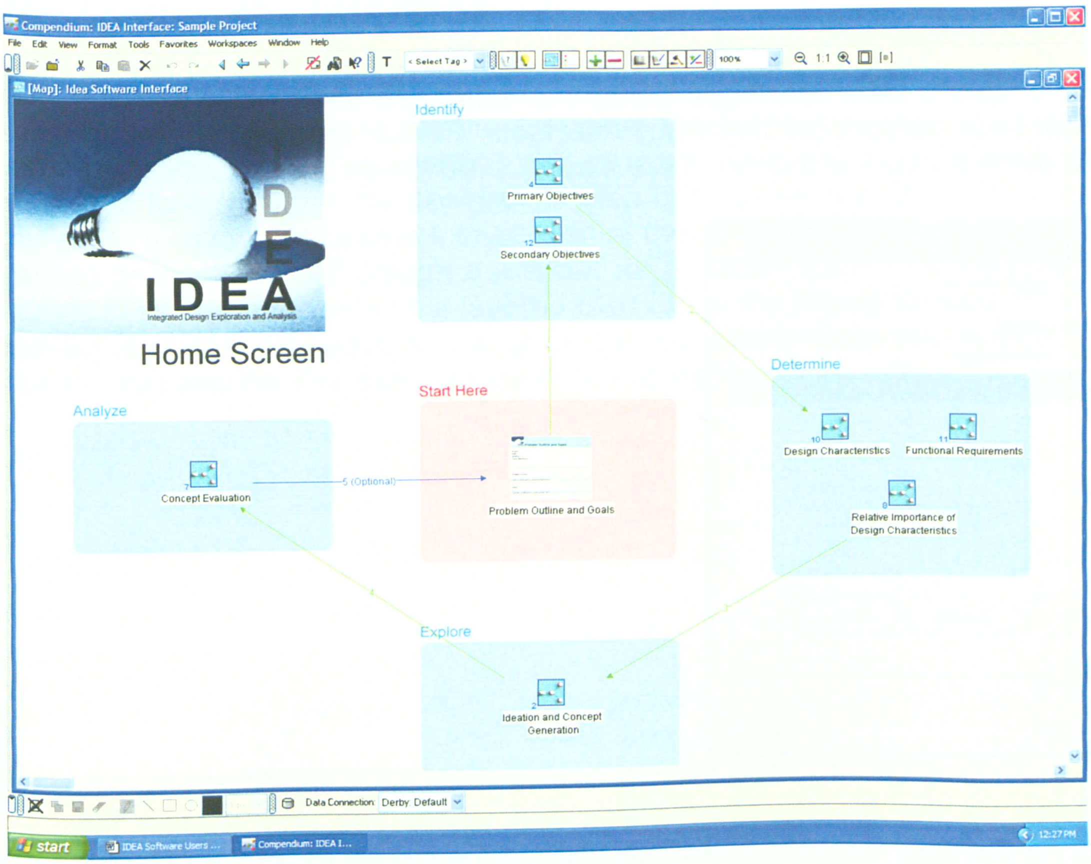
The **Login to Project** dialog box will open. Normally you would select the project you would want to open in the main window and then fill in your username and password, however since this is our first project Compendium automatically selects our sample project and fills in the username and password. If by chance these are not automatically filled in the username is **amasur** and the password is **mybooboo**, you will be able to change these later to suit your preferences, for now just click on **Ok** to open the interface.



On the next screen double-click the **IDEA Software Interface** icon to start the interface.

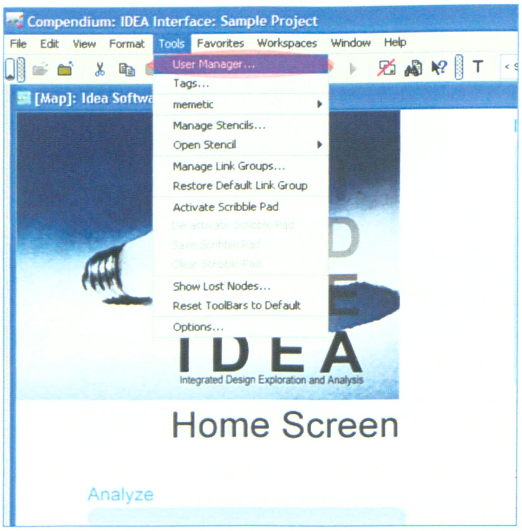


You will now be brought to the **Home Screen** for the IDEA Interface. Looking at the image below one can see that the home screen has been divided into different sub-sections corresponding to the current phase of concept design. By following the arrows on the home screen users are lead through the IDEA process while the software takes care of much of the bookkeeping, allowing designers to focus more on the design. Each of the sub-sections will be described in greater detail later in this manual.

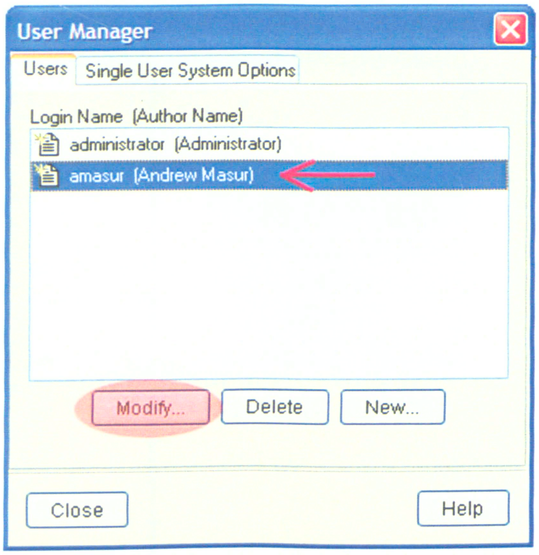


Changing Your Username and Password

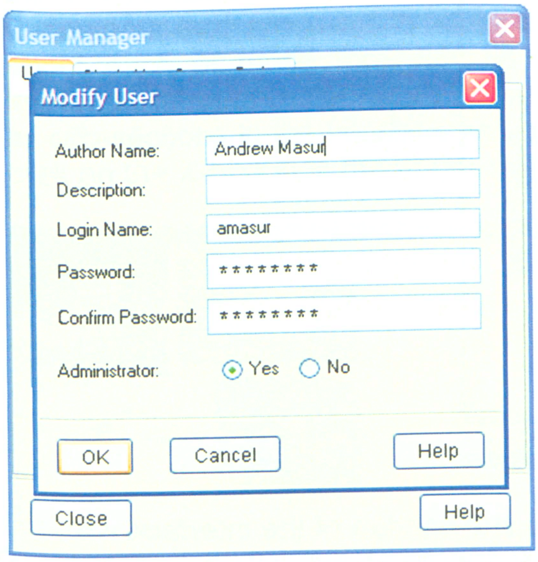
Because Compendium uses a username and password system to protect the contents of databases you may wish to change these from the default values that were set when you installed the IDEA interface. To do this you must first load the IDEA interface as described above. Once you have done this click on **Tools → User Manager…** in the menu bar as shown below.



This will bring up the **User Manager** dialog box which will show all of the user accounts currently associated with the interface. Select the account called **amasur (Andrew Masur)** and click on the **Modify…** button as shown below.

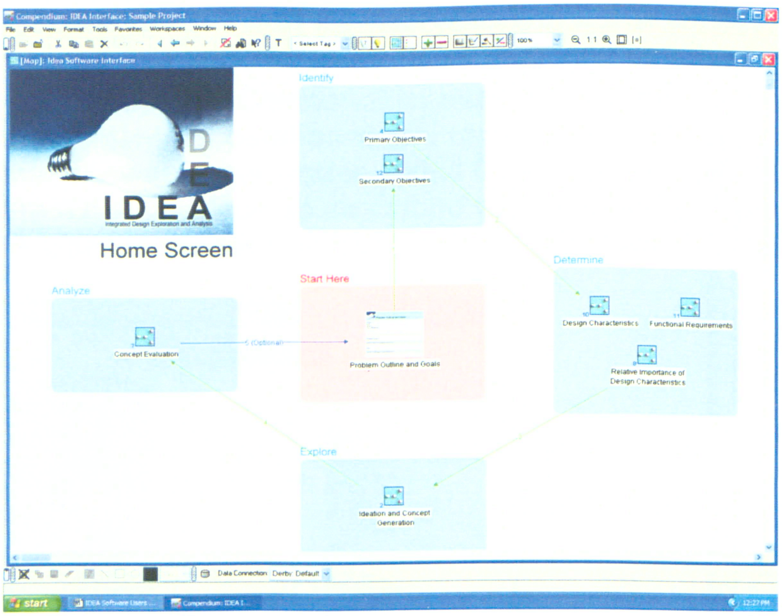


This will bring up the **Modify User** dialog box where you can change the parameters for the user account. In the box change the values for **Author Name**, **Login Name**, and **Password** to ones that you can easily remember. Make sure that the **Administrator** setting is set to **Yes**. Finally click on **Ok** and then **Close** in User Manager dialog box to save these changes. All of these settings can be seen below.

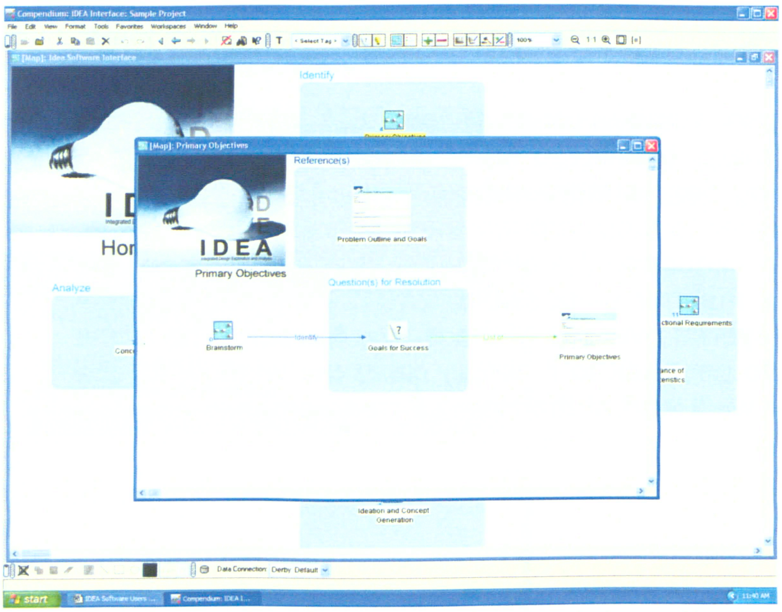


Navigating the Interface

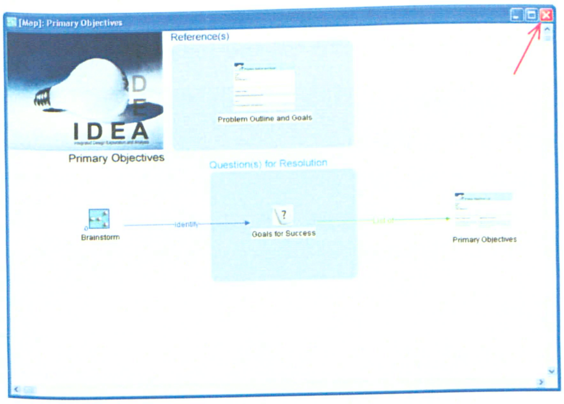
Navigating through the various sub-sections of the IDEA interface is very simple. As we can see in the image below the home screen is divided into a series of modules each containing one or more sub-sections. The modules are designed to encourage the designer to step through each of them in sequence. By going through the modules in sequence the designer is forced to look at the design problem in as thorough a manner as possible. Each of the sub-sections opens in its own window and is layered over top of the **Home Screen**. The home screen is always available, and to get back to it simply close the window for the current sub-section. For example we'll start at the Home Screen again.



Now, if we want to access the Primary Objectives sub-section of the interface we only have to double-click its icon to bring up the window as shown below.



We can clearly see that the Home Screen is still displayed behind the window for the current sub-section. Looking at the layout of the sub-section we see that it has also been divided into several distinct areas. The grey-shaded area along the top of the window displays references which may be useful for the current sub-section. These references could be files created in previous steps, or internet links which contain useful information. Looking straight down we can see a blue shaded area labelled **Questions for Resolution**. The icons in this area denote issues for the designer to think about as they work their way through the sub-section. The user is encouraged to think about as many of these issues as possible since this will help to create a better understanding of the problem. Icons to the left of this blue section denote tools that can be used to help answer these questions while icons to the right of the blue section typically denote documents that are used to store the results from these tools. Further details about each of these tools will be documented later in this manual, but the important thing to make note of now is the basic layout of the sub-section, since all sub-sections in the interface will follow it. To return to the Home Screen simply click on the X-button in the upper-right corner of the window as shown below.

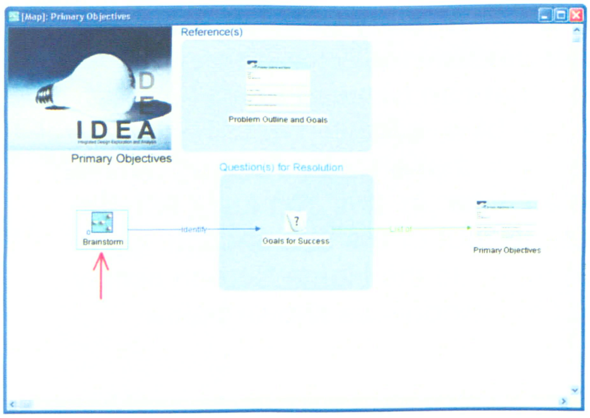


Working with Nodes

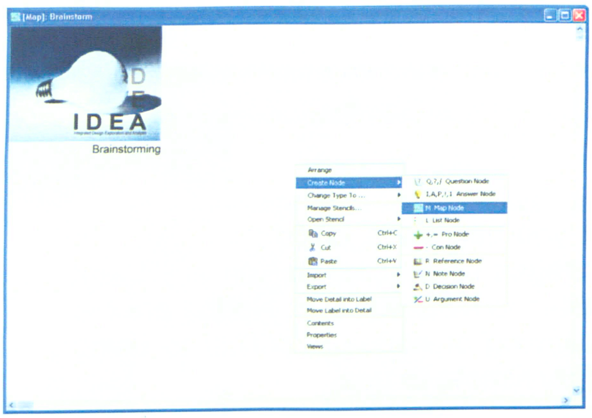
One of the most important things that you can learn to do in IDEA is to create and delete nodes. Much of the early stages of concept design are spent brainstorming and IDEA includes many templates to help with this. Nodes are the key to using these templates, so an understanding of nodes is critical to using IDEA. Don't worry however, nodes are incredibly simple to use.

Creating Nodes

Let us cover the creation of nodes by way of example. To begin start on the **Home Screen** and double-click on **Primary Objectives**, this will open the window for the primary objectives sub-section. Next, double-click on the **Brainstorm** icon as shown below; this will open a brainstorming template.

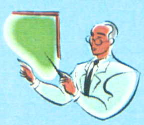


Now that we have a blank brainstorming template we can use it to practice creating nodes. There are actually several ways to create nodes, but the most foolproof method involves right-clicking on a blank area within the template to bring up a context-sensitive menu. As can be seen below this menu contains an entry for node creation and clicking on this entry will open a list of the available nodes.



Clicking on one of the available nodes in the list creates that node in the template. For this example we will create an answer node and call it Sample Node. We can change this name later by simply clicking on the text below the icon.

An alternative and faster way of creating nodes is to use keyboard shortcuts. Using this method you simply press a letter key on the keyboard that corresponds to the type of node that you want to create, the node is then created where the mouse cursor is currently pointing. The list of nodes and their shortcuts is as follows.



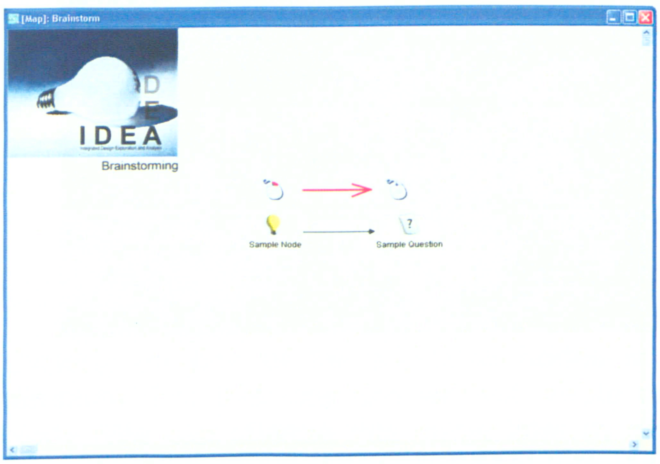
Node Type	**Keyboard Shortcut**
Question	Q
Answer	A
Map	M
List	L
Pro	+,=
Con	-
Reference	R
Note	N
Decision	D
Argument	U

Linking Nodes

Another important aspect of working with nodes is linking nodes together. By linking nodes we are able to represent a flow of information moving from one node to another, which in turn represents the flow of information and ideas.

To create a link between nodes we first create another node using the steps discussed in the previous section. For example let us create a question node called **Sample Question** as shown below.

The creation of a link between two nodes uses the same mouse mechanics as clicking and dragging in many Windows applications. To creak the link right-click on the answer node and then while still holding down the right mouse button drag the mouse over to the answer node and release the right mouse button. This will create an arrow between the two nodes, with the arrow pointing to the question node. To create an arrow pointing in the opposite direction we would have reversed the direction of the drag. You can also label the link by clicking on the middle of the arrow; this will allow you to insert a short text label describing the meaning of the link.



Deleting Nodes

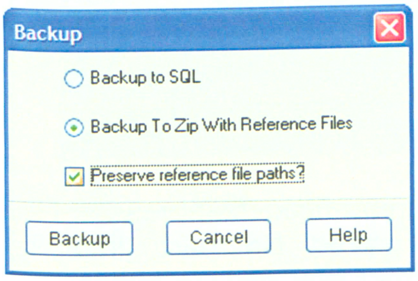
Deleting nodes simply requires that you click on the node you want to delete once and then press the delete key on your keyboard. If the node you are deleting is linked to one or more nodes those links will be deleted as well.

Saving and Loading Projects

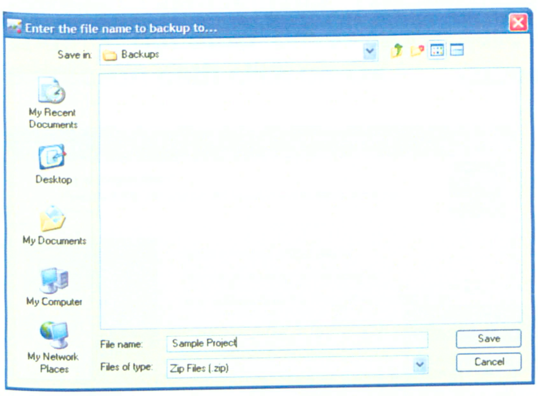
The single largest drawback to using Compendium as the backbone of the IDEA interface is the manner in which the user is forced to save and load projects. Compendium does not have the capability to save and load documents in a manner which would be familiar to most users. Instead users are forced to export and import projects in order to see previous work, a process which is far more complicated than it should be.

Saving Projects

Saving projects in IDEA is accomplished through Compendium's **Backup** command which not only saves the contents of all of the templates, but also all of the changes made to external files linked to the interface. The first step in saving a project is to have that project open using the techniques previously described in the manual. Once the project has been opened click on **File → Backup...** to open the backup dialog box as shown below.



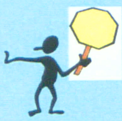
To properly save a project you should select the same options as shown in the above dialog box, namely you should select **Backup to Zip With Reference Files** and also ensure that the **Preserve Reference File Paths** box has a checkmark in it. Once these settings have been confirmed click on **Backup**.



Clicking on backup will now open the above dialog box where you can specify the location where you would like to place the save file. Simply select the desired location and give the file a descriptive name, in this example we will use the name **Sample Project**. Finally click on **Save** to backup the project to the selected file and location. If the program gives you any warnings about reference files not being found simply ignore them and click on **Ok**.

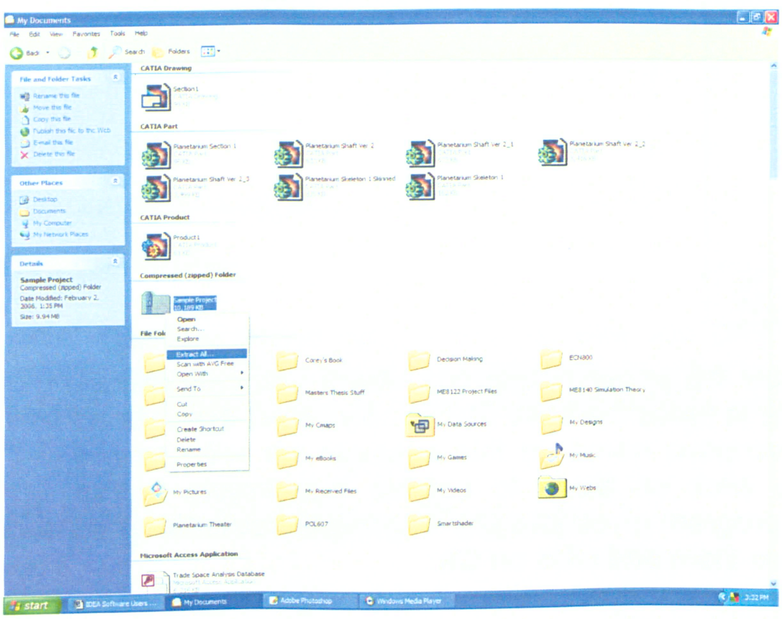
One of the most common problems forwarded to a corporate IT center usually revolves around a user not paying attention to where they saved their files. The best piece of advice is to **pay attention** to where you are saving the backup file. If you do not specify a location for the backup Compendium will save it to ***X*:\Program Files\Compendium\Backup** where *X* is the drive that you installed Compendium to. The default installation drive is C. Now, if you can't remember the drive.....

Loading Projects



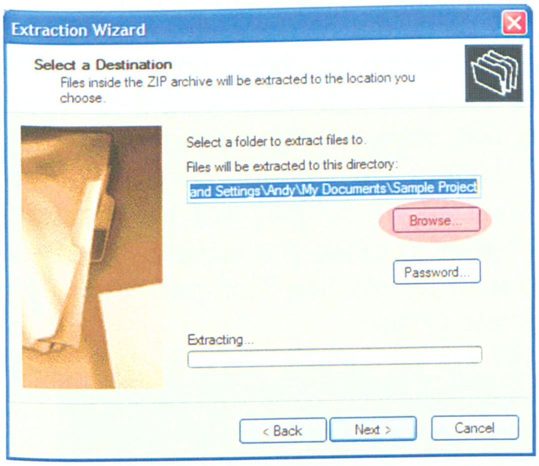
Due to the way that Compendium works with databases it is not possible to work on more than one project at a time. When you load a new project all of the linked files from the current project are **overwritten**. Thus it is imperative that you **SAVE YOUR CURRENT PROJECT** using the techniques from the previous section before loading an existing project that you have previously saved.

Compendium saves your IDEA projects in a standardized compressed format known as Zip. Before you can import the saved file back into Compendium you must uncompress the files back into the Compendium folder. The following instructions assume you are using Windows XP which has built in Zip tools, in other versions of Windows you will have to download an appropriate utility, but the general procedure will remain the same. In this example our **Sample Project.zip** file has been saved to the My Documents folder. The first step is to right-click the icon representing the file. This will bring up a menu which will have an entry called **Extract All...** as shown below, click on this.

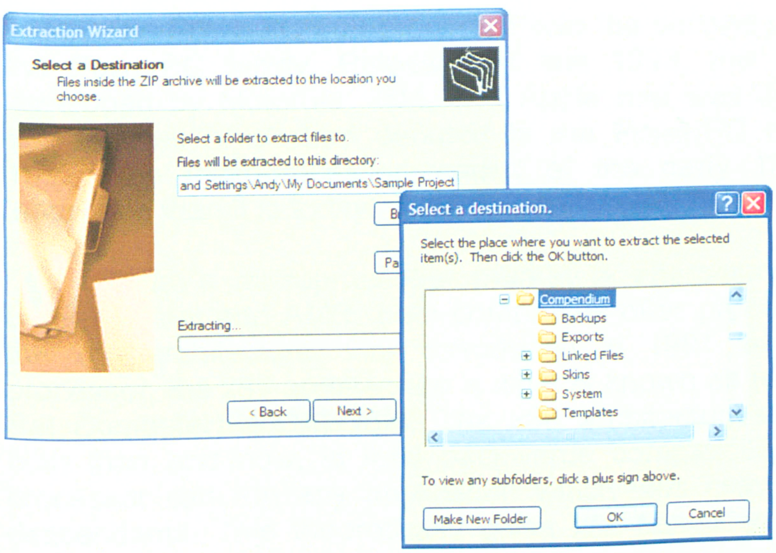


Free, downloadable Zip extractors can be found at http://www.winzip.com and http://www.winrar.com .

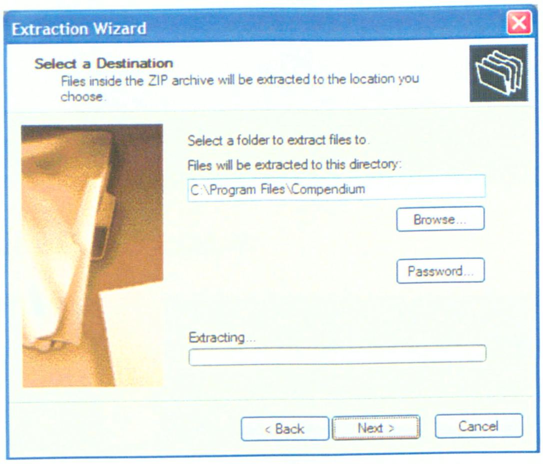
This brings up the **Compressed Folders Extraction Wizard**. The first screen merely welcomes you to the wizard, click on **Next**. The next screen details the location where you will be extracting the files. The default location is incorrect and has to be changed to the Compendium installation directory. To do this, first click on the **Browse** button as shown below.



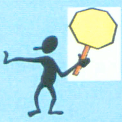
This brings up a dialog box which allows you to specify the extraction location. We need to extract the files to the Compendium folder. By default this is located in ***X*:\Program Files\Compendium** where *X* is the letter of the drive where you installed Compendium. The example below shows Compendium installed to the C drive, but other drives would follow a similar path. Once you have verified the extraction path click on **Ok.**



You can see that the path has now been changed to your selection. Click on **Next** to continue.



Once the files have finished extracting click on **Finish** to close the wizard. You may now open your previous project in Compendium by starting Compendium, clicking on **File → Open** and selecting the appropriate project.



Due to the way that Compendium works with databases it is not possible to work on more than one project at a time. When you load a new project all of the linked files from the current project are **overwritten**. Thus it is imperative that you **SAVE YOUR CURRENT PROJECT** using the techniques from the previous section before loading a previous project.

Chapter 4

Sample Project: Introduction

One of the best ways to learn a new technique is through an example. It provides a baseline to which to compare your current efforts, and illuminates difficulties and shortcuts better than simply describing a new technique. With this in mind, the next several chapters of this manual will proceed through a sample project to better demonstrate the use of the interface and the tools which it contains. This chapter will mainly be used to describe the background of the problem for the project as well as demonstrating the first stage of the IDEA process.

Background: Design of the PowerPC 603 Processor

The sample project which we will cover in the next several chapters is a case study on the design of the PowerPC 603 microprocessor. The development of this processor is well documented and as such serves as a good example upon which to conduct a case study since we will be able to compare our results and decisions to those of the engineers who worked on the actual project. The following introduction will outline the background of the PowerPC 603 design project. This kind of background information is important when performing design tradeoffs, so reading this entire section is highly recommended.

The PowerPC 603 processor was the second processor to be released in the PowerPC family. Released in late 1993, the PowerPC 603 was jointly developed by Motorola, IBM, and Apple and was designed to build upon the technologies which had debuted in the PowerPC 601 processor released in 1992. The microprocessor market of the early 1990's was a far different landscape than it is today.

Today's microprocessor market is characterised by fierce competition between Intel and AMD; both companies offer processors which are based on the x86 architecture. First introduced in 1981 with the original Intel 8086 processor, the x86 architecture is what is known as a CISC (Complex Instruction Set Computer) architecture. The x86 architecture's single largest advantage, both then and now, is total backwards compatibility. Any modern x86-based processor can run any application which was created for 8086 or any of its descendants. This theoretically gives x86-based computers the advantage of access to almost 30 years of software. However, the use of a three-decade old architecture also has many drawbacks.

The foremost is that x86 is a very inefficient architecture which has many bottlenecks which can be traced back to its roots in the late 1970s. It is possible to get around these bottlenecks. Modern processors have solved them quite effectively, but doing so requires vastly more complex processor designs to overcome these inefficiencies.

This leads us to the processor market as seen in 1993. Intel had just released its original Pentium microprocessor, the fifth generation of the x86 architecture. This processor offered vastly improved performance over its predecessors, however even at this juncture Intel was already encountering design difficulties due to the barely decade old x86 architecture. The original Pentium was notoriously power-hungry and users of many of the first Pentium-based systems complained of overheating problems.

On the other side of the market sat Apple Computer who had seen its market-share steadily eroded since the release of the IBM PC in 1981. In an attempt to reverse this trend the executives at Apple decided that they would create a new home computer which would not only have the legendary ease of use of the Macintosh, but would also be the highest-performing computer ever offered for home use.

In order to achieve these goals Apple contracted Motorola and IBM to design a totally new processor for the new range of computers. By forgoing support for legacy applications the designers believed that they could avoid many of the problems that continued to plague existing processors while at the same time achieving industry-leading levels of performance.

The result of this collaboration was the PowerPC 601 processor, the first processor to implement what was known as the PowerPC Architecture. The PowerPC architecture was designed based upon the principals of RISC (Reduced Instruction Set Computer) a completely different way of processing instructions which would theoretically lead to much simpler processor designs and higher performance.

While the PowerPC 601 met with great critical acclaim, it did not actually fully support all of the advantages of the PowerPC architecture. One of the most pressing problems was that from the outset the PowerPC was designed to be the highest performing processor available. Yet the 601, at best, only equalled the performance of the Intel Pentium. Because of this many potential costumers were not swayed by the new processor since it would force them to buy entirely new software to support it and offered no tangible performance advantages in return. Additionally, while the PowerPC 601 did indeed use somewhat less power than the Pentium, the power draw still precluded the use of the new processor in notebook computers.

It was decided that the second generation of PowerPC processor would not only have to perform better than any other processor on the market, but would also have to minimize power draw so that it could be used in notebook computers. Thus the main targets for the design of the PowerPC 603 were set; it would have to achieve the performance of workstations from the time period while operating at notebook power levels.

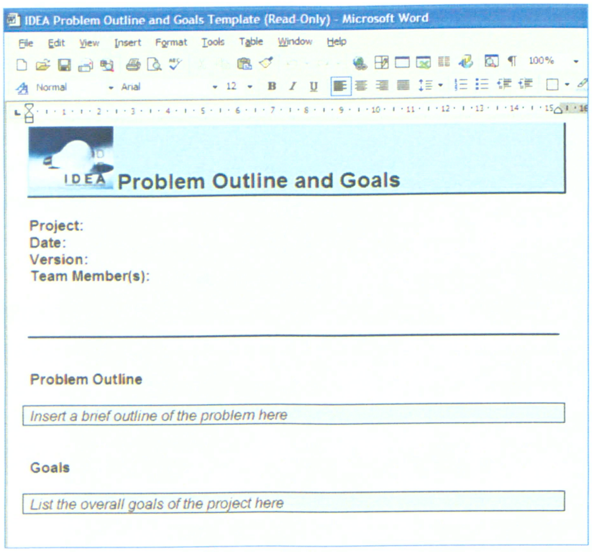
The instructions for a CISC (Complex Instruction Set Computer) processor are designed around the premise of packing as much information into as small a space as possible. This type of thinking is a hold-over from the early days of computing when memory was extremely scarce and expensive. The engineers at the time devised an instruction set which would allow them to fit as many different functions as possible into an instruction word. While this is very efficient from a storage standpoint, CISC instructions are notoriously inefficient to execute. No two instructions are the same length, making it very difficult to optimize a processor to run instructions in parallel since one instruction may take longer to execute than another and if the instructions are interdependent then one instruction pipeline will stall while it waits for the other instruction to be completed.

RISC (Reduced Instruction Set Computer) instructions, on the other hand, are designed to break down a problem into very small segments. Each individual instruction in a RISC architecture contains very little information, but they are designed in such a way that all instructions are the same length. Thus it takes many more instructions for a RISC processor to complete the same task as a CISC processor, but due to the small standardized size of each instruction a RISC processor is able to process many instructions in parallel very quickly, improving performance. In fact, all modern-day processors currently are RISC processors. They simply contain a layer of hardware which takes the old-style CISC x86 instructions and converts them to RISC instructions for use in the processor. This hardware interpreter is actually the single most complex device in a modern processor.

IDEA Step 1: Problem Outline and Goals

The first step in the IDEA process is to generate a crisp definition of the problem at hand as well as outlining the goals for successful completion of the project. This is an important step in the concept design process because it ensures that all team members and stakeholders are in agreement on the overall direction of the project. Writing out the problem outline and goals also has the benefit of forcing the designers to look at those goals one last time and catch any unrealistic expectations or requirements before the later stages of design.

To record the problem outline and goals first start Compendium and open our sample project using the steps previously described in the manual. Double-click **Problem Outline and Goals** in the red **Start Here** box to open the template.



Our first task is to determine the problem outline for the sample project. The problem outline should be a single, unambiguous statement which describes exactly what we hope to accomplish by doing the project. In the case of our sample project a possible problem outline might go something like this "The design of a PC microprocessor that delivers workstation performance levels at notebook power levels".

Notice how the problem outline does not contain any hard numbers or suggestions as to which configuration possible concepts should take; it merely states what the final product should do in order to be considered successful. The next task is to list the goals of the project which will help to steer the direction of not only concept development, but also provide desirable qualities for the final product which helps with decision making during concept selection. Some goals for our sample project are as follows:

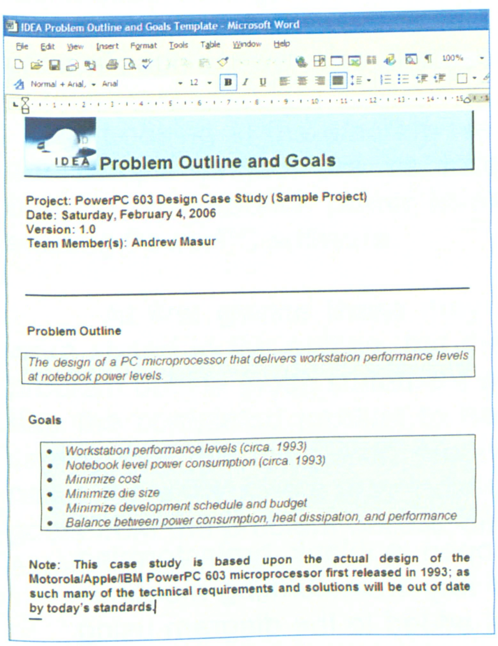
- Workstation performance levels (circa. 1993)
- Notebook power consumption levels (circa. 1993)
- Minimize cost
- Minimize die size
- Minimize development schedule and budget

- Balance between power consumption, heat dissipation and performance

The inclusion of the first two goals is self explanatory; they are taken directly from the problem outline. The third goal, minimize cost, is important because in 1993 the first iteration of the PowerPC platform had met with critical success, but with limited commercial success. People seemed unwilling to switch to the new platform due to the increased software demands that it would entail. Minimizing the cost of the new processor would lead to a corresponding reduction in the price of the systems based upon it. The availability of cheaper systems could lead to greater interest in PowerPC-based systems, thus increasing the success of the processor design.

The next two goals are results of attempting to minimize the overall cost of the processor. Processors are manufactured on large wafers which are then split into the individual processors. By minimizing the size of each processor die it is possible to fit more processors per wafer which makes manufacturing more efficient and thus cheaper. Minimizing the development schedule and budget leads to a reduction in the overall cost of the project which translates to a lower per-unit price on the processor for system builders and consumers. The final goal suggests that finding the best balance between power consumption, heat dissipation and performance will probably be the key to a successful design.

We enter this data into the template as shown below and **save** it. Because the template is in **Microsoft Word** format it has most likely been opened using **Microsoft Word**. The changes to the template can thus be saved by using **CTRL+S**. If you use another word processor that can open Word files please use the save procedure for that program instead. Once you have saved the changes, exit the program to return to the IDEA interface.



Chapter 5

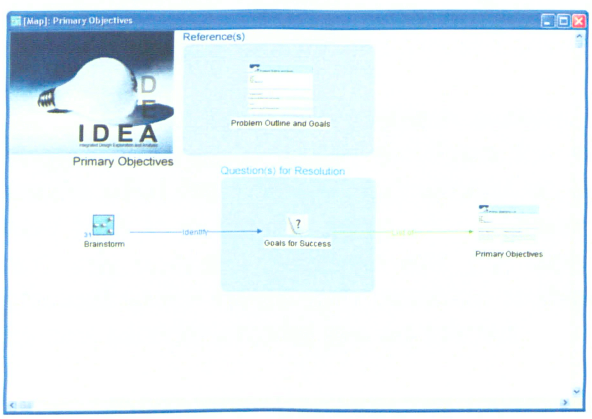
Identify

Now that we have a broad understanding of the problem at hand, the next step is to determine the primary and secondary objectives for the product. In essence we are taking the very high-level goals that we determined in the previous step and are taking them to an extra level of detail to determine more precisely the characteristics of a successful product.

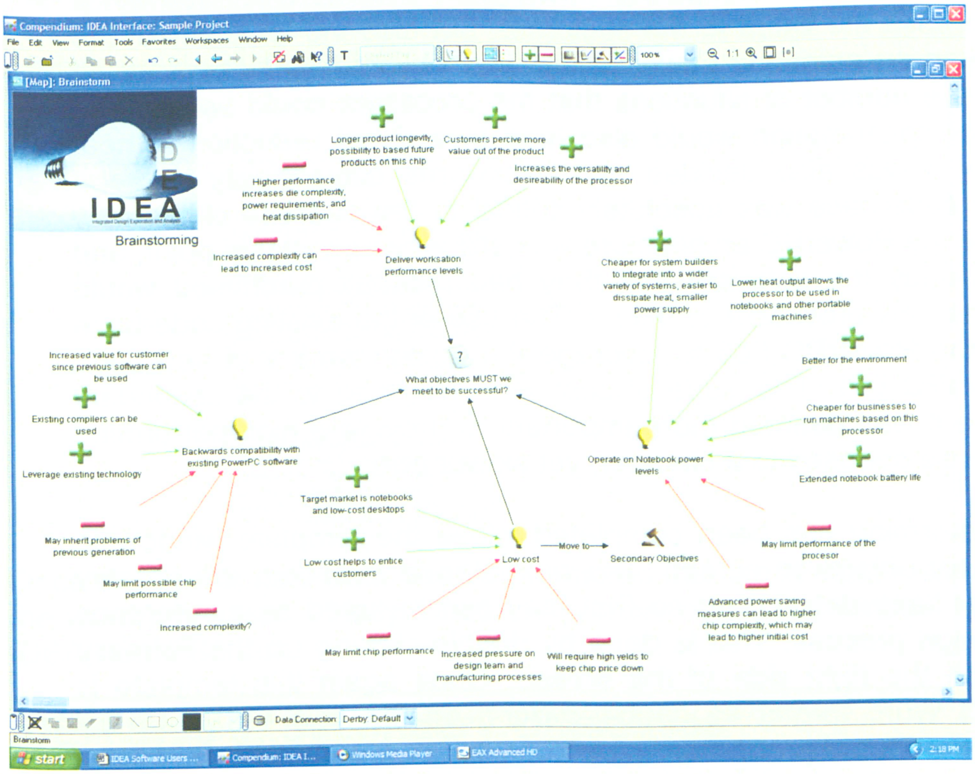
IDEA Step 2: Primary Objectives

In any design project, the primary objectives are of maximum importance. These are things that the final design MUST achieve in order to be considered a success. Most of the time these objectives are derived from the goals generated during the Problem Outline and Goals step of the IDEA process. The designer must carefully consider exactly what will be included in the primary objectives since they will have a very large impact on the final design of the product. Needlessly including something in the primary objectives can greatly increase the difficulty of the problem.

Brainstorming is a good way of determining the primary objectives for the design problem and the IDEA interface contains a brainstorming template to assist with this. In order to access this template, double-click on the **Primary Objectives** icon in the light-blue **Identify** section of the Home Screen. This will open the following window.



To open the brainstorming template simply double-click the **Brainstorm** icon near the left-side of the window. This will open a blank brainstorming template which can be used to record the results of any brainstorming meetings between team members. All nodes and links can be added to the diagram using the techniques previously discussed in this manual. The shortcuts for node creation will be repeated at the end of this chapter. An example of a completed brainstorming template for our sample problem is shown below.



As we can see, the layout of this diagram closely follows the standard layout that most students are taught when they are first introduced to brainstorming techniques. The question that requires resolution is in the center of the diagram and the potential answers to that question arrayed around it with the pros and cons of each answer also displayed.

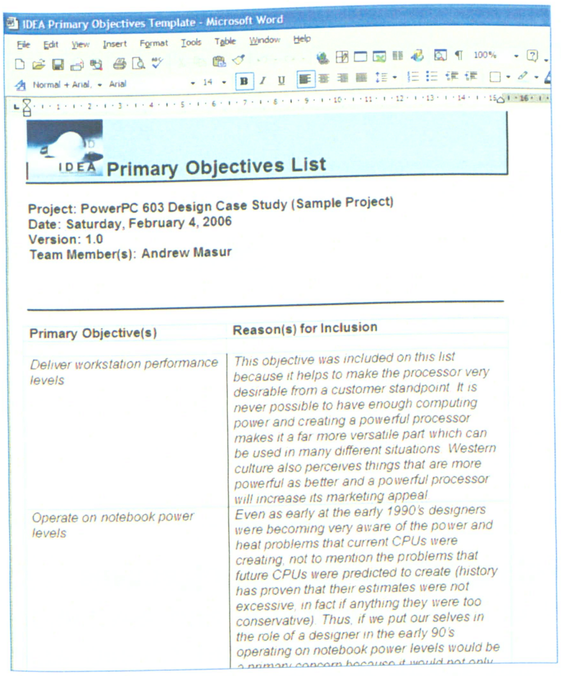
Looking at the diagram we find that some possible primary objectives for our sample project are as follows: deliver workstation performance levels, operate on notebook power levels, low cost, and backwards compatibility with existing PowerPC software.

At first glance these may seem to simply repeat the project goals that were stated in the outline, but there is a subtle yet important difference. In the problem outline these statement were simply goals, characteristics that we would **like** the completed product to have. By placing these statements here we are saying that the completed product **must** have these characteristics. In some cases the entries here may indeed be nothing more than a repeat of some of the goals from the problem outline, but most of the time the primary objectives will be expansions on these goals.

We can also see that of the initial four entries, only three have survived the brainstorming process to remain as primary objectives. The delivery of workstation performance levels was included because of the simple fact that it is impossible to have too much computing power, and increasing performance would not only increase the value to the customer, but also the longevity of the processor. Operation on notebook power levels was included because it would not only increase the number of situations that the processor could be used in, but lowering power consumption would also lead to a direct reduction in heat generation. Finally it was also decided that backwards compatibility with the existing PowerPC code-base was absolutely critical to the success of the new processor because the PowerPC architecture was still very new at this point, and forcing users to re-buy the software that they had recently purchased would surely condemn the new processor to failure. The use of the existing code-base would also allow the PowerPC 603 to leverage the design work that had been done on the PowerPC 601. The issue of cost was relegated to the secondary objectives because it was deemed more important to provide a high-performing compelling product as opposed to a cheaper, yet slower one.

Once brainstorming has been completed, close the template window to return to the primary objectives screen. The next step is to record the primary objectives that you have determined so that they can be used as a reference later on in the design process. This is done by double-clicking on the **Primary Objectives** icon on the right side of the screen. Once again this template is saved in Microsoft Word format so double-clicking on the icon will open the template in your default word processor.

The main body of the template is divided into two columns. The leftmost column is for recording the name of the primary objective, while the column on the right is where the reasons for including that objective are recorded. While this may seem redundant, much of that work done earlier was done graphically. Forcing the user to write down the reasons for including the various primary objectives not only forces designers to consider each objective one last time, but the act of converting information stored graphically to a textual format can sometimes spark further ideas and discussion. A completed version of the template for the sample project is shown below.



IDEA Primary Objectives Template - Microsoft Word

IDEA Primary Objectives List

Project: PowerPC 603 Design Case Study (Sample Project)
Date: Saturday, February 4, 2006
Version: 1.0
Team Member(s): Andrew Masur

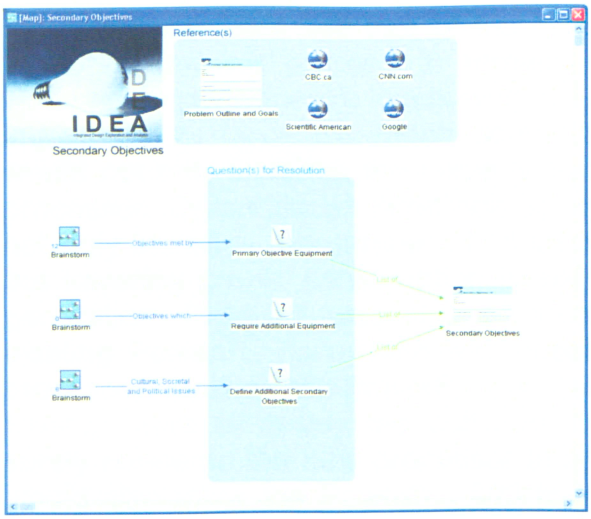
Primary Objective(s)	Reason(s) for Inclusion
Deliver workstation performance levels	*This objective was included on this list because it helps to make the processor very desirable from a customer standpoint. It is never possible to have enough computing power and creating a powerful processor makes it a far more versatile part which can be used in many different situations. Western culture also perceives things that are more powerful as better and a powerful processor will increase its marketing appeal.*
Operate on notebook power levels	*Even as early at the early 1990's designers were becoming very aware of the power and heat problems that current CPUs were creating, not to mention the problems that future CPUs were predicted to create (history has proven that their estimates were not excessive, in fact if anything they were too conservative). Thus, if we put our selves in the role of a designer in the early 90's operating on notebook power levels would be*

Once the appropriate data has been entered into the template **save** it and close the word processor. Finally, close the primary objectives window to return to the IDEA Home Screen.

IDEA Step 3: Secondary Objectives

Unlike primary objectives which **must** be met in order for a design to be considered successful, secondary objectives are characteristics which customers would **like** to see in a design, but are not required. If a secondary objective must be traded-off in order to achieve one or more of the primary objectives then it is expected that the designer will do so. These objectives are meant to add value to the design, however for the most part they are strictly optional.

To access the secondary objectives sub-section of the interface, double-click on the **Secondary Objectives** icon in the light-blue **Identify** section of the home screen; the icon can be found directly under the icon for the primary objectives. This will open the following window.



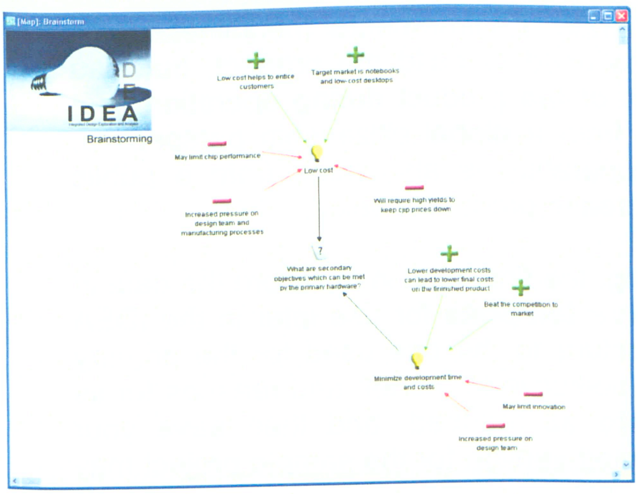
We can see here that unlike the primary objectives, secondary objectives are segregated into three groups.

The first group of secondary objectives are those that can be met by equipment that is used for the primary objectives. For example if we have a spacecraft that uses a camera system for visual control of a robotic arm as its primary function, the same system could theoretically be used to conduct further scientific observations as a secondary objective. Therefore, this group of secondary objectives is generally simpler to implement and is thus more desirable to include.

The second group of secondary objectives are those which require additional equipment in order to function. While these do not pertain to our sample project, this group of secondary objectives is generally more difficult to implement than the first. If the project is running into time or fiscal constraints, these are generally some of the first objectives to get cut.

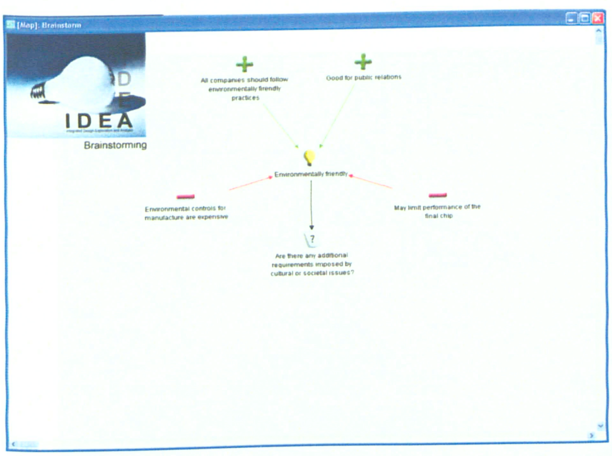
The third and final group of secondary objectives are those which are imposed by cultural, societal and political issues. While at first these may seem to be trivial, they actually can have a great impact on the final design of a product. For example, one would not design a home heating system for use in Central America, or an industrial process for use in a North American factory that dumps an inordinate amount of toxic waste.

Again, like the primary objectives, brainstorming is a good tool to use for determining secondary objectives. Double-clicking on the topmost icon in the left-side column opens the brainstorming template for the **Secondary Objectives Met by Primary Objective Equipment**. Use this template to record discussions during any team meetings by using the node and link creation techniques described previously. An example of a completed brainstorming diagram for the sample project is as follows.



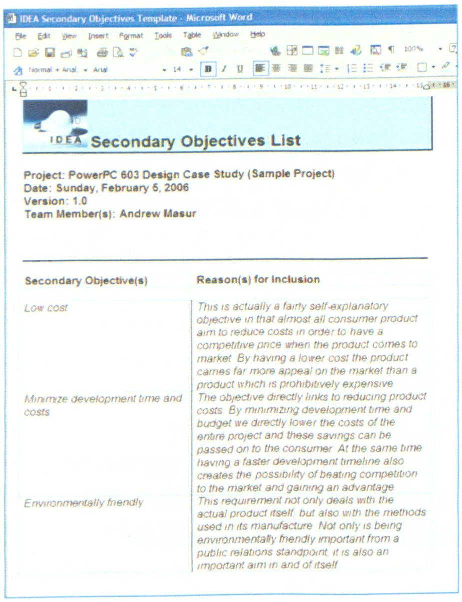
We can see that the low cost objective that was originally in the primary objectives has now been moved to this location in the secondary objectives. It was placed here because while it was felt that achieving a high-performance processor was of primary importance, creating a low cost processor would be of great benefit, thus its inclusion with the "top" secondary objectives. The second objective seen here, minimize development time and costs, is directly linked with attempting to keep the costs of the entire project to a minimum which in turn would lead to a lower cost for the consumer.

While there were no secondary objectives requiring additional equipment for our sample project, there was one objective which was imposed by societal concerns. To access the brainstorming template for **Cultural, Societal and Political Issues** double-click on the bottom icon in the left hand column in the **Secondary Objectives** sub-section window. A completed version for our sample project is shown below. Again, this template can be completed during meetings using the node and link creation techniques previously covered in the manual.



The third possible secondary objective for the sample project is making sure that the project is environmentally friendly. Environmental friendliness for a project not only deals with the processes used to build the product, but also extends to the environmental friendliness of the product during operation. Not only are environmental issues important from a global citizenship perspective, but having an environmentally friendly operation is also good from a public relations standpoint, which may help to increase processor sales.

After the completion of brainstorming, the final step is to record the data in the templates in the **Secondary Objectives template** for use later in the IDEA process. The template is identical in structure to the one used for the primary objectives, with a short description of the objective to the left and reasons for its inclusion to the right. Fill in the template as required, and then **save** your changes and close the template. A completed version for the sample project may look something like this.

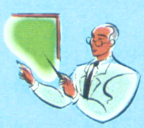

IDEA Secondary Objectives Template - Microsoft Word

Secondary Objectives List

Project: PowerPC 603 Design Case Study (Sample Project)
Date: Sunday, February 5, 2006
Version: 1.0
Team Member(s): Andrew Masur

Secondary Objective(s)	Reason(s) for Inclusion
Low cost	*This is actually a fairly self-explanatory objective in that almost all consumer product aim to reduce costs in order to have a competitive price when the product comes to market By having a lower cost the product carries far more appeal on the market than a product which is prohibitively expensive*
Minimize development time and costs	*The objective directly links to reducing product costs By minimizing development time and budget we directly lower the costs of the entire project and these savings can be passed on to the consumer At the same time having a faster development timeline also creates the possibility of beating competition to the market and gaining an advantage*
Environmentally friendly	*This requirement not only deals with the actual product itself but also with the methods used in its manufacture Not only is being environmentally friendly important from a public relations standpoint, it is also an important aim in and of itself*

These are the node creation keyboard shortcuts reproduced from earlier in the manual.



Node Type	Keyboard Shortcut
Question	Q
Answer	A
Map	M
List	L
Pro	+,=
Con	-
Reference	R
Note	N
Decision	D

Argument	U

Chapter 6

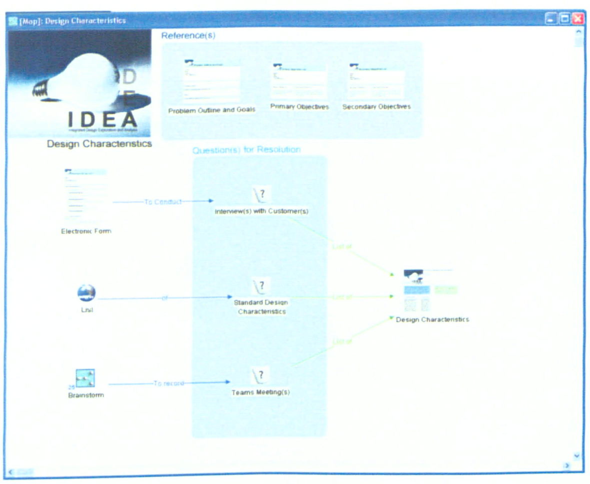
Determine

Now that we have identified the primary and secondary objectives for the project, the next step is to determine the design characteristics, functional requirements, and the relative importance of the design characteristics for our problem. IDEA has several tools that make this process fast and easy.

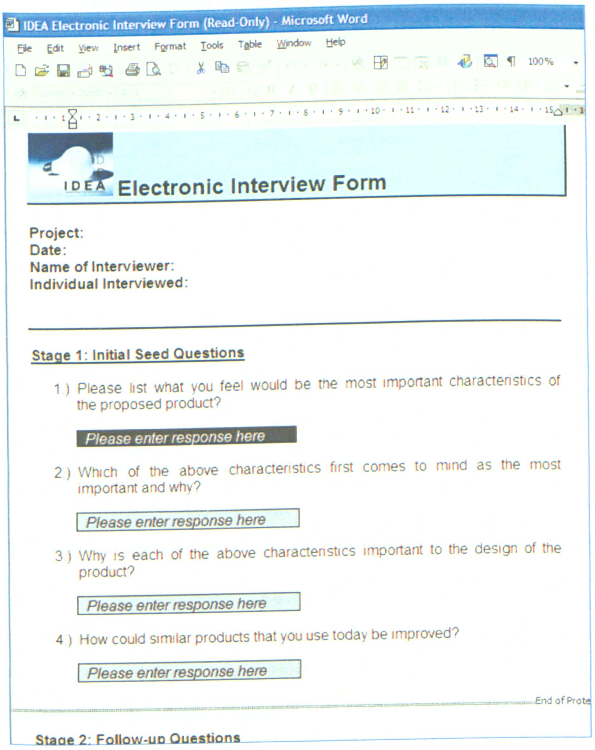
IDEA Step 4: Design Characteristics

The determination of design characteristics is an exploration of what designers think the final product should **be**. Design characteristics define the criteria that are used to measure the effectiveness of concepts during concept development. They do not imply how the various concepts will meet the requirements, but they do define certain physical criteria that can be measured to determine concept performance and requirements compliance. For example, if one of the requirements for a project states that the final product must be light weight an obvious design characteristic would be weight or mass. This characteristic would provide a common metric that one could use to decide amongst concepts; in this case a lighter weight concept would be preferable to a heavier one.

In the IDEA interface, the design characteristics tools can be accessed by double-clicking on the **Design Characteristics** icon in the light blue **Determine** area on the right-hand side of the home screen. This brings up the following window.



We can see that there are a number of tools available for the determination of the design characteristics. The first of these is an electronic form that can be used for interviews with customers or stakeholders. Double-clicking on the **Electronic Form** icon opens the following form in your word processor.


IDEA Electronic Interview Form (Read-Only) - Microsoft Word

File Edit View Insert Format Tools Table Window Help

IDEA Electronic Interview Form

Project:
Date:
Name of Interviewer:
Individual Interviewed:

Stage 1: Initial Seed Questions

1.) Please list what you feel would be the most important characteristics of the proposed product?

Please enter response here

2.) Which of the above characteristics first comes to mind as the most important and why?

Please enter response here

3.) Why is each of the above characteristics important to the design of the product?

Please enter response here

4.) How could similar products that you use today be improved?

Please enter response here

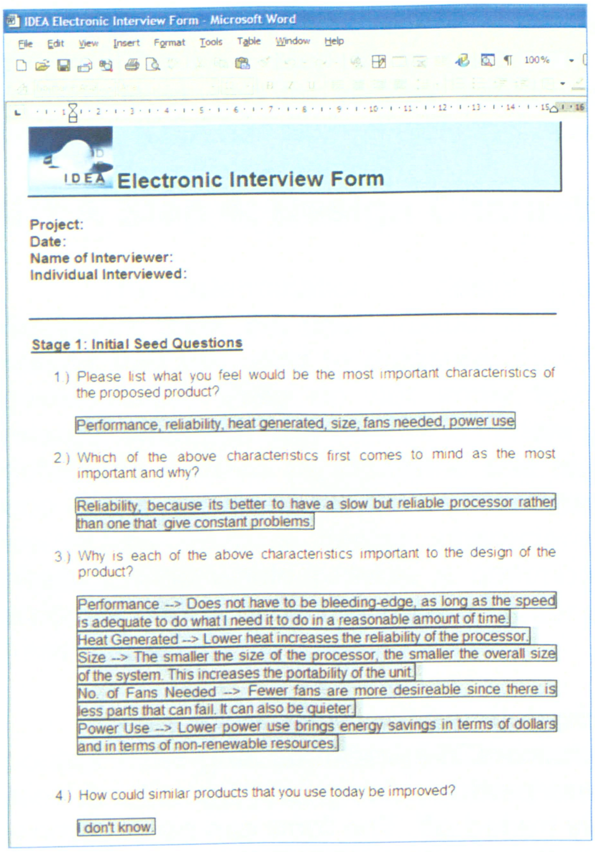
End of Prote

Stage 2: Follow-up Questions

Interviews are one of the best ways to determine exactly which design characteristics are important to potential customers. By understanding what is important to those who will be using the final product, designers are able to understand which characteristics can be safely traded off. The form can be used in a number of different ways depending on the number of individuals to be interviewed.

If the product being designed is for consumer use it is preferable to obtain as large a sample of individuals as possible since this will help to ensure that all possible preferences are captured. In this case it is advantageous to electronically distribute multiple copies of the form to various customers and have them fill it out at their leisure. The completed forms can then be returned and the results analyzed in order to determine trends.

If the product being designed is not for public use, or is expected to have a very limited market it is often preferable to conduct one-on-one interviews with a much smaller sample. One-on-one interviews are the preferred method of obtaining information because an interview is an interactive process. Depending on the answers of the individual being questioned the interviewer can spark further thought with relevant follow-up questions and observations. This leads to a more complete examination of the problem at hand and generally results in higher quality data. The electronic form is designed to support a one-on-one interview environment and includes areas for follow-up questions and other miscellaneous observations. An example of a completed interview form for our sample problem is shown below. These answers were obtained from a one-on-one interview, but they could have been obtained electronically just as easily.



IDEA Electronic Interview Form - Microsoft Word

IDEA Electronic Interview Form

Project:
Date:
Name of Interviewer:
Individual Interviewed:

Stage 1: Initial Seed Questions

1) Please list what you feel would be the most important characteristics of the proposed product?

Performance, reliability, heat generated, size, fans needed, power use

2) Which of the above characteristics first comes to mind as the most important and why?

Reliability, because its better to have a slow but reliable processor rather than one that give constant problems.

3) Why is each of the above characteristics important to the design of the product?

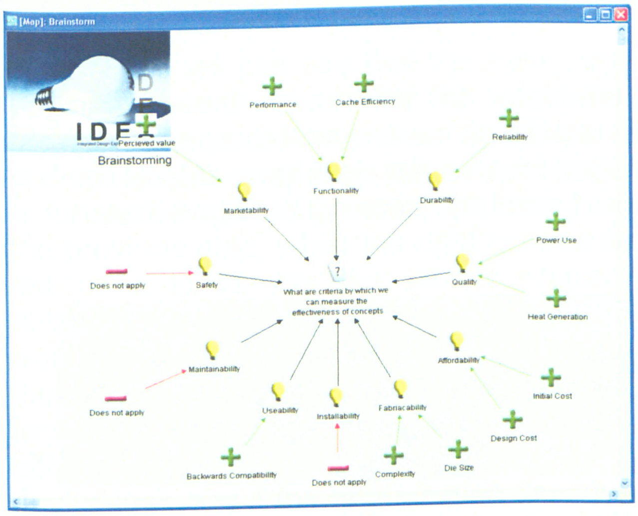
Performance --> Does not have to be bleeding-edge, as long as the speed is adequate to do what I need it to do in a reasonable amount of time.
Heat Generated --> Lower heat increases the reliability of the processor.
Size --> The smaller the size of the processor, the smaller the overall size of the system. This increases the portability of the unit.
No. of Fans Needed --> Fewer fans are more desireable since there is less parts that can fail. It can also be quieter.
Power Use --> Lower power use brings energy savings in terms of dollars and in terms of non-renewable resources.

4) How could similar products that you use today be improved?

I don't know.

A second tool that can be used to determine design characteristics for a design problem is a standardized list of design characteristics that most engineering problems have. A copy of this list can be accessed on the internet by clicking on the **List** icon in the column on the left-hand side of the design characteristics screen. While the list may contain some design characteristics that can be used directly, it is most useful when used as a tool to stimulate thinking for design characteristics that are tailored for the problem at hand. It can be very helpful to simply use the various categories defined in the list and then determine design characteristics for the current design problem.

For example, to generate a suitable list of design characteristics for our sample problem open the brainstorming template by double-clicking on the **Brainstorm** icon on the left-hand side of the design characteristics sub-section window. This opens a blank brainstorming template which can be used to record the results of discussions between team members. Using categories from the standardized list it is then possible to use results generated through brainstorming as well a results obtained through interviews with customers to generate design characteristics for the sample project. Node and link creation follows the same techniques described earlier in this manual. One possible result is shown below.

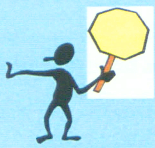


We can see here that several categories have been selected from the standardized list and that many of these categories lead to the definition of design characteristics. We can also see that several of the categories did not generate any design characteristics. These categories were initially included in the brainstorming web because it was felt that they addressed important aspects of the processor design. Upon further reflection however, it was decided that many of these characteristics were unimportant to the design of a solid state circuit. For example, maintainability is unimportant in the design of a processor since it is a solid-state device and is not subject to the type of wear that a user could reduce through maintenance. The remaining categories yielded the following design characteristics for the sample project.

- Performance
- Cache Efficiency
- Reliability
- Power use
- Heat Generation
- Initial Cost
- Design Cost
- Die Size
- Complexity
- Backwards Compatibility
- Perceived Value

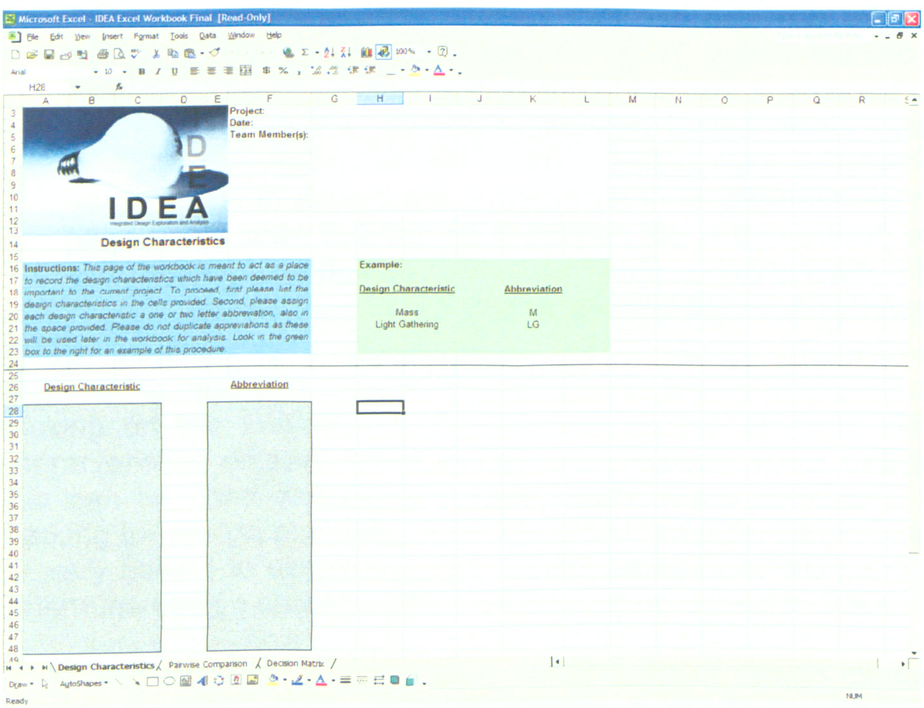
Once you have determined an acceptable set of design characteristics, the final step is to enter them into the **Concept Evaluation Workbook**. The workbook will be used later in the IDEA process to not only calculate the relative importance of the design criteria, but will also assist with the final selection of design concepts. For now, however, we will simply enter the list of design characteristics into the workbook to prepare it for the later stages of concept selection.

First, close the brainstorming template to return to the design characteristics sub-section window. The workbook can now be opened by double-clicking on the **Design Characteristics** icon on the right hand side of the window. The workbook is in Microsoft Excel format, so opening the file will start Excel or whatever spreadsheet program you use.

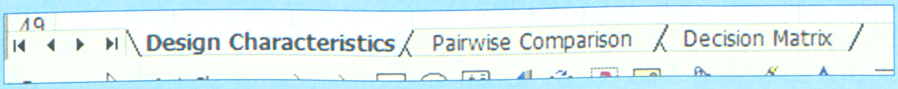


The workbook is used extensively throughout the rest of the IDEA interface, thus it is critical that an appropriate spreadsheet be installed or you will not be able to take advantage many of the steps described in this manual. This manual assumes you are familiar with entering data into a spreadsheet.

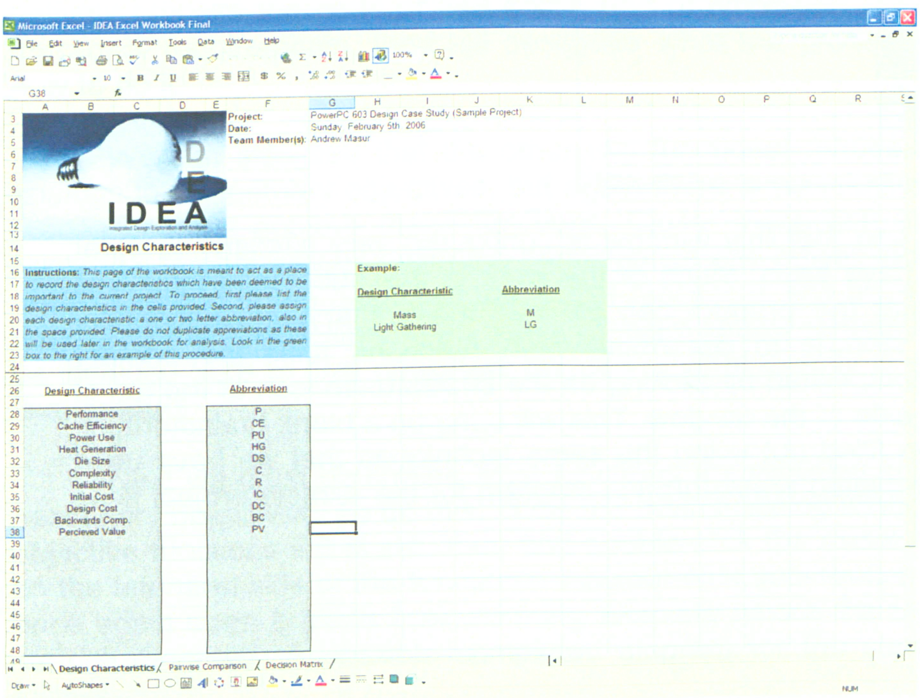
The file will load and you will be presented with the **Design Characteristics** page of the workbook as shown below.



If your screen looks different than above, the most common cause is that the file opened to the wrong page of the workbook. This is easily corrected by clicking on the appropriate tab along the bottom of the screen, in this case the **Design Characteristics** tab as shown below.



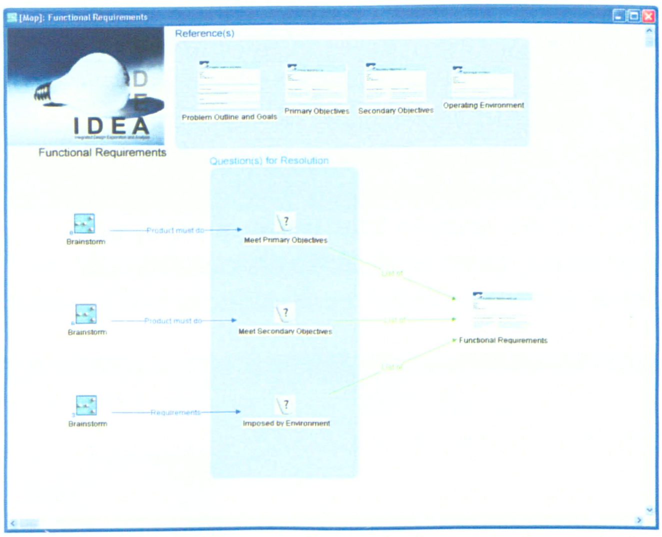
Here we can see that the basic organization of the design characteristics page consists of areas near the top of the page where information such as the project name, date and team members are recorded. The main area of the page is dominated by two grey columns where data is entered. The column on the left is for recording the names of the design characteristics that were determined in the previous step, while the right column is used to record the abbreviations that are used to identify the design characteristics in later steps. Since these abbreviations will be used to identify the design characteristics later in the workbook, ensure that they are clear and easy to remember. A good rule of thumb is to use the first letter in the name of the design characteristic; if there are two words in the name, use the first letter of each word. When generating the abbreviations ensure that you use each abbreviation only once (i.e. no duplicates), and if the abbreviation uses two letters ensure that there is no space between them. Once the data has been entered **save** and close the workbook. The completed page for the sample project is shown below.



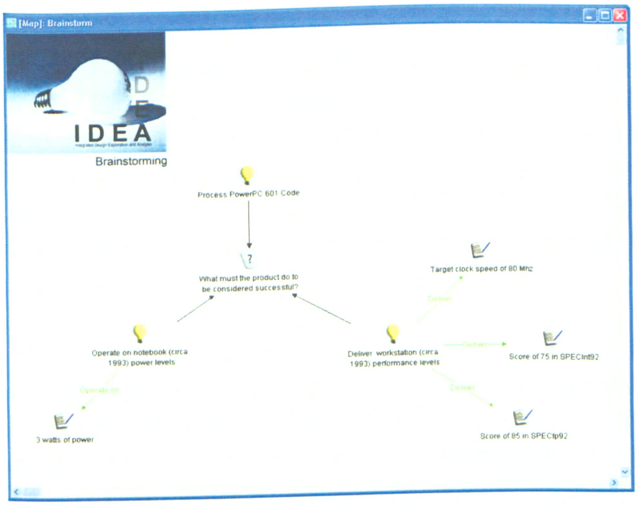
IDEA Step 5: Functional Requirements

Now that we have determined what the product should be, the next step in the IDEA process is to determine what the product should **do**. Up until this point much of the analysis has focused on the physical characteristics of the design; how much it weighs, how much heat it generates, the speed at which it operates etc. Functional requirements are meant to address what actions the product must undertake in order to achieve both the primary and secondary objectives.

The functional requirements tools in IDEA can be accessed by double-clicking on the **Functional Requirements** icon in the light-blue Determine box on the Home Screen. This will bring up the following window.



Again, we can see that this window is configured very similarly to other sub-sections that we have seen. Much of the work in this sub-section involves brainstorming ways that the product can achieve the primary and secondary objectives. The first step is to determine what functions fulfill the primary objectives. This is accomplished by clicking on the **Brainstorm** icon at the top of the column on the left side of the window. This will bring up a blank brainstorming template which can be filled out using the same techniques that we have used before. A possible completed brainstorming diagram for the sample project is as follows.

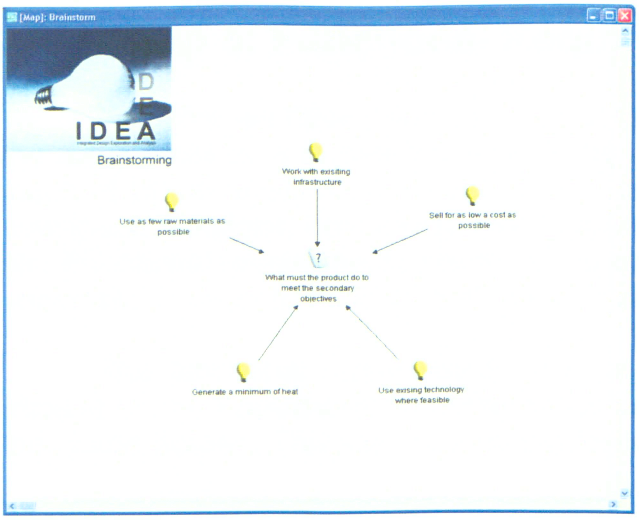


Here we can see that three functions have been defined that the final processor design must perform in order to achieve compliance with the primary objectives.

The first, process PowerPC 601 code, obviously links to the primary objective that the final processor must maintain full backwards compatibility with the existing PowerPC 601 code base. We can see however that there is no mention of how the processor is to accomplish this; for example, whether it will natively process the code or use some kind of emulation. This is the nature of functional requirements, they tell designers what the product must do, but not how to do it.

We can also see that the other two functional requirements have direct analogs in the primary objectives, but again there is no mention of how the final processor must achieve these functions. Closer inspection also reveals some numerical targets included with the performance and power requirements. These were used during the design of the actual processor and were included here as targets for the final design. In an actual problem similar targets, if available, could be included here to give designers an idea of what exactly is expected from the final product.

Next we must determine what functions are required in order to meet the secondary objectives. To do this, close the brainstorming window for the primary objective functions if it is still open and double-click the middle **Brainstorm** icon on the left-hand side of the functional requirements sub-section window. This will once again open a blank brainstorming template where the results of meetings can be recorded. A possible completed diagram for the sample project is as follows.



Once again we can see that many of these functional requirements are directly linked to the secondary objectives. However, closer inspection reveals that many of these functions are of a different nature than those defined for the primary objectives.

Where those objectives dealt with actions that the processor would actively undertake, these are more passive in nature. For example, in the primary functions we stated that the processor would have to process PowerPC 601 code. This is an active function; we are stating that the processor must actively do something.

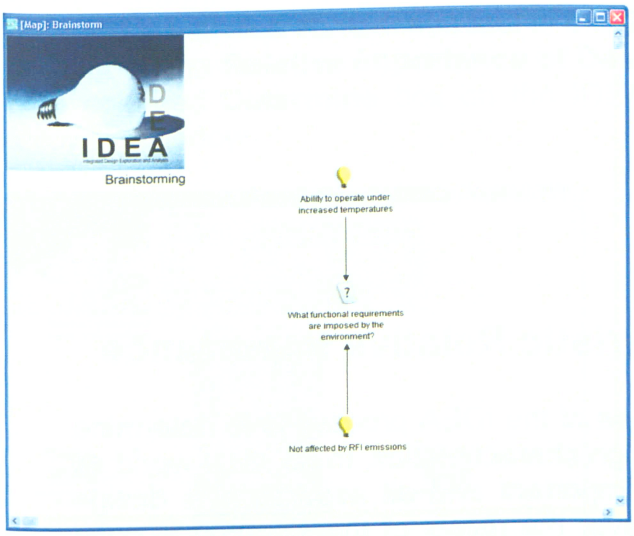
If we look at an example from the secondary objective functions, such as work with existing infrastructure, we see that the functional requirement is not defining something that the processor must actively do. Instead it defines a function that the processor undertakes passively, yet it makes no mention of how the processor is to do this. Whether working with the existing infrastructure means socketing into existing motherboards, or being able to use existing compilers, or both is left up to the designer. In any case, we can see that functional requirements can encompass both active and passive functions and the designer should ensure that such concerns are addressed.

The final class of functional requirements are those that are imposed by the environment. Depending on the operating environment these can have a profound effect on the final design and should be addressed at an early stage.

For example if we were informed that the processor in our sample project would now have to operate in a space environment, the final design would be dramatically altered. In space there are thermal and radiation issues that are simply not encountered on Earth and any electronic components that are designed to work in this environment have to undergo extensive measures to ensure that they are not affected.

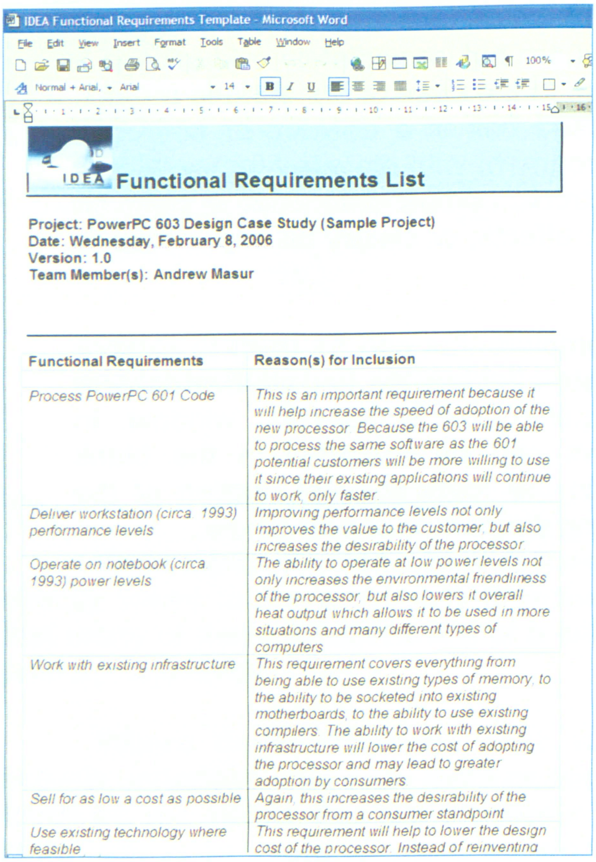
In many cases environmental concerns are not immediately evident and a more thorough investigation is often beneficial. In the case of our sample project, the processor will be working within the closed box of a desktop or notebook computer. This is actually quite a harsh environment with elevated temperatures and high levels of RF interference. Any electronic components used in such an environment must either actively cancel such effects, or simply be impervious to them.

IDEA also contains a brainstorming template that can be used to explore these issues. This can be accessed by **double-clicking** the bottom **Brainstorm** icon on the left side of the functional requirements sub-section window. In addition, any relevant environmental factors can be recorded in the blank **Operating Environment** template located in the **Reference(s)** section of the window. One possible completed brainstorming diagram is shown below.



We can see that the functional requirements here address the environmental issues discussed above.

The final step in determining the functional requirements is to record the information contained in the brainstorming templates for reference later in the IDEA process. Once again, this is accomplished by entering the requirements along with reasons for their inclusion into a blank Word template. The template can be accessed by double-clicking the **Functional Requirements** icon on the right side of the sub-section window. Once the data is entered, **save** the template and close your word processor. A completed template for the sample project is shown below.



IDEA Functional Requirements Template - Microsoft Word

File Edit View Insert Format Tools Table Window Help

IDEA Functional Requirements List

Project: PowerPC 603 Design Case Study (Sample Project)
Date: Wednesday, February 8, 2006
Version: 1.0
Team Member(s): Andrew Masur

Functional Requirements	Reason(s) for Inclusion
Process PowerPC 601 Code	*This is an important requirement because it will help increase the speed of adoption of the new processor. Because the 603 will be able to process the same software as the 601 potential customers will be more willing to use it since their existing applications will continue to work, only faster.*
Deliver workstation (circa. 1993) performance levels	*Improving performance levels not only improves the value to the customer, but also increases the desirability of the processor.*
Operate on notebook (circa 1993) power levels	*The ability to operate at low power levels not only increases the environmental friendliness of the processor, but also lowers it overall heat output which allows it to be used in more situations and many different types of computers.*
Work with existing infrastructure	*This requirement covers everything from being able to use existing types of memory, to the ability to be socketed into existing motherboards, to the ability to use existing compilers. The ability to work with existing infrastructure will lower the cost of adopting the processor and may lead to greater adoption by consumers.*
Sell for as low a cost as possible	*Again, this increases the desirability of the processor from a consumer standpoint.*
Use existing technology where feasible	*This requirement will help to lower the design cost of the processor. Instead of reinventing*

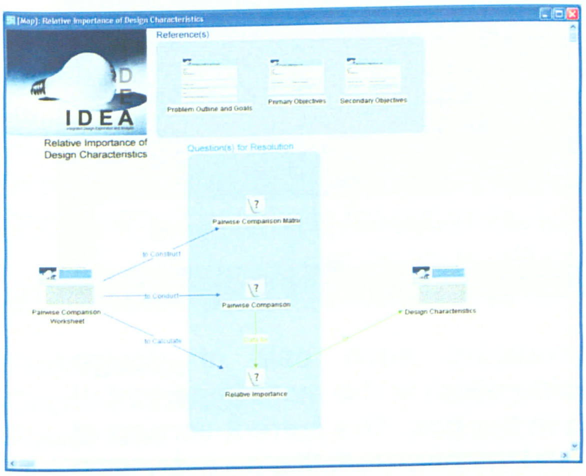
IDEA Step 6: Design Characteristic Relative Importance

The final task in the determine phase of the IDEA process is to determine the relative importance of the various design characteristics. In an ideal world all design characteristics would be equally important and all aspects of a design would be as optimized as possible, however the reality of the situation is quite different. Due to factors such as budget and time constraints, or unforeseen changes in project requirements, the design characteristics for a project are not equally important. Certain characteristics are more important than others and trading off one characteristic for the benefit of another is a common practice.

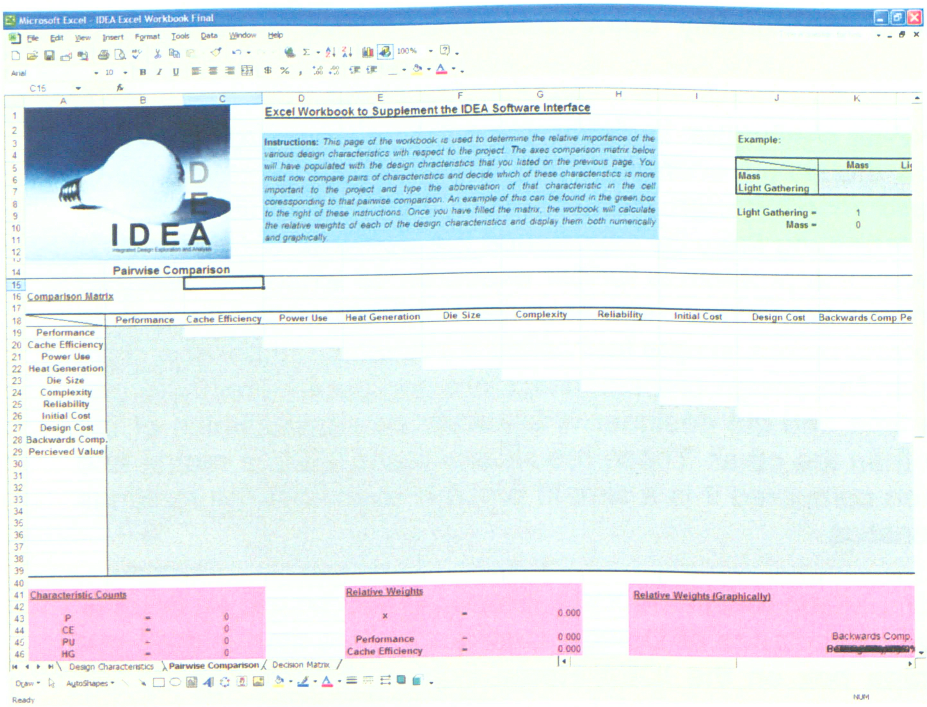
One of the challenges of concept design is determining just how much more or less important a characteristic is than another. If the relative importance of the design characteristics is incorrect it is possible to design a product that is optimized in completely the wrong way. For example, an engineering firm is asked to design a widget-making machine for a company with a small factory. The firm decides that the widget-making speed of the machine is of prime importance, to the detriment of all other factors. The machine is delivered to the factory and it is discovered that it will not fit in the available space. The machine may be the fastest widget-maker on the planet, but it is useless to the company since it cannot be used in the factory. This is an extreme case, but it does illustrate the dangers of inaccurately determining the relative importance of design characteristics.

One of the greatest challenges in determining the relative importance of design characteristics is the simple enormity of the task. It is not uncommon for a design problem to have many design characteristics that are under scrutiny. Faced with the task of arranging them by importance the designer will often try to do the whole list at once. The problem is further exacerbated when the designer is then asked to determine *how much* more important one characteristic is than another. The IDEA process uses a simple tool called a pairwise comparison chart to solve these problems. The design characteristics are compared directly to one another, one pair at a time and the designer is asked to determine which of the pair is more important than the other. These results are recorded in a matrix and after all pairs have been compared it is a simple process to extract the rankings of the various characteristics.

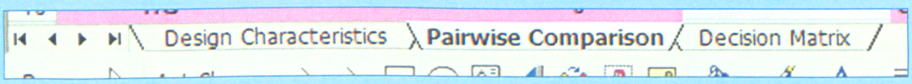
The pairwise comparison tool in the IDEA interface is accessed by double-clicking on the **Relative Importance of Design Characteristics** icon in the light-blue coloured **Determine** box on the IDEA Home Screen. This brings up the following window.



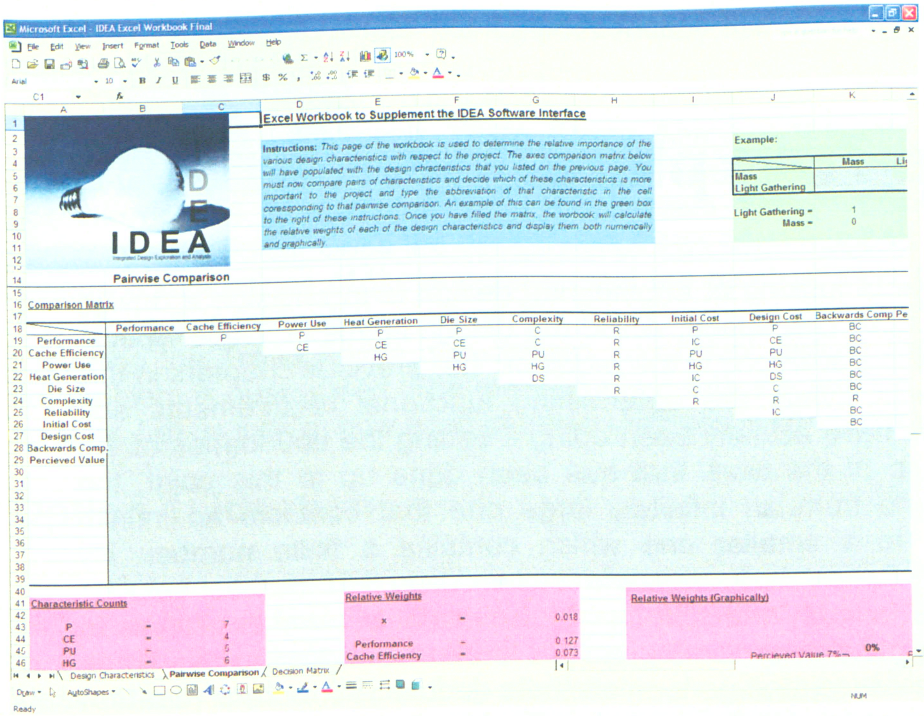
Double-click on the **Pairwise Comparison Worksheet** icon on the left side of the screen to open the IDEA Concept Evaluation Workbook which contains the pairwise comparison tool. This is the same workbook where the design characteristics were entered in a previous step. Normally setting up a pairwise comparison chart is a time consuming process, but because the design characteristics have already been inputted into the workbook it will create the matrix automatically. A blank chart which has been automatically set up for the sample project is shown below.



If your screen looks different than above, the most common cause is that the file opened to the wrong page of the workbook. This is easily corrected by clicking on the appropriate tab along the bottom of the screen, in this case the **Pairwise Comparison** tab as shown below.



Fill the comparison matrix by comparing each pair of design characteristics where there is a white box. Decide which is the more important of the characteristics and record it's abbreviation in the box. The current version of the workbook does not support ties between design characteristics so a decision must be reached as to which characteristic is most important for that pairing. Continue this procedure until the matrix has been filled in. An example of a completed pairwise comparison matrix for the PowerPC 603 project is shown below.



The three red shaded boxes along the bottom of the worksheet display the results of calculations the worksheet performs in order to determine the relative importance of the characteristics based upon the contents of the matrix. The leftmost box displays the number of times each characteristic appears in the matrix. The center box displays a numerical representation of the relative importance of the various criteria while the rightmost box displays the same data graphically. The results for our sample project are as follows.

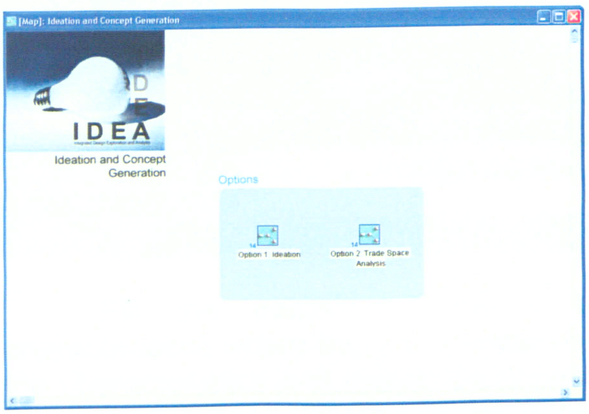
Design Characteristic	Weight
Performance	0.127
Cache Efficiency	0.073
Power Use	0.091
Heat Generation	0.109
Die Size	0.036
Complexity	0.073
Reliability	0.182
Initial Cost	0.055
Design Cost	0.018
Backwards Compatibility	0.164
Perceived Value	0.073

Now that the relative weights have been calculated, **save** the workbook and close the spreadsheet problem. Close the Relative Importance of Design Characteristics sub-section window to return to the Home Screen.

Chapter 7

IDEA Step 7: Explore

One of the ways of looking at engineering design is to envision a design problem as a large space. This space, called the design space or Trade Space, contains all of the possible solutions to the problem. The previous chapters in this manual have dealt with things like determining functional requirements and design goals. What we have actually been doing it setting the boundaries of the design space. Because of the work that has been done up to this point, the design space has gone from an infinitely large one that contains an infinite number of solutions, to a smaller one which contains a finite number. By constraining the Trade Space we have made it more likely that the best solution to the problem can be found. The IDEA interface contains a number of tools to assist the designer in generating concepts, also known as exploring the Trade Space. To access the conceptual generation tools double-click on the **Ideation and Concept Generation** icon in the light-blue **Explore** box at the bottom of the Home Screen. The following window will appear.



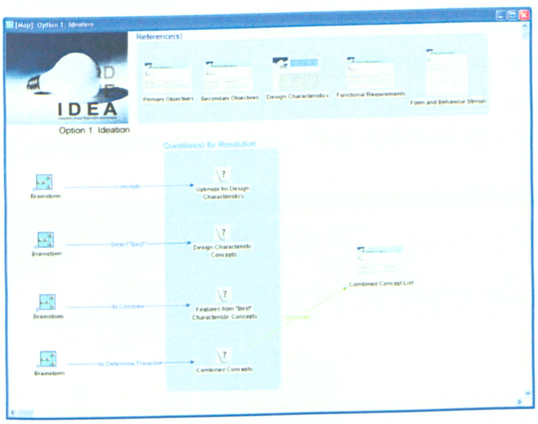
We can see here that IDEA offers two main options for conducting concept generation; Ideation and Trade Space Analysis. While both are designed to help zero-in on the best possible solution they do so in completely different ways.

Ideation is more of an intuitive process where concepts are generated such that each concept is designed to optimize for one of the design characteristics at the expense of all others. While none of these designs are well balanced each of them does one thing very well. All of these concepts are then examined in an attempt to combine them together in various configurations. The goal is to combine many concepts that are optimized for only one design characteristic, into a single combined concept that excels in several design characteristics.

The second approach, Trade Space Analysis, requires the designers to look at concepts from a functional level. The designers attempt to determine all possible options for each functional requirement, thus mapping out the design space in great detail. Once the design space has been mapped out designers try various combinations of these options until the most suitable concept is found. The Trade Space Analysis method has the advantage that given ample time and assuming the Trade Space was defined accurately, finding the best solution to the problem is guaranteed. The main problem with the method is that it tends to be slow. Let us look at how each of these methods is conducted in turn.

Option 1: Ideation

Due to the nature of the information provided for the PowerPC sample project, it was not feasible to conduct an ideation analysis to determine concepts for the sample project. Instead this section will simply serve a guide on how the ideation tools work and how they are used in the design process. To access the ideation tools double-click on the **Option 1: Ideation** icon in the Ideation and Concept Generation sub-section window. The following window will appear.



As in many of the sub-sections that have been covered up to this point, the primary tool used in ideation is brainstorming. The basic procedure for ideation is as follows.

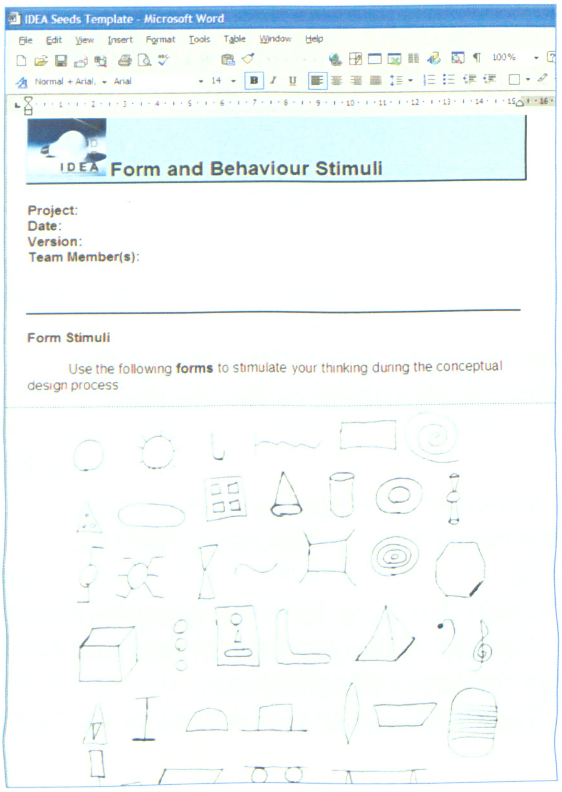
1.) Assign each team member a design characteristic to optimize. If there are more characteristics than team members some team members will be responsible for more than one characteristic.
2.) Each team member generates concepts which optimize for their assigned characteristic. Team members with multiple characteristics should generate separate sets of concepts for each design characteristic.
3.) The team then reunites and selects the "best" concepts that optimize for each design criteria.

4.) Attempt to combine the attributes from the remaining concepts to create integrated concepts which are optimized for multiple design characteristics.

5.) Select the most plausible of the combined concepts to undergo further analysis.

All of the brainstorming is conducted using the node and link creation methods described earlier in the manual. Simply start with the brainstorming icon at the top of the left-hand column and work down to complete the ideation process. Once the most plausible combined concepts have been generated enter them into the concept template by double-clicking on the **Combined Concept List** icon on the right side of the ideation window. Because the template will open in your word processor you must use the appropriate command for that program to **save** the changes to the template so that they can be used as reference later in the IDEA process.

One other item of note in the main ideation sub-section window is the icon labelled **Form and Behaviour Stimuli** in the light-grey references box. It has been shown that creative thought can be stimulated through the use of simple images which vaguely depict both mechanical form and function. These plant idea seeds in the designers' mind which can then be fleshed out to achieve a more complete design. Double-clicking on this icon opens a document which contains several pages of these stimuli. A section of this document is shown below.



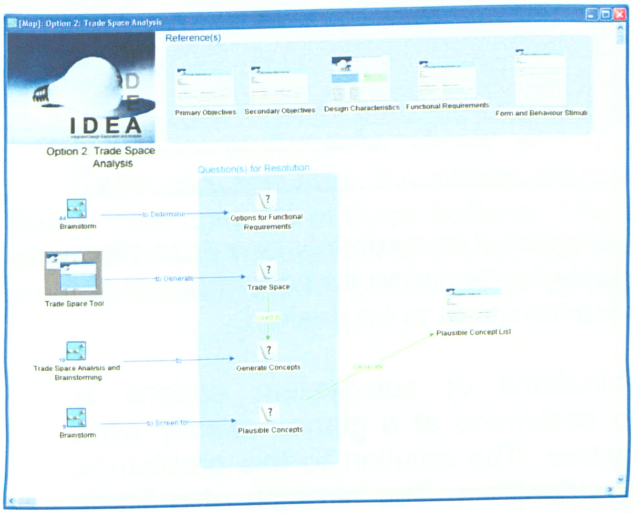
Option 2: Trade Space Analysis

Of the two options available in the concept generation process, Trade Space Analysis is the most straightforward and the most time consuming. Trade Space Analysis is best used when designers have a good idea of the various options that can be used for functional requirements in the final product or if a company uses the same options in various configurations between different projects. Alternatively, if the sub-systems within a product are well understood designers can use these in place of the functional requirements.

Because the performance of these options is well understood, designers are able to constrain the design space very quickly by simply using their experience. The IDEA interface includes several tools to further speed up the Trade Space Analysis process, making it even more desirable.

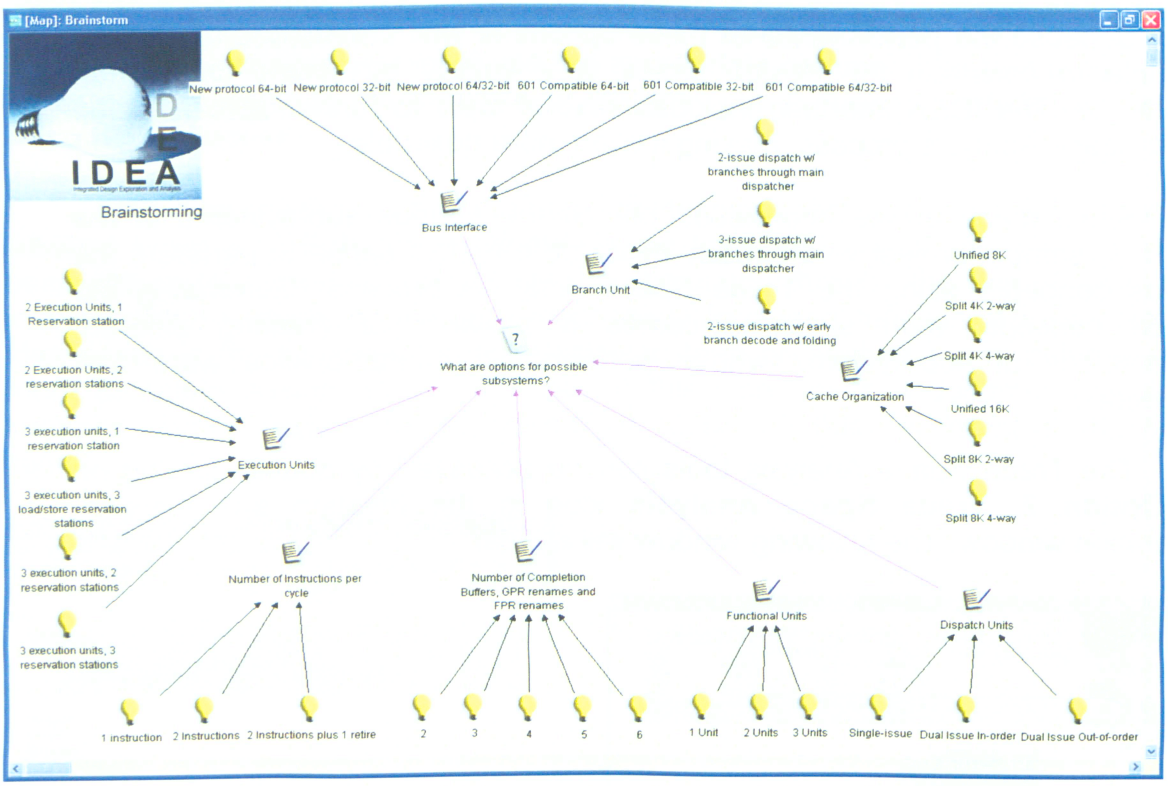
Since much of the information behind the various sub-systems in the PowerPC sample project was already available it makes sense to generate the various concepts using the Trade Space Analysis process. Generating the various concepts then simply becomes a problem of creating different combinations of sub-systems whose performance characteristics were well understood.

The Trade Space Analysis tools can be accessed by double-clicking on the **Option 2: Trade Space Analysis** icon in the Ideation and Concept Generation window. The following window will appear.



The first step in the Trade Space Analysis process is to determine options for the various sub-systems in the product. Normally this is accomplished using brainstorming and other similar techniques, but it is also possible that a company will have a standardized set of options that they use for various sub-systems.

In the case of the PowerPC sample project detailed information was available on the various options that designers investigated in order to determine the optimum combination of sub-systems for the processor. Regardless of the source of the information it can be recorded by double-clicking on the **Brainstorm** icon at the top of the leftmost column in the sub-section window. This is will open the standard brainstorming template that we have seen many times in this manual. Nodes and links can be created using the standard techniques described earlier. A completed template for the sample project is shown below.

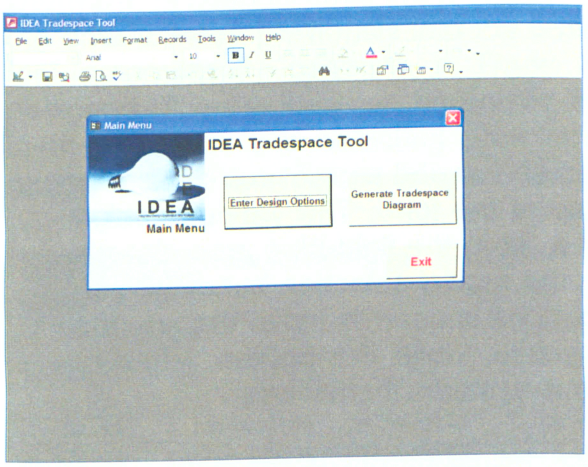


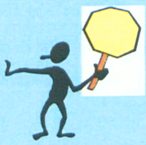
Looking at the above brainstorming diagram it immediately becomes clear that while it has efficiently packed a great deal of information into a comparatively small space, the readability of the diagram leaves much to be desired.

While determining different combinations of sub-system options is possible from this diagram it is difficult to determine at a glance exactly what options are available and how they fit together. The solution to this problem is found in what is known as a Trade Space Diagram. The strength of a Trade Space Diagram lies in the fact that it displays each of the sub-systems and their relevant options in a manner which makes it possible to generate different combinations of sub-systems at a glance. While generating the diagram is not difficult, it is quite a tedious and time consuming process. To overcome this deficiency the IDEA interface includes a tool which automatically generates a

Trade Space Diagram from a set of user specified options. In order to demonstrate the use of this tool we will generate a diagram based upon the information contained in the above brainstorming diagram.

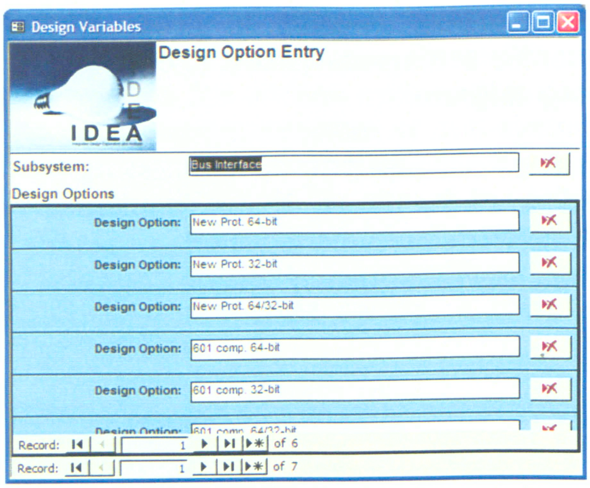
The Trade Space Diagram generator can be accessed by double-clicking the **Trade Space Tool** icon on the left-hand side of the Trade Space Analysis window. The tool is in Microsoft Access format and will be opened in the appropriate database program for your computer. The main menu is shown below.



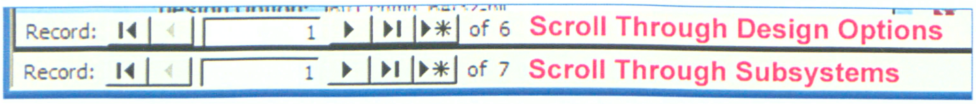


While use of the tool is recommended it is not required since many of the same techniques can be applied to generate Trade Space Diagrams manually. The use of the tool requires a database program, therefore the installation of such a program is highly recommended. The Trade Space Tool has only been tested in Microsoft Access. While it is possible that it may work in other database programs, functionality is not guaranteed.

We can see from the above diagram that the main menu for the Trade Space Tool consists of three buttons. The first of these, **Enter Design Options**, allows the user to enter all of the design options for the various sub-systems in a product. Clicking on the button brings up the following window.



As we can see here the Design Option Entry form lists the current sub-system in the box contained in the white area at the top of the form. The various design options for that sub-system are displayed as a list in the light-blue area directly underneath. The lists are navigated by using the record selection buttons along the bottom of the form. The upper group of buttons is used to move through the list of design options for a particular sub-system, while the bottom group of buttons is used to scroll through the different sub-systems themselves.



Entering design options into the Trade Space Tool is accomplished through the following simple procedure.

1.) Click on the **Enter Design Options** button on the Trade Space Tool main menu.
2.) Click in the box next sub-system label in the white area at the top of the Design Options Entry form.
3.) Enter the name of the sub-system in the box.
4.) Click on the white box next to the topmost Design Options label in the light-blue area of the form.
5.) Enter the design option and press the **Tab** key to move to the next design option box.
6.) Repeat step five until all design options for that particular sub-system have been entered.
7.) When the final design option for a particular sub-system has been entered click on the ▸ button to move to the next sub-system.
8.) Repeat steps 2-7 for each additional sub-system

If you make a mistake entering the design options into the form, simply click on the ✗ button next to the erroneous entry to delete it from the form. When all sub-systems and design options have been entered simply close the form to return to the main menu.

To delete a sub-system from the Trade Space Tool you must first delete all related design options in the light-blue area directly below the sub-system name. The sub-system may then be deleted by using the button next to the sub-system name.

The second button of interest on the main menu, **Generate Trade Space Diagram**, is used to have the tool generate a Trade Space Diagram based upon the data entered previously into the form. Simply click on the button to have the tool automatically generate a Trade Space Diagram for the product being designed. The diagram can be printed by simply right-clicking on an empty area in the diagram and selecting print from the pop-up menu that appears. An example of a Trade Space Diagram for the sample project is shown below.

Trade Space Diagram

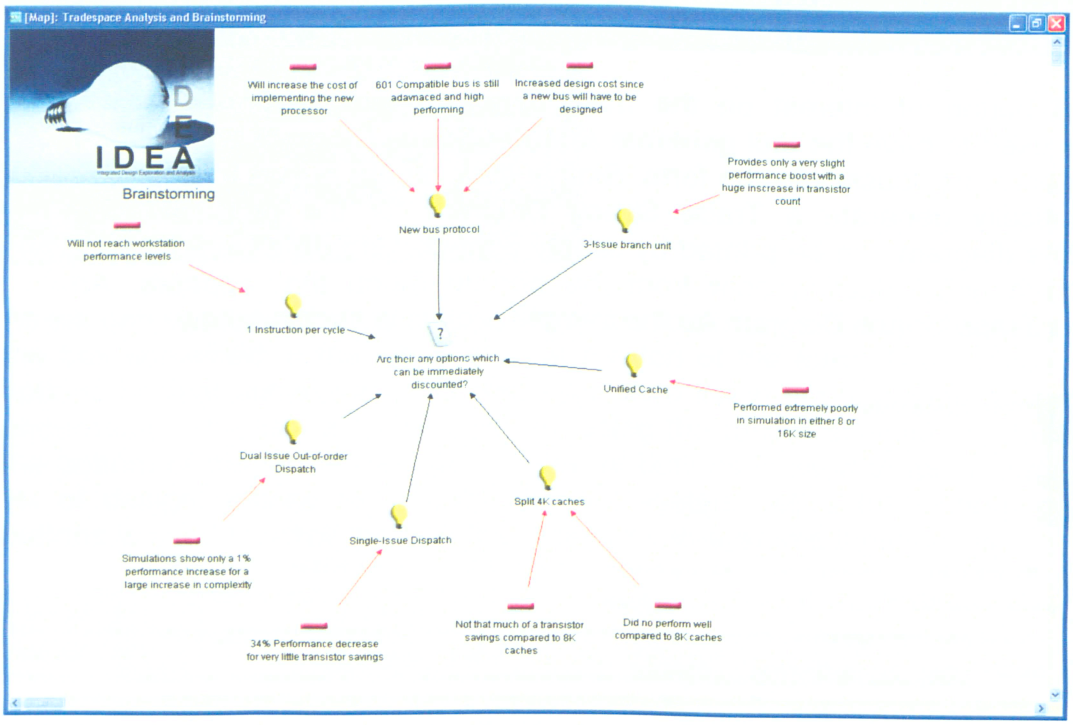
Bus Interface	601 comp. 64-bit	601 comp. 32-bit	601 comp. 64 32-bit	New Prot. 32-bit	New Prot. 64 32-bit	New Prot. 64-bit
Branch Unit	ssue early decode and f	2-Issue through main	3-issue through main			
Cache Organizati	Split 4K 2-way	Split 4K 4-way	Split 8K 2-way	Split 8K 4-way	Unified 16K	Unified 8K
Dispatch Units	Dual Issue In-Order	Dual Issue Out-of-order	Single-Issue			
No. of Completio	2	3	4	5	6	
Instructions per	1 Instruction	2 Instructions	2 Instruction - 1 Retire			
Execution Units	2 - 1 Reservation	2 - 2 Reservation	3 - 3 Reservation	3 - 1 Reservation	3 - 3 Load store	3 - 2 Reservation

To exit the Trade Space Tool simply close the diagram window and click on the **Exit** button in the main menu.

Upon closer inspection of the Trade Space Diagram one thing immediately becomes clear; for even a modestly complex problem the number of concepts is enormous. In the case of our sample project there are a total of 29160 possible concepts represented in the Trade Space Diagram. Clearly while the size of the Trade Space has been reduced because of the previous steps in the IDEA process, it is still very large.

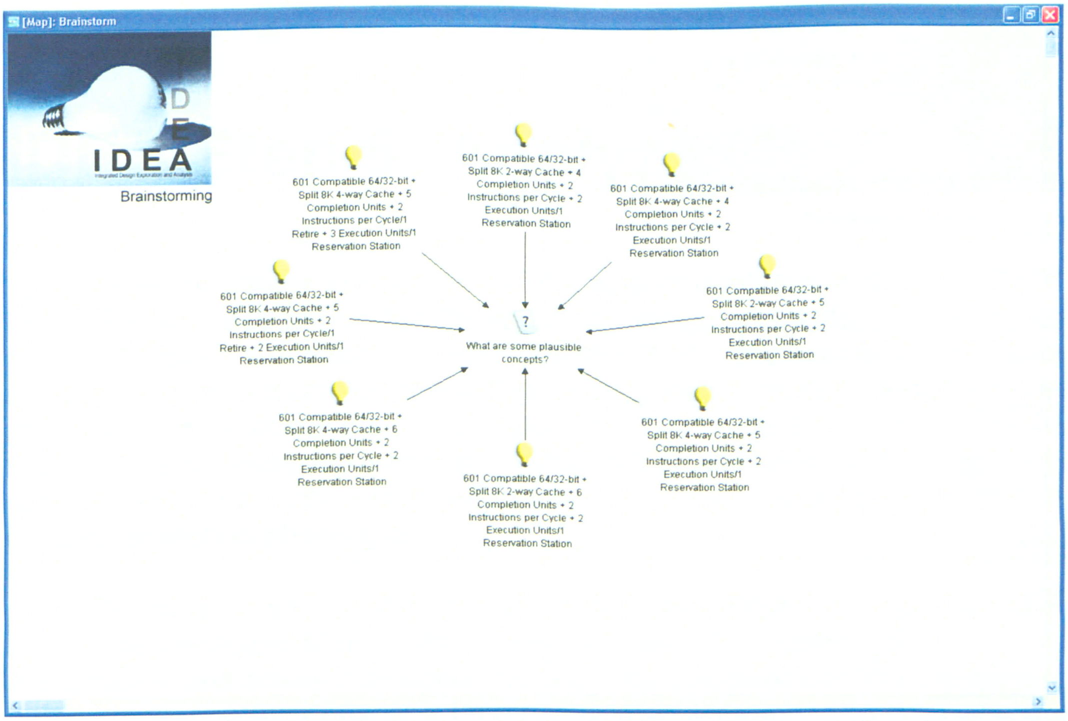
Thus the next step in the Trade Space Analysis is to further reduce the size of the Trade Space by looking at the options available and deciding which can be discounted outright, or looking for options that may not work with one another, eliminating large numbers of alternatives.

A simple way of doing this is to once again conduct brainstorming sessions to determine which options can be removed and which options will remain. The IDEA interface includes a brainstorming template for this purpose which can be access by double-clicking on the **Trade Space Analysis and Brainstorming** icon in the Trade Space Analysis sub-section window. An example of a completed brainstorming diagram for the sample project is shown below.



Here we can see that a number of options have been discounted for the various reasons stated in the diagram. The most common reason for the removal of options was that the proposed option simply did not provide enough performance when compared to the others, or that a design option increased the complexity of the design too much for the performance increase that it provided. The removal of these options lowers the total number of possible concepts to 720, still a large number, but an enormous improvement over the previous total.

The final step in the Trade Space Analysis process is to actually generate viable concepts for further analysis. Concepts are generated by looking at the remaining options in the Trade Space Diagram and choosing various combinations to create viable alternatives. This is typically done through brainstorming sessions and the IDEA interface includes a brainstorming template for this purpose. This template is accessed by double-clicking on the **Brainstorm** icon at the bottom of the leftmost column in the Trade Space Analysis window. An example of a completed diagram for the sample project is shown below.



Now that the concepts have been generated they must be recorded for use in the concept selection phase. To do this, double-click on the **Plausible Concept List** icon on the right side of the Trade Space Analysis window to open a blank template in your computers word processor. Fill in the template and **save** the changes. The completed document for the sample project is shown below.

IDEA **Plausible Concepts List**

Project: PowerPC 603 Design Case Study (Sample Project)
Date: Saturday, February 18, 2006
Version: 1.0
Team Member(s): Andrew Masur

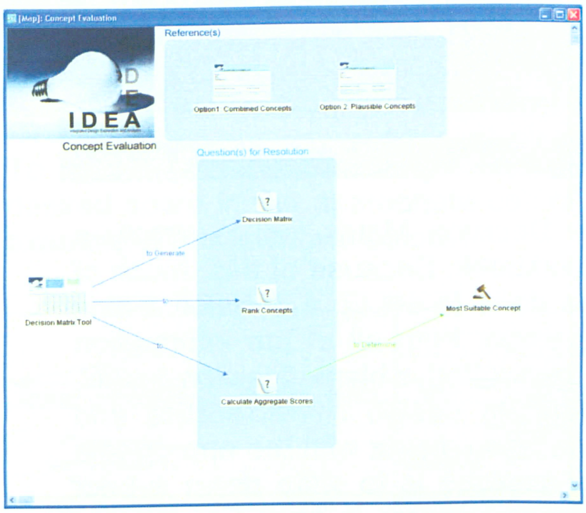
Brief Description of Concept	Image (If available)
Concept A: *601 Compatible 64/32-bit + Split 8K 2-way Cache + 4 Completion Units + 2 Instructions per Cycle + 2 Execution Units/1 Reservation Station*	*N/A*
Concept B: *601 Compatible 64/32-bit + Split 8K 4-way Cache + 4 Completion Units + 2 Instructions per Cycle + 2 Execution Units/1 Reservation Station*	*N/A*
Concept C: *601 Compatible 64/32-bit + Split 8K 2-way Cache + 5 Completion Units + 2 Instructions per Cycle + 2 Execution Units/1 Reservation Station*	*N/A*
Concept D: *601 Compatible 64/32-bit + Split 8K 4-way Cache + 5 Completion Units + 2 Instructions per Cycle + 2 Execution Units/1 Reservation Station*	*N/A*
Concept E: *601 Compatible 64/32-bit + Split 8K 2-way Cache + 6 Completion Units + 2 Instructions per Cycle + 2 Execution Units/1 Reservation Station*	*N/A*
Concept F: *601 Compatible 64/32-bit + Split 8K 4-way Cache + 6 Completion Units + 2 Instructions per Cycle + 2 Execution Units/1 Reservation Station*	*N/A*
Concept G: *601 Compatible 64/32-bit + Split 8K 4-way Cache + 5 Completion Units + 2 Instructions per Cycle/1 Retire + 2 Execution*	*N/A*

Chapter 8

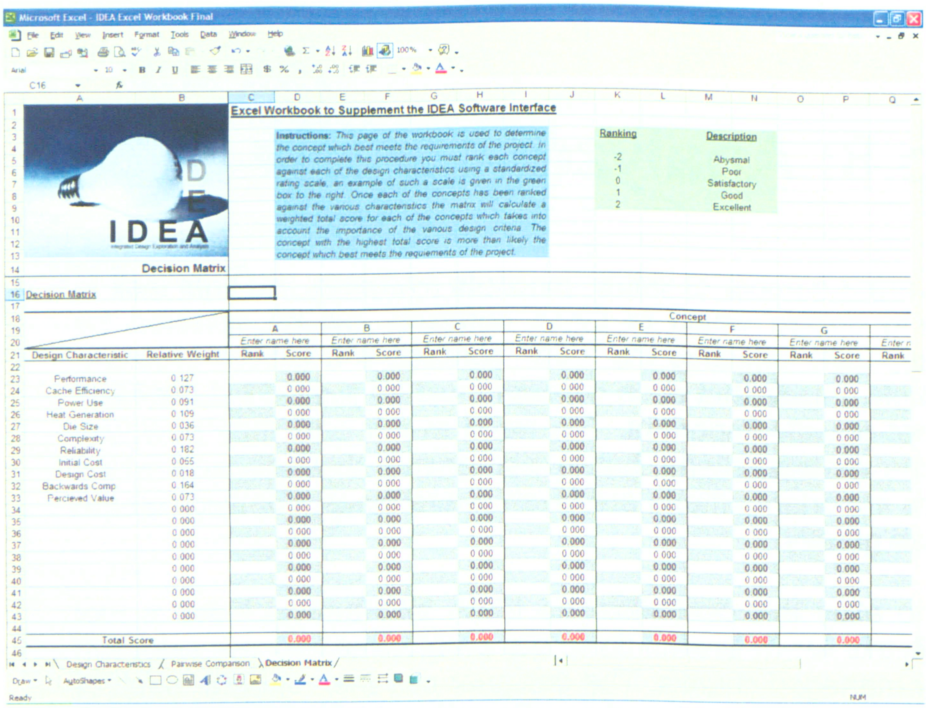
IDEA Step 8: Analyze

Now that there are a number of viable concepts available, the final task in the IDEA process is to decide which of these concepts will be selected for detailed design. Conceptually selecting the "best" concept is a difficult problem. Not only are their multiple criteria that each concept must be judged by, but each of these criteria are not equally important. The need to not only rank how well each concept meets the various criteria, but to also ensure that concepts which rank better in the more important criteria are given precedence, complicates the concept selection phase greatly.

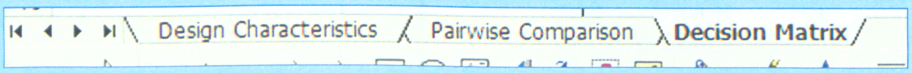
In an attempt to allay this confusion the IDEA interface includes a tool called the **Decision Matrix** which breaks down the problem of concept selection into smaller and easily managed chunks. The designer can then analyze each of these smaller chunks one at a time rather than attempting to deal with the whole problem at once. To begin, the concept evaluation tools can be accessed by double-clicking the **Concept Evaluation** icon in the light-blue **Analyze** box on the Home Screen. The following window will appear.



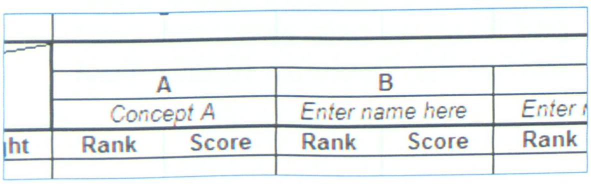
We can see that unlike many of the other sub-sections in the IDEA interface there is only one main tool used in the concept evaluation process, the **Decision Matrix Tool**. This tool is used to generate the decision matrix itself, assist with the ranking of the various criteria, and finally to help select the most suitable concept. To access the tool double-click on the **Decision Matrix Tool** icon. The following window will appear.



If your screen looks different than above, the most common cause is that the file opened to the wrong page of the workbook. This is easily corrected by clicking on the appropriate tab along the bottom of the screen, in this case the **Decision Matrix** tab as shown below.



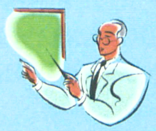
We can see from the above that the Decision Matrix tool is simply a worksheet in the IDEA Concept Evaluation Workbook. Because of this, much of the data entry that would normally have to be done to set up a decision matrix has already been taken care of in previous steps. With all of the information already available the worksheet has not only generated a blank decision matrix, but has already populated it with the appropriate design characteristics and relative weights. All that remains is to fill the decision matrix with the appropriate concepts and rankings. The first step in this process is to write down a brief name for each concept along the top of the matrix where is says *Enter name here.* This will serve to organize the matrix and will make entering the appropriate data easier. An example is shown below.



In our sample project we will simply use the names Concept A through H to designate the various concepts since the makeup of each concept does not lend itself to a simple description.

With the concept names entered into the matrix the next step is to actually rank how well each concept meets each of the design characteristics. While there are many methods that can be implemented to rank the design characteristics, IDEA utilizes the concept of ranking each of the design characteristics against a common ranking scale. This helps to ensure that there is no bias in the rankings since each of the concepts will use the exact same scale and be judged to the same standards. The scale used in the IDEA process is similar to that which is commonly found in the automotive industry and is as follows.

Ranking	Description
-2	Abysmal
-1	Poor
0	Satisfactory
1	Good
2	Excellent



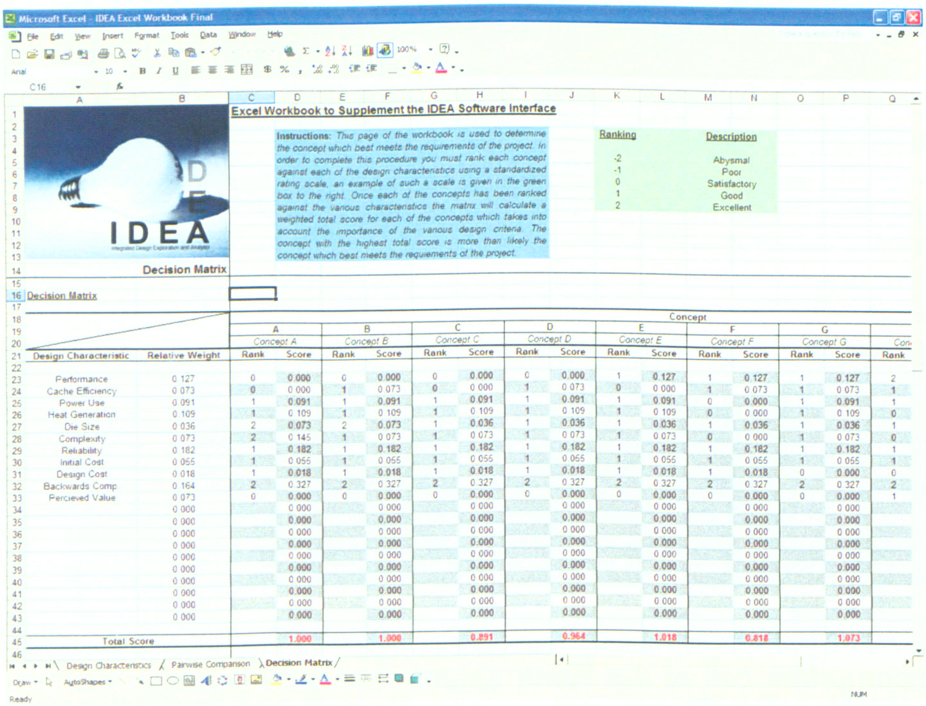
Another common method of ranking the design characteristics is to select one of the concepts and set it as the baseline. All of its design characteristics are given a score of 0, and the rest of the concepts are ranked how well they perform against the baseline.

To input the various rankings for the design characteristics simply enter the desired score in the appropriate cell of the **Rank** column for each concept. An example of this is shown below.

		A	
		Concept A	
Design Characteristic	**Relative Weight**	**Rank**	**Score**
Performance	0.127	0	0.000
Cache Efficiency	0.073	0	0.000
Power Use	0.091	1	0.091
Heat Generation	0.109	1	0.109
Die Size	0.036	2	0.073
Complexity	0.073	2	0.145
Reliability	0.182	1	0.182
Initial Cost	0.055	1	0.055
Design Cost	0.018	1	0.018
Backwards Comp.	0.164	2	0.327
Percieved Value	0.073	0	0.000
	0.000		0.000

The **Score** column is generated by simply taking the value entered next to it in the rank column and multiplying it by the relative weight of the design characteristic. This takes into account not only how well each concept meets each design characteristic, but also the importance of each design characteristic to the overall design. To complete the decision matrix simply fill in the rank column for each of the concepts under consideration.

An example of a completed decision matrix for the sample project is shown below. All of the values used in the rankings were derived from performance data that was generated during the actual design of the PowerPC 603 processor, thus these rankings are not guesses, but are based on actual verifiable data from real design project.



Microsoft Excel - IDEA Excel Workbook Final

Excel Workbook to Supplement the IDEA Software Interface

IDEA — Decision Matrix

Instructions: *This page of the workbook is used to determine the concept which best meets the requirements of the project. In order to complete this procedure you must rank each concept against each of the design characteristics using a standardized rating scale, an example of such a scale is given in the green box to the right. Once each of the concepts has been ranked against the various characteristics the matrix will calculate a weighted total score for each of the concepts which takes into account the importance of the various design criteria. The concept with the highest total score is more than likely the concept which best meets the requirements of the project.*

Ranking	Description
-2	Abysmal
-1	Poor
0	Satisfactory
1	Good
2	Excellent

Decision Matrix

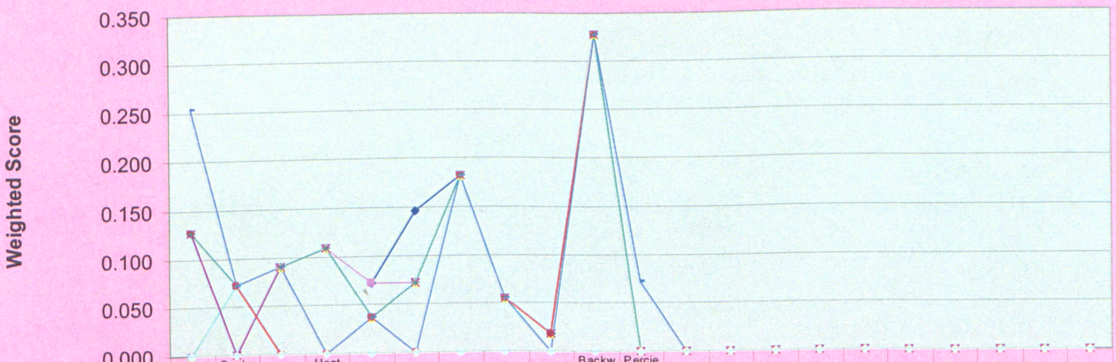
Design Characteristic	Relative Weight	A Concept A Rank	A Score	B Concept B Rank	B Score	C Concept C Rank	C Score	D Concept D Rank	D Score	E Concept E Rank	E Score	F Concept F Rank	F Score	G Concept G Rank	G Score	Con... Rank
Performance	0.127	0	0.000	0	0.000	0	0.000	0	0.000	1	0.127	1	0.127	1	0.127	2
Cache Efficiency	0.073	0	0.000	1	0.073	0	0.000	1	0.073	0	0.000	1	0.073	1	0.073	1
Power Use	0.091	1	0.091	1	0.091	1	0.091	1	0.091	1	0.091	0	0.000	1	0.091	1
Heat Generation	0.109	1	0.109	1	0.109	1	0.109	1	0.109	1	0.109	0	0.000	1	0.109	0
Die Size	0.036	2	0.073	2	0.073	1	0.036	1	0.036	1	0.036	1	0.036	1	0.036	1
Complexity	0.073	2	0.145	1	0.073	1	0.073	1	0.073	1	0.073	0	0.000	1	0.073	0
Reliability	0.182	1	0.182	1	0.182	1	0.182	1	0.182	1	0.182	1	0.182	1	0.182	1
Initial Cost	0.055	1	0.055	1	0.055	1	0.055	1	0.055	1	0.055	1	0.055	1	0.055	1
Design Cost	0.018	1	0.018	1	0.018	1	0.018	1	0.018	1	0.018	1	0.018	0	0.000	0
Backwards Comp	0.164	2	0.327	2	0.327	2	0.327	2	0.327	2	0.327	2	0.327	2	0.327	2
Percieved Value	0.073	0	0.000	0	0.000	0	0.000	0	0.000	0	0.000	0	0.000	0	0.000	1
	0.000		0.000		0.000		0.000		0.000		0.000		0.000		0.000	
	0.000		0.000		0.000		0.000		0.000		0.000		0.000		0.000	
	0.000		0.000		0.000		0.000		0.000		0.000		0.000		0.000	
	0.000		0.000		0.000		0.000		0.000		0.000		0.000		0.000	
	0.000		0.000		0.000		0.000		0.000		0.000		0.000		0.000	
	0.000		0.000		0.000		0.000		0.000		0.000		0.000		0.000	
	0.000		0.000		0.000		0.000		0.000		0.000		0.000		0.000	
	0.000		0.000		0.000		0.000		0.000		0.000		0.000		0.000	
	0.000		0.000		0.000		0.000		0.000		0.000		0.000		0.000	
	0.000		0.000		0.000		0.000		0.000		0.000		0.000		0.000	
Total Score			1.000		1.000		0.891		0.964		1.018		0.818		1.073	

Design Characteristics / Pairwise Comparison / Decision Matrix

Once all of the concepts have been ranked the worksheet will calculate aggregate scores for each of them. This score is generated simply by adding up the values in the weighted score column for each concept. The concept, or concepts, with the highest total score best meet the parameters of the design problem and would be the optimum choice for further detailed design. If two scores are very close than either of them would be a good choice for the project and the final selection becomes a matter of personal preference.

One useful feature of the decision matrix worksheet is that it generates a graphical depiction of the strengths and weaknesses of each concept relative to the others. Examination of this graph may show desirable trends in one of the concepts under consideration assisting with the final selection. The graph is located in the red tinted box directly under the decision matrix in the worksheet. An example of a graph generated for the sample project is shown on the following page.

Graphical Representation



	Performance	Cache Efficiency	Power Use	Heat Generation	Die Size	Complexity	Reliability	Initial Cost	Design Cost	Backwards Comp.	Percieved Value										
Concept A	0.000	0.000	0.091	0.109	0.073	0.145	0.182	0.055	0.018	0.327	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Concept B	0.000	0.073	0.091	0.109	0.073	0.073	0.182	0.055	0.018	0.327	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Concept C	0.000	0.000	0.091	0.109	0.036	0.073	0.182	0.055	0.018	0.327	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Concept D	0.000	0.073	0.091	0.109	0.036	0.073	0.182	0.055	0.018	0.327	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Concept E	0.127	0.000	0.091	0.109	0.036	0.073	0.182	0.055	0.018	0.327	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Concept F	0.127	0.073	0.000	0.000	0.036	0.000	0.182	0.055	0.018	0.327	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Concept G	0.127	0.073	0.091	0.109	0.036	0.073	0.182	0.055	0.000	0.327	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Concept H	0.255	0.073	0.091	0.000	0.036	0.000	0.182	0.055	0.000	0.327	0.073	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Enter name here	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Enter name here	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Examination of the decision matrix and the graphical representation reveals that Concept H, with the combined score of 1.091, is the highest scoring concept, and thus the most suitable to continue to detailed design. Concept H has the following design parameters.

- 601 Compatible 64/32-bit bus
- Split 8K 4-way Cache
- 5 Completion Units
- 2 Instructions per Cycle + 1 Retire
- 3 Execution Units with 1 Reservation Station per unit

These parameters are identical to the core decisions that were made during the design of the actual processor. While additional decisions and refinements were made during the actual design process, these decisions defined the core design of the chip and the overall direction of the project. Because we ended up with the same result as the actual design project some people may question the usefulness of the IDEA process, however it is not only the result that is important. While we may have achieved the same result, we have also generated an entire traceable flow of documentation which justifies that result. This not only increases confidence in the results but also allows our decisions to be justified if questions should arise.