The Role of Elegance in System Architecture and Design

by

Steven A. Billow

B.S., Chemical Engineering (1989) Clarkson University

M.S., Chemical Engineering (1996) University of Rochester

Submitted to the System Design and Management Program in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Engineering and Management

at the

Massachusetts Institute of Technology

February 1999

©1999 Steven A. Billow. All rights reserved.

The author hereby grants to MIT the permission to reproduce and distribute publicly paper and electronic copies of this thesis document in whole or in part.

Signature of Author

System Design and Management Program
February, 1999

Certified by

Seth Lloyd
Associate Professor, Mechanical Engineering
Thesis Supervisor

Accepted by

John R. Williams
Co-Director, System Design and Management Program
The Role of Elegance in System Architecture and Design

by

Steven A. Billow

Submitted to the System Design and Management Program on January 15, 1999 in Partial Fulfillment of the Requirements for the Degree of Master of Science in Engineering and Management

Abstract

The meaning and relevance of elegance in system architecture and design is considered. After establishing its desirability, elegance is defined by two criteria common to complex, elegant systems:
1. the system must function according to its stated purpose; and
2. the design pressures constraining the system design must be simultaneously relieved.
These criteria were developed after discussion with experts in a wide variety of disciplines and after research of seven examples given by the experts in their respective fields. Aside from the criteria above, characteristics common to many elegant solutions are identified and discussed. The relativity of elegance is discussed with respect to variations in objectives, language, timing, and culture; and means to improve one's sense of elegance are suggested. Finally, methods to create elegant solutions are considered with special attention paid to those methods recognized in the elegant solutions considered within this text.

Thesis Supervisor: Seth Lloyd
Associate Professor, Mechanical Engineering
1. INTRODUCTION

1.1 Why Care About Elegance? 9
1.2 Previous Looks at Elegance in Technology 11

2. METHODOLOGY

2.1 Overview 13
2.2 Interviewees 13

3. DEFINING ELEGANCE

3.1 Criteria for Elegance 15
3.2 Defining Function 16
3.2.1 Utility 16
3.2.2 Non-Triviality 18
3.3 Relief of Design Pressures 19
3.4 A Quantitative Description of Elegance 21

4. ELEGANT EXAMPLES

4.1 Geometric Proofs – The Pythagorean Theorem 23
4.2 Large Office Product – Conductive Carrier in Xerography 29
4.3 Small Electronic Consumer Product – The Palm Pilot 36
4.4 Fluid Mechanics – The Stream Function 39
4.5 Aerospace – The 707 43
4.6 Computer Software Product – The Apple Desktop 47
4.7 Architectural Structure – The Roman Pantheon 50

5. CHARACTERISTICS OF ELEGANT DESIGNS AND ARCHITECTURES 56

5.1 Simplicity and Efficiency 56
5.1.1 Defining Complexity 58
5.1.2 Simplicity in Computer Systems 61
5.1.3 Simplicity in Other Commercial Products 64
5.1.4 Elegance in the Scientific Enterprise 65
5.1.5 Simplicity in Organizational Design 66
5.2 Form Follows Function 67
5.3 The “Wow” Factor 69
5.4 Aesthetics 71
5.5 Clear Goals 72
5.6 Consistency 72
5.6.1 Consistency in Function and User Interface 73
5.6.2 Consistency with Environment 74
5.7 Capable of Redefining or Creating a Business 75

6. The "Relativity" of Elegance 76

6.1 Variations in Objectives 76
6.2 Variations in Language 77
6.3 Variations in Timing and Culture 78

7. Gaining a Sense of Elegance 79

8. Creating Elegant Solutions 80

8.1 Traditional Approaches to Architecting, Creativity and Innovation 81
8.1.1 Normative (Pronouncement) Methodology 81
8.1.2 Rational (Procedural) Methodology 82
8.1.3 Argumentative (Noise) Approach 83
8.1.4 Heuristics 86
8.2 Critical Features in Many Elegant Solutions 89
8.2.1 Solve the Real Problem 89
8.2.2 An Interdisciplinary, Systemic, and Holistic View 91
8.2.3 Looking to Other Disciplines, Times, Cultures, and the Arts 93
8.2.4 Tenacity 94
8.2.5 Make mistakes!! It's Human!! 95

9. Conclusions 96

10. Bibliography 99
1. Introduction

Practicing engineers, scientists, and artists are familiar with the concept of elegance in their domain. More than likely, they have all experienced the joy, enlightenment, and understanding that comes after recognizing a design or architecture as elegant. The fortunate ones have had some of their own creations endowed as being elegant. It is clearly something we all desire in our solutions and products, and is something about which we all hold opinions. But is it possible to use elegance as a qualitative or even quantitative measure to assess the worth of various system designs and architectures? Is there a definition for elegance on which practicing engineers and artisans can agree? Lastly, can the production of elegant solutions be systematized by recognizing and producing characteristics common to elegant products? These are some of the questions this thesis seeks to address.

A bounty of examples claimed by authors to be elegant can be found in the literature. In many instances, authors reference their own product or solution as being elegant. In other references, elegant solutions are reviewed to inspire similar solutions in the work of the reader. In the Harvard Business Review, John O’Connor reminds us of a few products common to everyday life so that their elegance does not go unnoticed.1 Some examples he proposes are the familiar paper clip (invented by the Norwegian Johann Valer in 1899), Adams Chiclets (from 1900), the Pilot Razor Point Pen, the Rolodex Rotary File (1950), Scotch Transparent Tape (1925), the U.S. dollar bill (standardized in 1928), the Olympus XA camera, and the paper airplane. This is a topic about which many in the scientific community have significant energy. When IEEE Spectrum, for example, published an article proposing some elegant examples in the electrical engineering discipline, it received overwhelming response from its readers, some in agreement with their propositions and some not. In this article, Donald Christiansen suggested the following as elegant examples (being careful to phrase them as candidates): ferrite core memory, the flip-flop, the negative-feedback circuit, the Zippo lighter, milk-bone dog biscuits, the Oreo cookie, the paper clip, and Post-It notes.2 Schuett gives an example of elegance to be spreadsheets such as VisiCalc™, which allow average computer users, not just programmers, to tailor applications in a simple way to their specific needs.3 In The Power of

Product Platforms, Meyer and Lehnerd are fairly liberal when describing products as elegant. Among the many they mention as elegant, some examples they offer are the Gillette Sensor razor system, HP’s ink jet printers, Braun coffee makers, Lexus automobiles, Quicken personal finance software, and the IBM Think pad. The “elegance” adjective certainly appears to have been applied to many and varied products in just these few references, and it is far from clear that each author used similar criteria in applying it.

As seen in the examples given above, elegance can mean different things to different people. Perspectives and styles come into play in determining what each of us find appealing. Russell Ackoff recognizes the importance of style in decision making. In rational decision making, given multiple means to the same end, a rational person would prefer all means equally. But that is clearly not the case. We all have our own personal styles and tastes. These same styles and tastes impact what each of us find elegant.

This is a controversial topic, one for which there is no definitive answer, perhaps because a satisfactory definition of elegance in the realm of design and system architecture is still lacking. Textbook definitions offer a meager starting point in trying to define what elegance means for a complex system. The Merriam-Webster Dictionary defines elegance as “refined gracefulness, tasteful richness (as of design).” The Jargon File on the Internet defines it as: “elegant (from mathematical usage) adj.: combining simplicity, power, and a certain ineffable grace of design.” Webster’s Third dictionary suggests elegant designs have “the qualities of scientific precision, neatness, and simplicity.”

Technical writers do a better job. David Gelernter believes beauty, a term he uses nearly interchangeably with elegance, in a mathematical proof or machine “lies in a happy marriage of simplicity and power.” He later defines power as the ability to accomplish a wide range of tasks or to get a lot done. Jon Bentley echoes these same thoughts, saying, “Simplicity is one part of elegance; power is another.” In the context of mathematics and science, John O’Connor conjectures that an elegant solution is one that is correct, efficient, and pleasing to contemplate.

---

What does all this mean to a system architect? Rechtin describes the architecting process as a “combination of arts and sciences that specifies the functions to be performed and describes the system to be built.”10 The architecture of a system determines how the product will function, what it will do, and what it will look like. If elegance is a desired trait (and I’ll argue in the next section that it is), it is prudent that the aspiring system architect consider what elegance means in a system’s design and architecture and how it might be achieved.

My intent is to survey what has been said about elegance in system design and architecture, both from the point of defining it and describing its importance. For the most part, attention will be focused on complex systems (i.e., the Oreo cookie does not qualify). Furthermore, by studying examples of elegant systems and products as a starting point, this thesis will attempt to define what elegance means in the development of complex systems and propose methods for developing elegant solutions. Readers familiar with the examples discussed in Section 4 may skip that section with little loss of continuity. The intent is to promote discussion on a topic that has been largely passed over by most traditional engineering and system design courses of study.

1.1 Why Care about Elegance?

There is much more at stake than simply pretty pictures, and good design can significantly improve the communication value of the interface, leading to increase usability.11

-- Jakob Nielsen, on the concept of elegance in computer system user interfaces.

There is perhaps no greater compliment to grant an engineer than to call his or her solution elegant. That single word speaks volumes when it comes from an informed source. But if that is all there is to elegance, a nice compliment paid to the creator, then it is of little value except, perhaps, in building self-esteem. If an elegant product possesses no competitive advantage in the marketplace, then it is not a trait worth pursuing with great earnest.

Some writers have claimed that elegance “is the ultimate guarantee of success with the public.”12 As will be shown in some of the elegant examples presented in Section 4, this is clearly not the case: elegant solutions have failed in the marketplace in the past and will surely do so in the future.

12 Gelernter p. 9
Despite this mixed track record, an elegant product does possess significant advantages over its rivals. These advantages benefit all the stakeholders. Customers realize elegance when they experience a product and its use.\textsuperscript{13} They find it is easier to make an elegant product do what they want it to do; to them, the product's operation is intuitive. Users of Intuit's Quicken\textsuperscript{TM} software, for example, are so enthralled with the product that their endorsement of the product has replaced traditional marketing. In 1991, the $33 million product was sold with just two salesmen!\textsuperscript{14} The owners benefit by higher margins on the product. Because of their typical acceptance in the marketplace, elegant products can command higher prices. In addition, elegant solutions often times are very efficient, producing more output for less input, and as a result have lower costs to produce, operate and maintain. The clever solution to reduce rear-end accidents, for example, was to spend a few dollars on an extra brake light in the rear window rather than implement one of the multitude of more complex suggestions.\textsuperscript{15} To paraphrase Bruce MacLennan, "What's good for the user is also good for the designer."\textsuperscript{16} As will be discussed in much greater detail, elegant designs are usually simpler than their kludgy counterparts, making them easier to specify, and they require fewer accommodations for complex interactions. Computer software systems are perhaps one of the best examples of this advantage. An elegant system could easily require half the code of a more poorly written system; writing, debugging, documenting, and maintaining less code is much easier and cheaper. The best programmers, says Gelernter, are obsessed with beauty.\textsuperscript{17} Lastly, manufacturers of elegant products find them much easier to make since they often have fewer components and are less sensitive to variability in the manufacturing process. The Intruder\textsuperscript{TM} Mousetrap, for instance, reduced the number of parts and the number of part types in the design. The result was a simplified, streamlined manufacturing process.\textsuperscript{18} These advantages to the stakeholders are so pronounced that truly elegant products and services often create or redefine an industry.

The advantages imparted by elegant solutions are undeniable. Unlike some authors, who contend elegance is a desirable trait simply because it is pleasing to humans due to some innate, physiological evolutionary trait common to humans,\textsuperscript{19} I believe elegance offers more than just comforting, satisfying feelings. Significant commercial advantage can be realized by its pursuit.

\textsuperscript{13} Meyer and Lehnerd, p. 82.
\textsuperscript{15} Bentley
\textsuperscript{17} Gelernter, p. 29.
\textsuperscript{18} Meyer and Lehnerd, p. 86.
\textsuperscript{19} Gelernter, p. 4.
As systems and products become more complicated, the need for elegance is felt even more earnestly. Christiansen conjectures that elegant designs may be more difficult to come by as systems grow in complexity. This may be true, but as complexity increases, the cry for elegance becomes louder and more urgent. Elegance is one of the few weapons against complexity.

We crave elegant solutions not just because they are pleasing to contemplate. We pursue elegance because it imparts great advantage to its holder. Truly elegant solutions are rare and difficult to come by, but when they do occur, they are a powerful impetus toward success.

1.2 Previous Looks at Elegance in Technology

According to legend, in Eleusis, part of ancient Greece, there lived an innkeeper by the name Procrustes. The story of Procrustes describes him as good host, though perhaps a bit overzealous. He had but one size of bed in his inn which unfortunately was rarely of the right proportions for his guests. Rather than permit his guest to sleep in an ill-fitted bed, Procrustes was determined to make bed and guest conform. So, to the rack Procrustes took the short guests for a good stretching, and the ax befell those who were regrettably too tall. Though Procrustes met his final goal, his brutish solution was far from elegant.

Elegance in design is certainly not a new topic, and the ancient Greek proverb teaches the problems of force-fitting a solution to a need. Even back in 25 B.C., much of what Vitruvius teaches in his *Ten Books on Architecture* is concerned with creating structures that are pleasing to the eye, resilient to the forces of nature, and meet the needs of the user by a harmonious combination of design, art, and material.

Modern references also abound, as already demonstrated by the numerous citings in the preceding discussion. The concept of elegance has even "gone commercial", with consultants proclaiming they can assess the "Technical Sweetness of Fit" for investors or inventors considering a new technology.

Highly constrained industries or technologies are typically familiar with elegance and its power. The aerospace industry, for example, seems to evolve and optimize an existing

---

20 Christiansen.
21 Gelernter, p 22.
22 Http://www.ifitf.org/1infotech.htm
architecture until little more can be gained from it. A new, elegant solution, such as jet engines or swept wing design, then takes the design space to a higher level. As a result, much thought has been given to elegance in these constrained disciplines. Antoine de Saint-Exupery, the French aircraft designer, for example, summed up much of the essence of elegance when he said, “A designer know he has achieved perfection not when there is nothing left to add, but when there is nothing left to take away.” Examples of elegance in this and other mature industries are not hard to find.\textsuperscript{23}

For an industry lacking constraints, elegance is many times a remote concept. In software development this is perhaps more true than anywhere. As will be discussed in section 5, the computer domain in which software developers create is nearly without bounds compared to the material world in which most design disciplines struggle. As a result, the recent literature is abundant with the pleas of software development veterans calling for elegance in computer systems.\textsuperscript{24} Some of these writings borrow from more mature disciplines for insight. Organizational design is in a similar state as software, and here the concept of elegance has recently emerged as well.\textsuperscript{25}

Still other writings, Owen Edwards in \textit{Elegant Solutions}, for example, deal more with the emotional reaction effected by elegance. In his S.M. thesis at MIT,\textsuperscript{26} Robert Leff deals largely with this emotional response and the role of industrial design and creativity in the development of elegance. This emotional reaction will be discussed briefly in Section 5.3.

Finally, an abundance of literature exists on the creative process, a critical component in developing elegant solutions. See, for example, Moshe Rubinstein's \textit{Patterns of Problem Solving}. Most texts addressing complexity and how to cope with it also touch on the creative process.\textsuperscript{27} The creative patterns common to many elegant solutions will be covered in Section 8.

\textsuperscript{23} See, for example Brennan, J; \textit{The Elegant Solution}, D. Van Nostrand Company, Princeton NJ, 1967; and Allen, F.; “The Letter that Changed the Way We Fly”, \textit{Invention and Technology}, Fall 1998, pp. 6-13.

\textsuperscript{24} See, for example, Gelernter, pp. 119-129; Jackson, “A Plea for Leaner Software”, \textit{Computer}, 28(2), February 1995, pp. 64-68; Kennedy, R.; “Elegant Kludge”, \textit{Byte}, 20(8), August 1995, pp. 54-60;


\textsuperscript{26} Leff, pp. 8-26.

2. Methodology

2.1 Overview
To gain some insight into what elegance means in the architecture and design of a system, several elegant examples from widely varying disciplines were studied. By studying widely diverging products or solutions, the elements common to elegant solutions were sought. David Billington believes there are three E's common to many technological activities: Efficiency, Economy and Elegance. Following a similar vein, it is hypothesized that many elegant solutions share attributes which bestow them this distinction.

Somewhat unfortunately, however, elegance is at least partially in the eye of the beholder. This makes the selection of examples a difficult but critical step in this process. For this reason, discipline experts were solicited for their opinions on what constitutes elegance and for examples of elegant solutions within their discipline. The list of these interviewees follows, and seven of the examples suggested by them are discussed in greater detail in Section 4.

2.2 Interviewees
Interviewees were selected because of their expertise in a discipline, not because they had considered the concept of elegance in their discipline any great depth. In general, the interviews were interactive, working sessions in which ideas were formulated throughout the discussion. As a topic, elegance is rarely discussed in strictly technical fields and so many of the thoughts were being expressed for the first time. Without exception, the interactions were enjoyable – the subject matter seemed appealing to all those who gave of their time for this investigation. Some of these were fairly short exchanges, perhaps fifteen to thirty minutes on the phone or communications via email. Others were more in depth and in person, occurring over several separate meetings possibly an hour or more in length.

Interviewees:

- Robert Armstrong – MIT, Department Head of Chemical Engineering; expert in fluid mechanics and polymers.
- Ed Crawley – MIT, Department Head of Aero & Astro; expert in aerospace system design and system architecture.
- Dan Frey – MIT, Associate Professor in Aero & Astro Department; expert in manufacturing systems and product design.
- Daniel Jackson – MIT, Associate Professor in Laboratory of Computer Science; expert in software engineering.

---

• Tom Jadwin – Eastman Kodak Company Research Scientist, EPEC/Toner Development; expert in electrophotography.

• Seth Lloyd – MIT, Associate Professor in Mechanical Engineering Department; expert in measures of complexity, information theory, and quantum mechanics.

• John May – Eastman Kodak Company Research Scientist (retired), Color Sensitizing Operations; expert in electrophotography.

• Alex Slocum – MIT, Professor in Mechanical Engineering, expert in mechanical design.

• Steve Weinstein – Eastman Kodak Company Research Scientist, Coating Technologies Division; expert in fluid mechanics.

• Dan Whitney – MIT, Senior Research Scientist in Ctr Tech, Pol & Ind; expert in product development, design, and assembly.

• Mark Zaretsky – Eastman Kodak Company Research Scientist, Support Coatings; expert in electrostatics.

3. Defining Elegance

Defining elegance is perhaps a dangerous job – a kludgy, awkward or cumbersome definition would certainly not be fitting. Gell-Mann teaches that the length of the most concise description of an object can determine its effective complexity. Elegance would lose much of its potency if it were a complicated, unwieldy subject, so its description must be kept succinct yet encompass its full meaning.

Gell-Mann describes these concise descriptions of objects as schema, and they are developed by condensing observations of the regularities in the world around us. In developing a schema for the concept of elegance, the existing literature, the examples of elegance (Section 4), and the numerous opinions of experts and colleagues were distilled and condensed to define what elegance means in a complex system. This definition was formulated after much of the interviews and research on the examples were completed so that an anticipated definition would not bias the selection of the examples nor the opinions of the experts.

The preceding discussions introduced much of what exists in the current literature on the subject of elegance. Those authors that seek to define elegance usually speak of simplicity and power. Gelernter describes elegance as “beauty that springs from union of power and simplicity,” a thought paraphrased by other authors as well. Many discussions relate elegance closely to efficiency. Meyer and Lehnert in The Power of Product Platforms propose a working definition for an elegant system as “one in which all subsystems, taken as a whole,

---

30 Gell-Mann, p 17
32 See, for example Anthes and Bentley.
create the greatest output performance for the least inputs. Similarly, Brooks believes that the “ratio of function to conceptual complexity is the ultimate test of system design.”

All of these thoughts touch the essence of elegance. Simplicity is a dominating characteristic of many elegant solutions and is perhaps one reason elegance is so appealing. Simplicity alone is not sufficient – the product must also perform, and this is why power is also part of the equation. An elegant product, proof, or solution must sacrifice nothing by way of effectiveness for the sake of simplicity alone. Indeed, in some instances simplicity is not even the most important consideration in some systems, and some complexity would gladly be accepted for relief of another design pressure.

The definition proposed below encompasses this idea of power with simplicity but also seeks to broaden the thought to include other design constraints. The definition follows closely that proposed by Ed Crawley in his discussions of System Architecture.

3.1 Criteria for Elegance
Besides being too complicated, a definition of elegance faces two other potential pitfalls: being so specific that it is not applicable to many types of systems, or being so broad that, although applicable, it says little to assist the system architect. I have perhaps failed by being too general here but hope that the discussion on characteristics common to many elegant solutions (Section 5) helps those seeking more substance find it.

To be considered elegant, a system must meet two criteria:
1. the system must function according to its stated purpose; and
2. the design pressures constraining the system design must be simultaneously relieved.

The first criterion might be considered tables stakes. In an engineering sense, a product or system which fails to meet its goals should not be considered elegant. This could be considered a bit unfair – a system might be capable of meeting its goals, but, because of outside forces or externalities (e.g., regulations), be prohibited from doing so. Likewise, it might be argued from an aesthetic standpoint that a solution need not function to be considered elegant. A mathematical proof, for example, may fail to reach the desired conclusion but, because of its beauty, simplicity and ingenuity, may inspire others and in this way be considered elegant. For the purposes of creating an engineering system definition, however, it is sensible to restrict the

33 Meyer and Lehnerd, p. 95.
34 Brooks F.; The Mythical Man-Month. Addison-Wesley, New York, Ed. 2, 1995, p. 43
35 Crawley, E.; System Architecture class notes, Fall 1997.
"elegance" tribute to solutions which meet the desired, functional objectives set for the architect and designer.

The second criterion speaks about the tradeoffs necessary in the conventional design methodology. It is standard practice in design to make tradeoffs in functional performance, one aspect of performance against another when all functional requirements can not be met simultaneously. Cost versus speed, weight versus versatility, robustness versus manufacturing costs, etc. are some of the many tradeoffs dealt with in complex systems. Finding the optimum within a set of constraints is true engineering and is standard fare in engineering curricula. In designing a cryptography solution, for example, Anderson suggests seeking a "robust solution" which may be "less than optimal but provides some benefit."36 The engineering literature is overflowing with examples of these difficult tradeoffs. A truly elegant design, however, does not force a trade-off; it meets the needs of the user without substantial sacrifice while staying within the constraints of the environment. As will be seen in Section 5.1, this criterion also helps explain why many elegant solutions are also quite simple.

These two criteria will be discussed in more detail in the following section.

3.2 Defining Function
In his hints for software design, Lampson cautions that "neither abstraction nor simplicity is a substitute for getting it right."37 We hold this true for elegant solutions as well. They must accomplish the goals set before them. Being a critical part of the definition of elegance, some time is worth spending to briefly discuss what is meant by function. For our case here, two elements will comprise function: utility and non-triviality.

3.2.1 Utility
"Function is typically some key aspect of performance that can be measured."38 This definition of function becomes more complicated when there are several aspects of performance are important and must be considered simultaneously. Techniques are widely available, however, to deal with this complexity. The Pugh concept selection matrix, for example, is useful for identifying strengths and weaknesses of existing and proposed systems and for suggesting alternative hybrid systems which are superior to any of those alternatives.39

38 Meyer and Lehnard, p. 99.
For systems that exist only in the minds and the sketches of their designers, assessing the performance becomes a job of estimation. Often, the likelihood or probability of the system meeting the functional requirements is used in lieu of actual performance measures. Nam-Suh, in his theory of Axiomatic Design, suggests “information” (a measure of the merit of a design and a quantity to be minimized) be defined as \( \log_2(1/p_i) \) where \( p_i \) is the probability of success.\(^{40}\) In the general case of \( n \) functional requirements, each with \( p_i \) probability of success, information, \( I \), becomes:

\[
I = \sum_{i=1}^{n} \log_2 \left( \frac{1}{p_i} \right)
\]

For products already in the marketplace, it is tempting to use some measure of the product’s success to imply the product’s level of function. The biological concept of “fitness” appears at first to be a particularly attractive abstraction. Gell-Mann describes it succinctly (my italics):

The idea underlying the biological concept of fitness is that propagation of genes from one generation to the next depends on survival of the organism until it reaches the stage of reproduction, followed by the generation of a reasonable number of offspring that in turn survive to reproduce. Different rates of survival and reproduction can often be roughly described in terms of a fitness quantity, defined so that there is a general tendency for organisms with higher fitness to propagate their genes more successfully than those with lower fitness.\(^{41}\)

Using this abstraction, one might conjecture that a product which is more fit for the task (i.e., that performs at a higher level), would be the most likely to survive and “procreate” via new releases or versions. Just as market success is a poor measure of elegance, however, it is just as bad at suggesting utility. Products that work often fail.

Ultimately, it is the customer who defines how functional a product is. Rechtin sums this up nicely in one of his heuristics: “Success is defined by the beholder, not by the architect.”\(^{42}\) In determining the success or failure of a programming language, MacLennan points out that social factors are many times more important than scientific factors. “Economy is a social issue.”\(^{43}\) These observations are true because the interface between the customer and the product is the single most important interface a system architect must create.

\(^{40}\) Suh, Nam Pyo; Axiomatic Design: Advances and Applications, Draft of Chapters 1 and 4, 1997, p. 41.
\(^{41}\) Gell-Mann, p. 248.
\(^{42}\) Rechtin, p. 34.
\(^{43}\) MacLennan, p. 34.
This is abundantly clear when considering user interfaces with computer systems. "The human-computer interface strongly determines the grade of success a user reaches when working with whatever powerful software and hardware."44 Simply changing a user interface by making function more apparent to the user can increase utility without any modifications to the actual function of the product.

A good counter example of how a user-interface can increase the effective-function of a software package might serve to illustrate this point. Excel™ by Microsoft is no doubt a powerful tool being capable of many functions. For most tasks, it is relatively easy to use and straightforward. Needing to find the slope of some data, I naturally turned to Excel but was disappointed, though not surprised, by the procedure required for this simple task. A user reasonably well-versed in spreadsheet usage would require something similar to the following steps to find the slope of a line in Excel:45

- Find the name of the command in ‘Help’: LINEST, perhaps not intuitive but not too difficult to remember.
- Look up the syntax: LINEST(known_y’s, known_x’s, const, stats).
- Discover it must be entered as a formula array (after getting an error message – this minor detail is not mentioned in the Help file).
- Learn how to enter formula arrays (which entails entering the formula with an non-intuitive ctrl+shift+enter).

This is an example of a product with immense function, much of which is difficult for the average user to access.

In summary, to assess a product’s utility, how it functions in the hands of a consumer must be considered. If a product is still on the drawing board, the human-machine interface must be included when determining the likelihood of success. For existing products, the limits of function are irrelevant – what matters is how the user can extract function from the product. Lastly, and perhaps obviously, market success is not a measure of utility.

3.2.2 Non-Triviality

Gelernter believes for a machine to be beautiful, it must "accomplish something significant. If I make a casual toss, you can not make a beautiful catch."46 An good solution might solve a simple problem well, but if the problem is relatively trivial, one would be hard-pressed to call the solution elegant. The notion of depth is useful in this context.47 Crudely stated, it considers

45 Microsoft Excel™ for Windows 95™, Version 7.0a.
46 Gelernter, p. 4.
47 Gell-Mann, p. 100-102.
the number of alternative solutions that are capable of attaining the objectives set by the problem. More precisely, depth is a measure of the probability of devising a solution that would function. If a problem is so straightforward that many adequate solutions exist to meet the needs set for the product and they do so within the constraints set by the environment (e.g., required cost, workable pressures and temperatures, adequate speed, etc.), then a solution can not be said to be elegant. By inclusion of this criterion, I reject objects such as the Oreo cookie, the Zippo Lighter, and the Sunbeam Electric Iron as being elegant complex systems,\textsuperscript{48} not because they are not good solutions and good products, but because there are many alternatives which adequately meet the functional requirements of the system. The discussion within this thesis will be restricted to examples for which alternatives meeting the functional requirements are few or non-existent.

\textbf{3.3 Relief of Design Pressures}

Ackoff suggests there are three ways of reacting positively to a problem:\textsuperscript{49}

- To Resolve: reach an outcome that is merely satisfactory.
- To Solve: optimize the outcome of a problem.
- To Dissolve: remove the problem.

The first method fails to truly address the real problem but merely provides a stop-gap to prevent further adverse consequences. The second method, "solving a problem" is what is entailed during a typical design effort. Constraints are recognized, and the performance of the product is optimized within those constraints. In the third method, "dissolving" a problem, the constraints that once limited the solution are no longer an issue. There are two ways this can be accomplished. 1) Remove the constraints. If the constraints are man-made, this might entail lobbying a legislature or striking corporate partnerships. If the constraints are rooted in physics, the engineer has little hope of removing them. 2) Architect an elegant solution. As described in Section 3.1, an elegant solution simultaneously relieves the design pressures on a system so that all the objectives of the system can be met without compromise. Some authors fail to recognize this power of elegance, stating that elegance merely balances constraints.\textsuperscript{50} But it is this criterion that distances truly elegant solutions from more typical ones.

Design pressures are indeed the source of motivation to create elegant solutions. If necessity is the mother of invention, design pressures and constraints are the mothers of elegance. The converse is also true: where no constraints exist, elegance is a much lower priority, gets paid little attention and is unlikely to describe the ultimate solution. Brooks makes a similar

\textsuperscript{48} These objects were described as elegant by some authors, Christiansen, Meyer, and Lehnerd.

\textsuperscript{49} Ackoff, p. 170-172.
observation regarding structural architecture and the Stretch computer: “The worst buildings are those whose budget was too great for the purpose to be served. I am sure the Stretch computer would have had a better architecture had it been more tightly constrained.”

A similar effect is observed in constrained styles of writing such as poetry: “In poetry, you are often tightly constrained in format. Your objective is to express meaning and beauty within a strict set of functional requirements.”

Nearly every elegant example can be traced back to design pressures. The geodesic dome, for instance, distributes the structural forces so well that the structure can be made from a minimum of materials with minimal weight. Another excellent example is the solution to pumping the pressurized water in Submarine Thermonuclear Reactors. The water is kept at very high pressure to keep from boiling but naturally cannot be allowed to leak because of the possible radiation exposure. The original attempt used high performance, tightly specified fittings to prevent the leaks, but no fitting remained water-tight for a satisfactory duration. Ultimately, the truly elegant solution was to embed the pumps within the pipes and provide energy to them electromagnetically through the pipe walls, eliminating the fittings all together.

The Chicago-style font developed for the original Macintosh systems is also the result of design pressures. The original Macintosh systems used low-resolution screens, capable of only 72 dot-per-inch. The creators of the Macintosh wanted the screen to be legible even when the screen was “inactive” and went to a 50% gray. The thick vertical elements of the Chicago-style font makes text legible at low resolution and at low screen intensities. The font has become such a trademark of Mac systems that the font remains despite the relaxation of the original design constraints.

Cost is naturally a common constraint and nearly every design pressure is traded against the cost pressure. Many times, cost acts as a common denominator so that two different design pressures can be resolved. In a spacecraft, for example, weight can be traded for the higher costs of more exotic alloys. In a bicycling, the performance boost from a perfectly fitting, tailor-made frame is traded against the negative impact on the total budget for the bike. Fortunately, elegant solutions can also be effective at relieving the pressure of cost constraints as well, making them desirable from strictly financial perspectives as well as the engineering perspective.

---

50 Leff, p. 8.
51 Brooks, p. 47.
52 Anthes
53 Leff, p. 17.
54 Brennan, p. 37.
55 Mullet and Sano, p. 28.
Complexity is perhaps the most prevalent design pressure and, as a result, elegant solutions are, almost without exception, remarkable for their simplicity. Complexity drives costs throughout the entire product life-cycle: more complex designs require larger teams to develop, more involved manufacturing processes, more care by the user during use, and greater cost to maintain. Beyond costs, humans are capable of dealing with only finite complexity due to our limited cognitive capabilities. Abstractions and hierarchy help deal with this complexity but have their limits as well. This design pressure is relieved in all elegant systems, and simplicity and its measures will be dealt with in considerably more detail in section 5.1.

Design pressures result when the system is required to function at level not permitted by the current design in the operating environment. Design pressures can be considered to result from the failure of the system to meet its “functional requirements” as defined by axiomatic design. Using these semantics, the two part definition described above may be condensed into one simple statement: an elegant solution is one which meets all the functional requirements defined for the complex system. Care must be taken, however, to ensure all the functional requirements, including the inherent need for simplicity, are stated properly. The preceding discussion concerning utility, non-triviality, and design pressures are equally relevant when this definition is used.

3.4 A Quantitative Description of Elegance

An original tenet of this thesis was to explore the development of a quantitative definition and measure of elegance in complex systems. This proved to be a difficult task; although measures of complexity and function exist (see sections 3.2.1 and 5.1.1), the tradeoff between these two and other design pressures is a difficult entity to quantitatively evaluate. As stated above, cost is ultimately a useful “common denominator” to evaluate a system and some authors\textsuperscript{56} choose to use it as the metric to evaluate tradeoffs. Although this makes sense from a business standpoint and every product’s success is ultimately measured by its :return, measuring elegance by costs fails to capture the full essence of elegance. An elegant design will most likely be less expensive to specify (i.e., design) and manufacture and usually will result in lower costs for the firm. Costs fail to capture how well the product meets the needs of the user, however. Even if two products are capable of the same function, they may provide different benefits in the hands of a typical user. Furthermore, the commercial success of a product is a poor indicator of elegance, as will be demonstrated in the elegant examples of section 4.

\textsuperscript{56} Meyer and Lehnerd, for example, pp. 100-105.
Thermodynamic measures, such as the total free energy required for the user to extract function from the product, are more attractive alternatives. It is feasible to quantify the free energy inherent in the design through measures of complexity\textsuperscript{57,58} and the energy required to make and assemble the product is clearly quantifiable. However, it becomes more difficult to quantify the energy required of the user in extracting the function. It is possible to quantify the energy required in executing the steps most users would require (e.g., to save a document from a word processing program, for example, may require two or three menus, browsing subdirectories, and typing the name of the file), but this is not adequate. These measures fail to capture the feeling of comfort, confidence, and ease the user experiences as a result of using the product. Measures of aesthetics (see section 5.4) may provide some clues, but their development is too immature to be of great use in defining elegance. Therefore, this thesis will be confined to the qualitative description of elegance.

\textsuperscript{58} Lloyd, Seth and Pagels, Heinz; “Complexity as Thermodynamic Depth”, \textit{Annals of Physics}, Vol. 88, 1988, pp. 186-213.
4. Elegant Examples

The following examples are discussed briefly to introduce some elegant examples to support subsequent discussion on what elegance means in system design and architecture. They are purposely from a wide variety of disciplines: geometry, electrophotography, consumer electronics, fluid mechanics, aerospace, computer operating systems, and architecture. Only brief summaries or overviews are provided in order to illuminate characteristics that make the solution elegant. References are provided for the reader desiring greater detail.

4.1 Geometric Proofs – The Pythagorean Theorem

"The aesthetics of natural science and mathematics is at one with the aesthetics of music and painting – both inhere in the discovery of a partially concealed pattern."

- Herbert Simon, expert and pioneer of artificial intelligence

Perhaps nowhere else is elegance more readily apparent than in a concise geometric proof. The usage of "elegance" in the realm of design most likely stems from the mathematical treatment. The Jargon File on the Internet defines "elegant" as: " (from mathematical usage) adj. combining simplicity, power, and a certain ineffable grace of design." It is fitting then that the first example considered is a proof in geometry. To keep the treatment here concise and yet still valuable, a geometric theorem that is simple with many alternative proofs serves as a good example; the Pythagorean theorem meets these criteria nicely.

In light of the definition of elegance proposed above, the desired function and the design pressures for the geometric proof must first be recognized. Simon views a mathematical derivation as a change in presentation which is to make evident what was previously true but obscure. The function of a geometric proof is to make a proposition clear and unambiguous for the reader. As far as the design criteria, a geometric proof must be concise to maintain the train-of-thought of the reader. Brennan further adds that an elegant solution in mathematics exhibits precision, neatness and simplicity. In addition, it should be as general as possible (e.g., work for any right triangle, not just 45-45-90 or 30-60-90 triangles). A proof of the Pythagorean Theorem is considered elegant, therefore, if it clearly and unambiguously teaches the relationship amongst the lengths of the sides of a right triangle and does so concisely, precisely, neatly, generally and simply.

60 Simon, p. 153
61 Brennan, p. xi.
Although the presentation medium for this thesis is incapable of demonstrating such an example, a movie is an excellent way to demonstrate and prove Pythagorean theorem. Interactive Java applets on the world-wide-web, for example, show several proofs that clearly communicate the both the theorem and the thought process behind the proof. Attention will be confined to static proofs for the remainder of this section.

**Example 1**
One of the simplest and most straightforward proofs of the Pythagorean Theorem requires just one drawing and a few lines of algebra.

![Diagram of Pythagorean Theorem](image)

The abc triangle is replicated and rotated three times to create an inner and an outer square. The area of the outer square is the area of the inner square plus four times the area of the triangle. The triangles have been so arranged that the length of one side of the outer square is \( a + b \). From this we have the equality:

\[
(a + b)^2 = 4ab/2 + c^2
\]

A single line of algebra yields the theorem:

\[
a^2 + 2ab + b^2 = 2ab + c^2
\]

\[
a^2 + b^2 = c^2
\]

---

62 See, for example, the award-winning Java applet written by Jim Morey at http://SunSITE.UBC.CA/LivingMathematics/V001N01/UBCExamples/Pythagoras/pythagoras.html

Example 2
Another simple example uses vector notation and the linear algebra rules of the dot-product to prove the Pythagorean Theorem:

\[ \|u + v\|^2 = (u + v) \cdot (u + v) \]

by the distributive property of the dot product,

\[ \|u + v\|^2 = (u) \cdot (u) + 2(u) \cdot (v) + (v) \cdot (v) \]

since \( u \) and \( v \) are perpendicular, their dot product is zero and one is left with the Pythagorean Theorem:

\[ \|u + v\|^2 = \|u\|^2 + \|v\|^2 \]

This proof is particularly elegant. It is extremely concise and yet powerful: it works in \( \mathbb{R}^n \), requiring only that \( u \) and \( v \) be orthogonal to each other.

Example 3
It is instructive to consider inelegant proofs of the Pythagorean theorem as well, to serve as a comparison to the examples provided above. As shown in the first example, the use of a graphical representation can significantly assist the reader in grasping the thought-process of the proof. Simply adding visual aids is not sufficient, though. A geometric proof may be graphical and still can be convoluted and non-intuitive. An example is a proof of the Pythagorean theorem attributed to Leonardo da Vinci. The proof is shown below with intermediate steps included to make it easier for the reader to follow. An individual better versed in the precepts of geometry would likely find some of these steps superfluous – a concept which will be covered in more depth in the section considering the relativity of elegance as a function of the observer’s “language” or experience level.

1. We start from Euclid’s proposition 21 of Book III of The Elements, as da Vinci did, that “In a circle the angles in the same segment equal one another”. It follows that the hypotenuse of a right triangle circumscribed by a circle must form a diameter of the circle since the line segments forming the right angle have half the circumference as a base (Proposition 20 of Book III). The converse is also true: if three points on the perimeter of a circle are joined to form a triangle and one of those line segments forms a diameter, then the triangle is a right triangle. Triangle ABC exemplifies this below:

---

65 This proof is based upon the representation provided by H. Eves, *Great Moments in Mathematics before 1650*, MAA, 1983. A number of intermediate steps have been added to assist the reader.
66 For an excellent Java demonstration of this proposition, see:
2. It is also easily shown for the same reason that the center of a square, point O, with the hypotenuse as one side also resides on the perimeter of the circle:

3. The triangle AOC is of course also a right triangle since AC forms its hypotenuse. Also, it is clear that $\angle s'$ and $\angle t'$ must be $45^\circ$ since point O is the center of the square ACIJ and that the AO and OC arcs form exactly one-quarter of the perimeter of the circle.

4. $\angle s$ and $\angle t$ must also be equal to $45^\circ$ since they each have bases forming one-quarter the circumference of the circle (arcs AO and OC). This is also a result from Euclid's proposition 21 of Book III.

5. Now four quadrilaterals are created, encompassing squares on each side of the original triangle, $\square$ADFB, $\square$BFGC, and $\square$ACIJ plus two copies, $\triangle$IJH and $\triangle$EFB, of the original triangle, $\triangle$ABC:
6. Four quadrilaterals have been formed: ABHI, BHJC, ADGC, and DEFG. Furthermore since ∠ABO and ∠CBO have been shown to be equal to 45°, it is also apparent that all four quadrilaterals are equal.

7. Since the quadrilaterals are equal, so too are their areas. Summing the area (denoted as \(A\{\})\) of the bottom two quadrilaterals must equal the sum of the area of the top two quadrilaterals:

\[ A\{ABHI\} + A\{BHJC\} = A\{ADGC\} + A\{DEFG\} \]

8. Each side of this equation contains the areas of exactly two triangles which are identical to the original triangle ABC. Their areas may be removed without changing the equality, resulting, once again, in the Pythagorean Theorem:

\[ AC^2 = AB^2 + BC^2 \]

Many more examples, both elegant and inelegant are available but these examples provide some insight on what elegance means in a geometric proof. The first two examples were intuitive and their results obvious after they were presented. For someone unfamiliar with
proofs of the Pythagorean Theorem, a feeling of excitement and satisfaction would likely result after being shown the first example. The last proof, however, leaves a reader with little more intuition about the Pythagorean theorem than before being exposed to it. The first two proofs are direct and accomplish their function with concise and unambiguous arguments and without introducing additional postulates en route to the solution. In this sense they are also efficient. The generality of the second proof makes it quite powerful as well. In general, these concepts are consistent with what makes a system design or architecture elegant.

4.2 Large Office Product – Conductive Carrier in Xerography

"Conductive magnetic brush development has produced significantly improved copy quality by ... increasing customer-perceived "blackness" and by making the optical reflection density of lines and solids equal."

- L. B. Schein, veteran of the electro-photography business; from Xerox and IBM

Chester Carlson saw the need for a cheap, easy to use, safe, and reliable means to make copies in the offices of corporate America. Having a BS in physics and working in the patent offices of Bell Labs and the P. R. Mallory Company gave him the background to develop such a device and change the way business operates. It took him eighteen years to see his vision materialize into a reliable, affordable machine. Many technical hurdles had to be overcome during those development years and corporate backing was difficult to come by until Joe Wilson, president of the Haloid Corporation, shared Carlson’s vision and essentially bet the company on the success of electrophotography.⁶⁸

The elegant solution considered here is conductive magnetic brush development, a technology that revolutionized electrophotographic development and raised the quality of electrophotographic images to the level we expect today. A brief summary of electrophotography is provided first so that the reader can appreciate the elegance of conductive carriers.

The critical contribution and main invention by Carlson was the bringing together of two important ideas:⁶⁹ 1) use of a photoconductor (a material that is conductive when exposed to light and insulating otherwise) to form an electrostatic latent image and 2) creation of the real image by dusting the latent image with electrostatically charged powders (e.g., toner). The rudiments behind the electrophotographic process are essentially the same today as they were in Carlson’s first machine. The process consists of six basic steps, shown and described below⁷⁰:

---

⁶⁸ Brennan, p. 64.
⁷⁰ Adapted from Schein, p. 2.
Step 1: A uniform charge is sprayed onto the photoconductor by a corona discharge. The photoconductor, which is only conductive when exposed to light, maintains a majority of the charge sprayed on its surface as this step occurs in the dark.

Step 2: Areas of the photoconductor are selectively discharged after becoming conductive when light from the image (or from a laser in a printer) exposes the photoconductor. The charged and non-charged pattern is called the latent image.

Step 3: Toner particles, polymer beads embedded with carbon black, that have been charged with opposite polarity as the photoconductor are brought into contact with the photoconductor. The toner will adhere to the parts of the photoconductor that retains its original charge but not to those portions that were discharged by the light source.

Step 4: The toner that has adhered to the photoconductor is transferred to a paper copy. This process is greatly facilitated by charging the paper, again by a corona discharge, to be the opposite sign as the toner particles.
5. Fuse

Step 5: The toner that has been transferred to the paper is fixed with heat.

6. Clean

Step 6: Residual charge on the photoconductor is discharged by exposure to light then the photoconductor is cleaned of any excess toner by brush, forced air, or scraper blades.

Step 3, the development stage, determines the maximum quality of the final image that is possible. The simple process depicted above is called "cascade development": the toner is simply brought into contact with the photoconductor by gravity and the electric field created by the photoconductor is the only driving force for adherence. With no other electrode, one can see that the electric field will primarily be confined between the top surface of the photoconductor and its ground plate. Only near the edges where fringe fields exist will the electric field penetrate the surface of the photoconductor and attract the toner. This worked satisfactorily for text but created terrible results for solid area development. A solid black square would be reproduced as shown below:71

71 Schein, p. 51.
Schein offers this commentary regarding cascade development:72

Perhaps the most crucial technologies which are not well understood are those used in the development step, because this step most directly determines the quality of the images. It is this step that the "blackness" of the lines and solid areas, the cleanliness of the non-imaged areas, the uniformity of solid areas, and the ratio of the blackness of lines to solid areas are determined. Those who used Xerox copiers during the 1960s will remember that they would only reproduce the edges of solid areas, a copy quality defect attributable to characteristics of the open cascade development system.

Two fixes were implemented to reduce the problem with solid area development. First, an additional electrode was placed above the photoconductor to create an electric field over the entire solid area, not just the edges. This electrode was at a voltage of the same polarity as the photoconductor but at a lower amplitude; this also helped clean-up the background areas where development was not desired. The effect of this electrode is shown schematically below:73

---

72 Schein, p. VII.
73 Schein, p. 34.
The second fix was to incorporate carrier particles with the toner. These are about two orders of magnitude larger than the toner particles and have surface chemistries designed to tribocharge with the toner particles. The carrier particles would bring the toner into contact with the photoconductor where it would transfer from the carrier to the photoconductor surface. Addition of the carrier particles provided two distinct advantages:

- Because the carrier particles were easier to handle, a significantly larger amount of toner could be presented to the photoconductor during the development stage.
- The carrier particles charged the toner particles much more uniformly than previous methods (e.g., rubbing the toner particles against a brush).

To facilitate transport of the carrier particles and create additional agitation, these carrier particles were also made magnetic and the carrier/toner mixture was “stirred” during development to encourage charging of the carrier and toner particles.

The insulating magnetic carriers with a biased electrode produced solid area development much better than cascade development. It did suffer one serious drawback. When the carrier particles dispensed their toner onto the photoconductor, they retained their opposite charge and migrated along the electric field toward the biased electrode. Here they began to congregate, forming chains due to their magnetic dipoles, and weaken the electric field granted by the biased electrode because they are of opposite charge of the toner particles. This is shown schematically (and greatly simplified) in the drawing below:

---

74 M. Zaretsky, personal communication, December 1, 1998.
Ultimately, the carrier particles would become prevalent enough that the electric field would no longer direct toner toward the photoconductor and further solid area development would be inhibited.

This was solved by George Kasper and John May at Eastman Kodak\textsuperscript{75} by modifying the \textit{shape} of the carrier particles to make them more conductive. Because the carrier particles form chains due to their magnetic dipoles, the only electrical path to ground (via the biased electrode) for most carrier particles was through other carrier particles. Carrier particles of the time were very spherical so the contact area between particles was minimal – picture two billiard balls “kissing” one another. The conductive carrier by Kasper and May has an irregular, “sponge-like” shape which increases the contact between adjoining carrier particles. By making this change, the carrier particles could slowly bleed their charge into the bias electrode (where it would eventually bleed to ground), thus maintaining the desired electric field and depositing more toner particles onto the photoconductor. This small change had a profound effect. It allowed for \textit{dramatically} improved solid area development and was released in the Kodak Ektaprint 150 copier in 1975. Ironically, both Xerox and IBM, the two major power houses in photocopies at the time, decided that the technology of electrophotography had matured, and both companies largely discontinued their research efforts in the area in 1975.\textsuperscript{76} The Ektaprint 150 also included an organic photoconductor belt and a recirculating document feeder. “It was immediately recognized by the public and electrophotographers as a major advance in copy quality.”\textsuperscript{77}

This solution clearly meets the proposed definition of elegance. The function of the electrophotographic process (create faithful reproductions cheaply and fast) was indisputably

\textsuperscript{76} Schein, p. 8.
\textsuperscript{77} Schein, p. 9.
enhanced while the disadvantages of incorporating a carrier particle were overcome by making the carrier slightly conductive. It introduced no complexity into the product over the insulating carrier, and as a result, reliability did not suffer. It is ironic that such a small change could fix the single largest problem with electrophotography at the time. Like many elegant solutions, its obviousness is stark and its simplicity refreshing. The fact that the two major companies in electrophotography overlooked such a solution speaks to its ingenuity.

This development was such a leap forward that those familiar with the situation in 1976 believe Kodak could have dominated the copier market with this technology, but it did not seize the advantage. Kodak failed to create the manufacturing capacity to sufficiently dominate the market place, and replacement technologies were eventually introduced by other manufacturers. There are several possible reasons for this lack of action by Kodak. First of all, market projections were conservative. Secondly, Kodak still viewed itself as a traditional silver halide imaging company and did not see electrophotography as a core competency of the company. Furthermore, this new imaging technology was seen as a possible competitor to the lucrative silver-halide business. As will be seen again with the Apple Desktop, elegance does not guarantee market success.
4.3 Small Electronic Consumer Product – The Palm Pilot

"Amazing that a little computer can cause such an emotional response....How could I ever think of leaving my Palm? Seems ludicrous now."78

-Scott Andress, Palm Pilot owner

In an area such as consumer electronics that has seen a phenomenal amount of innovation in the recent past, a product must truly be extraordinary to create the kind of stir the Palm Pilot has. Within eighteen months of its release, Palm Computing Inc. had shipped more than 1 million Palm Pilots.79 It was "one of the fastest selling consumer-electronics products in history. The Pilot sold faster than cell-phones, pagers, even color TV. It was the fastest-selling computer product ever."80 "Today the Palm Pilot is more than a hot product. It's the center of an industry. More than 5,000 programmers are working on new software applications. Some 200 developers are designing hardware add-ons. And the Palm PDA is more than a business phenomenon. It's a cultural phenomenon, a hip techno-artifact featured in TV shows ("Murphy Brown") and Hollywood movies (Wag the Dog)."81

The functional requirement of a personal digital assistant (PDA), is "to allow users acceptably fast access to the information they request, everywhere they are, at any time they want, and in the quality they need".82 The first PDA, called the Newton Message Pad, was introduced by Apple and fell distantly short of this functional goal. It was a notorious failure for several reasons. Most noticeably, its handwriting software performed inadequately, creating frustrated and irritated customers in its wake. Some argue that the architecture of the database was the source of its demise. Ironically, Apple's Newton Message pad was announced by Apple to be a "knowledge navigator" before its introduction into the market. Following its introduction

---

78 Andress, Scott, editor of Palm Planet; http://palmtopplanet.interspeed.net/palm
80 Dillon, p. 98.
81 Dillon, p. 110.
experts complained that “the Newton has not reached this claim by any means”. The main reason, they argue, is that Apple did not pay attention to where the information that people need in everyday life should be stored or how to best gain access to it. The first PDA by Palm Computing, called the Zoomer, also shared some of these shortcomings; it was too big, too slow, and too expensive.

Jeff Hawkins, the visionary behind the Palm Pilot, toiled at its creation for fifteen years. With a background in electrical engineering and cognitive science, Hawkins based the Palm Pilot on two simple principles that turned out to be critical for its success. These principles are the essence behind its elegance. They redefined the problem and the product and ultimately the marketplace.

The first principle entailed reinventing the hand recognition software. Most attempts at hand recognition software were focused on creating a system powerful enough to decipher actual handwriting from many people using their own styles. This complicated and slowed the operation of these systems. Hawkins recognized this bottleneck and decided the appropriate place for the complexity is with the human. Hawkins on this subject: “People are smarter than appliances. They can learn. People like learning. People can learn to work with tools. Computers are tools. People like to learn how to use things that work.” This was the birth of Graffiti, Palm Pilot's handwriting-recognition software, it simplifies English letters such that the computer can readily recognize them. The trick to Graffiti's unique alphabet is that every letter is formed by a single stroke so the stylus never has to leave the touch pad to form the letter. The letter “A”, for instance, looks like a “Λ”; “β” looks like the Greek letter “β”. Hawkins was also correct that people could easily learn the alphabet. “Patience and practice, although not too much of either, are required to achieve pleasingly quick, accurate Graffiti input,” states one reviewer. This has been a defining feature for the Palm Pilot and a major reason behind its success.

The second premise behind the Palm Pilot was to simplify, simplify, simplify. Originally, PDAs were designed to rival desktops in function; the PDA designers believed consumers would buy PDAs as a replacement for desktops. What Hawkins and his team realized was that the average customer of the PDA planned to use the device to supplement their desktops; over 90% of Zoomer customers already owned a PDA. Only mandatory features were permitted. Whereas

83 Schuett et al, p. 355.
84 Dillon, p. 104.
85 Dillon, p. 106.
87 Dillon, p. 102.
the Zoomer was bulging with add-ons (printer and fax drivers, for example), the Palm Pilot offered only four: a calendar, an address book, a To-Do list generator, and a memo-writing feature.\textsuperscript{88} This also permitted a smaller box, roughly 0.7x3x5 inches, slightly bigger than a deck of cards. Hawkins and his team were obsessed at creating a product that would fit into a shirt pocket.

The results of these simplifications were astounding. Simplifying the product meant its development could be streamlined. Apple spent $500 million to develop the Newton; Palm Computing invested only $3 million to introduce the Palm Pilot.\textsuperscript{89} The product worked better too. Clifford Stoll, astronomer, author, and computer aficionado gives the Palm Pilot an example of elegance: “It answers a single need – for a pocket-size address book -- elegantly and without warts.”\textsuperscript{90} To top it off, it runs on only two AAA batteries.

Hawkins recognized 1) what people really wanted PDAs to do and 2) how to do it most effectively. Both of the Palm Pilot design principles did one thing: redefine what the true problem is and how to best solve that problem. This is a key lesson in developing elegant solutions. As will be discussed in greater detail later, many elegant solutions find their start when the problem to be solved is correctly stated.

\textsuperscript{88} Dillon, p. 106.
\textsuperscript{89} Dillon, p. 98.
\textsuperscript{90} Anthes.
4.4 Fluid Mechanics – The Stream Function

“Simplifications like this are like mathematical orgasm.”
- Steven Weinstein, fluid mechanics expert
on the concept of the stream function.

Many solutions of problems in fluid mechanics begin with the celebrated Navier-Stokes equations:\(^\text{91}\)

$$\rho \frac{Dv}{Dt} = -\nabla p + \mu \nabla^2 v + \rho g$$

where \(\rho\) is the fluid’s density, \(D(*)/Dt\) is the substantial derivative, \(v\) the fluid’s velocity (a three dimensional vector), \(p\) the pressure in the fluid (a scalar), \(\mu\) the fluid’s viscosity, and \(g\) the acceleration due to gravity vector. These equations are also called the equations of motion as they describe the pressure and velocity field everywhere within a fluid.

Many undergraduate problems in fluid mechanics are simplified to be steady-state, have pressure be a function of only one spatial variable, and a one-dimensional velocity field dependent upon only one other spatial variable. Most problems such as these are readily solvable analytically. To solve the full set of Navier-Stokes equations for an arbitrary flow field can be extremely difficult, and super-computers have been purchased and dedicated to solving such problems with finite element and finite difference schemes.

Because of these difficulties, the field of fluid mechanics is rife with clever “tricks” and assumptions to make seemingly unsolvable problems solvable. An elegant technique might make a problem simple or require fewer operational steps to arrive at the final solution.\(^\text{92}\) Most are efficient and beautiful in their efficiency. Many times these techniques combine art and ingenuity and rely heavily on the intuition and “physical feel” possessed by the investigator.\(^\text{93}\)

One of the most powerful of these techniques involves scaling the terms of the Navier-Stokes equations to determine which are relatively small and may be neglected with minimal error. Creeping flow in a viscous fluid, for example, has negligible inertia and as a result the left side of the Navier-Stokes equations may be set to zero. The resulting equations may still not be amenable to analytical techniques, but this single assumption reduces the complexity enormously.


\(^{92}\) Armstrong, personal communication.
Another example of the power of simplification is introduction of the stream function, commonly represented as $\psi$. The stream function is powerful in at least two respects: 1) it can significantly reduce the operations required to attain a solution and 2) constant values of the stream function in space represent streamlines. This last advantage makes the stream function an intuitively attractive solution technique. It is natural to sketch streamlines of a flow field to help visualize the fluid’s motion. Lines where $\psi=\text{constant}$ actually show the paths of tiny particles trapped in the fluid flow.\textsuperscript{94} In two dimensional flow, the flow rate between any two points within the flow domain may be determined by finding the difference of the stream function at the two points.

To illustrate the utility of the stream function, let us consider creeping flow of a viscous fluid around a sphere.\textsuperscript{95} Without simplification this problem would certainly require numerical methods to determine the velocity and pressure fields. Within the problem statement, however, enough information is given that we can safely neglect inertial forces. This eliminates the left side of the Navier-Stokes equations. Representing the problem in spherical coordinates further simplifies the problem by recognizing that $v_r=0$ (flow is irrotational) and furthermore, that the velocity and pressure fields will have no $\phi$-dependence:

\[ v_r = -\frac{1}{r^2 \sin \theta} \frac{\partial \psi}{\partial \theta} \quad \text{and} \quad v_\theta = +\frac{1}{r \sin \theta} \frac{\partial \psi}{\partial \phi} \]

\textsuperscript{93} Weinstein, personal communication.
\textsuperscript{94} Bird et al, p. 130.
\textsuperscript{95} This solution follows that presented in Bird et al, p. 132.
For incompressible flow, the continuity equation (akin to conservation of mass) is

$$\nabla \cdot \mathbf{v} = 0$$

The typical solution of a fluid mechanics problem involves solving the continuity equation simultaneously with the equations of motion. Part of the power of the stream function is that it automatically satisfies the continuity equation. The equations of motion are simplified in a similar way: each is differentiated with respect to the other dimension (i.e., the \(r\)-direction equation of motion is differentiated w.r.t. \(\theta\), and the \(\theta\)-direction equation is differentiated w.r.t. \(r\)). The two equations are then subtracted and the pressure term is eliminated in the remaining equation. The last step is the substitution of \(\nu_1\) and \(\nu_2\) with the expressions containing \(\psi\). The result is a 4th order equation:

$$E^4 \psi = 0$$

where the \(E^4\) operator in spherical coordinates is:

$$\left[ \frac{\partial^2}{\partial r^2} + \frac{\sin \theta}{r^2} \frac{\partial}{\partial \theta} \left( \frac{1}{\sin \theta} \frac{\partial}{\partial \theta} \right) \right]^2$$

Two boundary conditions are realized when the no slip condition is applied at edge of the sphere; i.e., \(\nu_1\) and \(\nu_2\) equal zero at \(r=R\), the radius of the sphere. A third boundary condition is realized when the velocity is forced to be equal to the free stream velocity, \(\nu_\infty\), as \(r \to \infty\). In terms of \(\psi\), this last condition becomes

$$\psi \to \frac{1}{2} \nu_\infty r^2 \sin^2 \theta \quad \text{as} \quad r \to \infty$$

From here, a bit of intuition is required by recognizing that this third boundary condition suggests a form of the solution to be

$$\psi = f(r) \sin^2 \theta$$

where \(f(r)\) is some function of \(r\). Substitution of this expression for \(\psi\) into the above expression yields a fourth order, homogeneous differential equation which ultimately suggests \(f(r)\) has the form

$$f(r) = A/r + Br + Cr^2 + Dr^4$$
The third boundary condition above requires \( D = 0 \) and \( C = \frac{1}{2} v_\infty \). The first two boundary conditions require \( A = -\frac{1}{4} v_\infty R^3 \) and \( B = \frac{3}{4} v_\infty R \). Appropriate substitutions can be made to find the velocity distributions:

\[
\frac{v_r}{v_\infty} = \left[ 1 - \frac{3}{2} \left( \frac{R}{r} \right) + \frac{1}{4} \left( \frac{R}{r} \right)^3 \right] \cos \theta \quad \text{and} \quad \frac{v_\theta}{v_\infty} = -\left[ 1 - \frac{3}{4} \left( \frac{R}{r} \right) + \frac{1}{4} \left( \frac{R}{r} \right)^3 \right] \sin \theta
\]

The value of the stream function can also be calculated as a function of \( r \) and \( \theta \); plotting contours of \( \psi \) yields streamlines. These are shown below for a sphere of unit radius:

The intent here was to show how an elegant concept such as the stream function can be capitalized by making a difficult solution more attainable and obscure concepts more obvious and intuitive. Without introducing the stream function or a similar assumption,\(^{96}\) the solution of this problem would be very difficult or require numerical methods. Using the stream function, an analytical solution was obtained in short order. Furthermore, it is straightforward from this method to calculate streamlines from which patterns of the flow are easily recognized. These advantages make the stream function an elegant technique to solving many 2-dimensional problems in fluid mechanics.

\(^{96}\) Landau and Lifshitz (Fluid Mechanics, Volume 6 of Course of Theoretical Physics, Translated by J.B. Sykes and W.H. Reid, Pergamon Press, Oxford, 1959; p. 64) do not use the stream function in their solution of this problem. They do, however, reason a solution for the velocity field as \( \nu = \text{curl}(A) + \text{constant} \) where \( A \), some unknown vector which they later derive, is related to the stream function. If \( A \) is represented as the z-component of \( \psi \) in Cartesian coordinates, then the curl of \( A \) is \( \nabla \times A = v_x \hat{x} + v_y \hat{y} \), the local velocity component, i.e., an isocontour of the stream function for 2-dimensional flow in the x-y plane.
4.5 Aerospace – The 707

The 707 Prototype, model 367-80, nicknamed the Dash-80

“So when the jet engine came along, we had an engine that could make us go fast, but that guzzled fuel at a terrific rate. So that in the long range, we wanted to go fast so that you could go a lot of miles before you ran out of gas.”

–George Schairer, Boeing chief aerodynamicist

The airplane that ushered in the jet age for the commercial airline industry in the United States, the 707, is a remarkable aircraft. At the time of its development, it represented a complete departure from traditional commercial aircraft, and all large jets of today resemble its architecture.\(^{97}\) Its prototype set new speed records nearly every time it flew, averaging over 600mph on some trips.\(^{98}\) Between 1959 and 1969, worldwide air travel increased 100%, in large part due to the introduction of the 707.\(^{99}\)

The 707 represented a huge deviation from the commercial transports it eventually replaced. Boeing’s commercial transport at the time, the Stratocruiser, was developed just after WWII. It was based on the B29 bomber, bearing straight wings and four propeller engines, the 3,500-horsepower Pratt & Whitney R4360 Wasp Majors.

---

\(^{97}\) “Boeing: A Brief History”, http://www.boeing.com/companyoffices/history/boeing

\(^{98}\) “Boeing: A Brief History”, http://www.boeing.com/companyoffices/history/boeing

\(^{99}\) Brennan, p. 23
The Boeing Stratocruiser

With a top speed of 375mph and a cruising speed of 300mph, the Stratocruiser was state-of-the-art in commercial transport.

The 707 is derived from the B-47 and B-52 bomber jets built by Boeing for the U.S. Air Force both of which featured swept wings and jet engines. Despite these previous products, the 707 presented many design challenges for the Boeing team. At half a city block long, over 3 stories tall, and weighing 81 tons, the 707 cost over $16 million (1955 dollars) and eight years to develop. The company self-funded the entire project and kept the development as secret as possible. The prototype was even named the “367-80” to disguise it as a revised Stratofreighter.

The development and maturation of the jet engine made planes such as the 707 feasible. The engines ultimately deployed in the 707, four Pratt & Whitney JT3 Turbojets, produced 11,000 pounds of thrust each. These were the most efficient jet engines, in terms of pounds of thrust per pound of fuel, developed up to that time. Jet engines also made possible flight at high altitude where high speed flight was believed to be attainable.

By 1943, jet engines such as these had evolved enough that the idea of powering a commercial transport with them did not seem completely unfathomable. However, using conventional designs in wind tunnel tests at high speeds which were thought capable with these jet engines, air would compress at the front tip of the wing, creating a sort of bow wave. This reduced the lift-to-drag ratio so severely that an airplane with this configuration would consume enormous

---

100 Brennan, p. 1.
102 “Boeing: A Brief History”, http://www.boeing.com/companyoffices/history/boeing
amounts of fuel even if an engine were capable of overcoming the increased drag. These
effects also made the aircraft less stable in flight.  

Adolf Busemann of the Aeronautical Research Institute at Braunschweig anticipated these high
speeds and conducted his own wind tunnel experiments in 1935. Busemann found three
factors could greatly improve the L/D ratio at high mach numbers:  

- reducing the wing thickness
- increasing the wing span (or length) relative to its width
- sweeping the wings back

Busemann reported his findings in Rome in 1935 but his results went largely unnoticed, nearly
to the demise of the Allies. Following the surrender of Germany after WWII, the U.S. Air Force
sent a Scientific Advisory Board to Germany to acquire military secrets. On this board was
George Schairer, head of the Aerodynamics Department at Boeing. At Braunschweig, Schairer
found drawings of a plane with wings swept back at a startling 45° and data confirming its
superior L/D at high mach numbers.

Boeing, meanwhile was developing the XB-47, a high speed bomber for the U.S.A.F., which
was suffering debilitating L/D at high speeds. Schairer immediately wired the German swept-
back-wing findings to Seattle with instructions to modify the XB-47 accordingly. The change
worked exceptionally well; the manufactured B-47 was so fast that it only needed armament in
the rear because no jet fighter of the day could fly fast enough to attack from any other angle.
The 707 later shared this concept, sporting wings with a 35° sweep-back angle.

One of the most difficult challenges facing the developers of XB-47 was the location of the
generators. Originally, the engines were to be located on the fuselage to minimize the negative
impact the large engines would impose on L/D. The danger of a fire eliminated this as a
possibility and the designers were forced to put the engines back on the wings. Serendipitously,
Schairer unwittingly discovered the solution for this at Braunschweig as well. The German
design located the engines on pylons mounted under the wing with the intake of the engine a
fair distance in front of the leading edge of the wing.

This had several positive effects. Placing the engine away from the high velocity air near the
airfoil reduced the drag caused by the engine and did not hamper lift. Also, a documented
problem with swept wings is the dreaded one-winged stall due to separation near the ailerons.

---

104 Brennan, p. 12.
Today, boundary layer theory suggests placing a 'boundary-layer fence' on the wing can avoid such a stall. These boundary-layer fences are small, solid pieces of sheet metal mounted vertically on the leading edge of the wing which prevent cross-flow near the surface of the wing. Although it was not known at the time, the pylon mounting of the engines created a similar effect and stall characteristics were significantly improved. Mounting the engines below the wing made for easier maintenance and a quieter ride. Naturally, it offered better fire protection than the alternatives as well.

The 707 incorporated all these and other advantages as well as more changes (like thrust reversers, fuel tanks in the wings, and tricycle landing gear) to make it appropriate for commercial transport. It and its military counterpart, the KC-135, have been dramatically successful. The 707 proved to be an extremely fast mode of transport, setting city-to-city records almost everywhere it flew. Its stability and control delighted pilots; test-pilot Tex Johnston even performed two "barrel rolls" during a demonstration in Seattle. Over one thousand B707's were sold between 1954 and 1991 when the B707 production line was closed. The B707 gave the U.S. the lead in commercial jet manufacturing.

The design of the B707 is an excellent example of how an elegant solution relieves nearly all design pressures simultaneously rather than just balancing one constraint against another. As aerospace products often are, the B707 had many conflicting goals: the optimization of one resulting in the decline of another. Other commercial aircraft of the era were simply optimizations within the existing solution space and were forced to obey the same constraints. The many innovations incorporated into the 707 broke these paradigms and moved the 707 into an entirely new realm of performance. As a testament to its superiority, the 707 created the architecture followed by every successful large commercial jet made to this day.

105 Schlichting, p. 253.
4.6 Computer Software Product – The Apple Desktop

"Like many other brilliant ideas, once introduced it is unimaginable to conceive of working without it."

- Steven Levy of the desktop model

The Apple Desktop operating system needs little in the way of an introduction. Its model is incorporated not only into computers made by Apple, but almost all major commercial operating systems sold today, including Microsoft products Windows NT™, Windows 95™, and Windows 98™. It is truly powerful as it allows a user to navigate the domain of the computer. It depicts a foreign and non-intuitive environment in a familiar way in which the user's real-world instincts produce expected results.

Many experts in computer science rank the development of the Apple Desktop operating system as one of the greatest innovations in computer systems. David Gelernter lists three of the “biggest hits” for software as Algol 60, object oriented programming, and the Apple Desktop. He claims the desktop-style OS is one of the key inventions of the computer industry. Frederick P. Brooks stated, “One of the most impressive developments in software during the past two decades has been the triumph of the Windows, Icons, Menu, Pointing interface.”

106 Gelernter, p. 87.
108 Gelernter, p. 32.
One of the key aspects of the Apple desktop is its use of the physical desktop as a mental model. Through this metaphor, users grasp the concepts almost immediately. Even the use of icons to "hide" documents or programs, a construct which extends the physical desktop metaphor, is easily understood after only a few minutes of exposure.

Many of the elements of the desktop metaphor (using a mouse to point, overlapping windows, dragging and dropping, and the high resolution screen, etc.) were first implemented in the Alto, possibly the world’s first PC. "The first Alto was assembled in 1973; it initiated the elegance revolution that led directly to the Macintosh, Windows, and today’s computer world."\textsuperscript{110} It was invented by scientists, Butler Lampson, Chuck Thatcher, Edward McCreight, and Alan Kay, at the Xerox Palo Alto Research Center. Despite its advances, Xerox rejected the idea of selling the Alto and failed to market it. Apple Computer Company later signed an agreement with Xerox to see some of PARC's discoveries. Steven Jobs was in that delegation from Apple, and he recognized the elegance of these concepts. After seeing PARC research, Jobs decided to incorporate many of their inventions into the follow-on for the Apple II, the Lisa. Many of the ideas were improved upon and others were added (e.g., the menu bar and double clicking). The Lisa, overpriced and under-powered, was not a rousing success. Many of these shortcomings were overcome and the following product, the Mac, started the desktop revolution.\textsuperscript{111}

A critical element in the desktop metaphor is the use of overlapping windows. First invented by Ivan Sutherland as a graduate student at MIT in the early 1960s, windows were introduced in a program called Sketchpad. Douglas Engelbart and his team from the Stanford Research Institute built another similar system; in both systems, however, windows were not allowed to overlap. Alan Kay, who spent much of his efforts developing tools to help children interact with a computer, invented overlapping windows at Xerox PARC. Without windows, the size of the screen limits the amount of information the computer can show you at a time; overlapping windows can significantly increase the user’s ability to communicate with the computer.\textsuperscript{112}

The pointing mechanism was also introduced commercially in the Apple desktop. Consistent with the desktop metaphor, the mouse, invented by Engelbart, allows a user to point and chose objects without calling them by name. By contrast, DOS requires you to type in the file name in full without any typos.

\begin{footnotes}
\item[110] Gelernter, p. 74.
\item[111] Gelernter, p. 80.
\item[112] Gelernter, p. 80.
\end{footnotes}
In addition to this physical metaphor to guide the user, the architects of the Apple desktop operating system encouraged application writers to use the same user interface by placing the controls for the interface in easily accessed read-only memory.\textsuperscript{113} As a result, nearly every application uses the menu bar in approximately the same way, creating consistency for developers and users. It has become so expected of third party applications that those which do not conform are criticized heavily by the trade magazines. This consistency is a significant delight for users, many of whom claim it as one of the main reasons Macs are so easy to use. Without any change of the architecture, consistency alone allows the user to extract greater function from any piece of software conforming to the desktop metaphor.

Two key lessons toward elegance are taught by the Apple Desktop model. First, the use of a more familiar metaphor, the physical desktop, to guide the user through an unfamiliar domain, the world of a computer, can be a powerful and simplifying construct. Secondly, consistency with this metaphor throughout applications and throughout the interface are critical to its success — users feel comfortable with the technology, intuitively know how the system will respond to their input, and can extract greater function due to this consistency.

The Apple desktop is another good example in which elegance does not guarantee or imply success. Ultimately, the idea of the desktop operating system persevered and has found its way into many products. Despite bringing this innovation to market first, Apple has failed to dominate the marketplace. There has been much written regarding the likely causes of this failure and interesting debates continue on the subject.\textsuperscript{114} None of this, however, detracts from the unquestionable elegance of the desktop model.

\textsuperscript{113} Brooks, p. 264.

\textsuperscript{114} Gelernter, for example, suggests that people are afraid to admit they like the elegance of the desktop model, that it was too “cute” to be powerful and appropriate for business. p. 31.
4.7 Architectural Structure – The Roman Pantheon

"Angelic and not human design..."
-Michelangelo on the Roman Pantheon

A Pantheon, a temple dedicated to all the gods of a people, is a reverent place seeking union between deity and human. The Roman Pantheon is the finest of all and a tribute to the Roman Architects, Artists, Engineers, and Laborers who erected it. The fact that it has withstood the test of time over 1800 years and is still magnificently preserved is a testament to those same creators. Its dome was the largest existing span until the Breslau dome was built in 1913.\textsuperscript{115} Despite some serious cracks in the dome, it still stands in all its glory, effectively unchanged since ancient times. Perhaps its most serious threats are the plundering of its materials\textsuperscript{116} and the downtown Rome traffic. It is “perhaps the most visionary and awe-inspiring structure the Romans built.”\textsuperscript{117} Even by today’s standard it is a remarkable building; given the materials and tools available at the time, it is an incredible architectural feat.\textsuperscript{118}

The first Roman Pantheon, begun by Marcus Agrippa, was marred several times by fire. Roman buildings around the time of Christ contained limestone rock which cracked badly in

\textsuperscript{115} Billington, p. 177.
\textsuperscript{116} The porch, for instance, now has timber for its cross-members. At the time of construction these were bronze which was later usurped by armies needing metal for their cannons. Source: Moore, David, \textit{The Triumph of Roman Concrete}, 1995.
\textsuperscript{117} http://web.kyoto-inet.or.jp/org/orion/eng/hst/roma/pantheon.html
fires such as those that struck Rome between 60 A.D. and 110 A.D. The Pantheon was repeatedly damaged by these fires and eventually required replacement except for the foundation and parts of the lower porch.\footnote{Moore.}

The replacement was undertaken by Hadrian and construction progressed between 118 A.D. and 128 A.D., though the precise time period is debated by experts.\footnote{Ward-Perkins J. B., \textit{Roman Imperial Architecture}. Penguin Books, New York, 1985, p. 111.} The tombs of Italian kings, important Popes, and the painter Raphael are housed within its walls. Hadrian rebuilt the Pantheon much like the original and for the same intent, i.e., for all the gods of the Roman people. In 609A.D. Pope Boniface appropriated the Pantheon for the Roman Catholic church, dedicating it to all Martyr Saints and the Virgin Mary.\footnote{http://web.kyoto-inet.or.jp/org/orion/eng/hst/roma/pantheon.html}

The Pantheon is geometrically simple and correct. Its shape is that of a sphere 43 meters in diameter inscribed by a cylinder and hemisphere:

\begin{center}
\includegraphics[width=0.2\textwidth]{pantheon_diagram.png}
\end{center}

It has a ratio of 1:1 in all directions. This geometric shape was not new for Roman architecture, but its magnitude is unique and remarkable. The front porch is supported by Corinthian columns and has a Greek-style temple facade. The main entrance faces north as was dictated by tradition and opens into the interior by way of a double door, 21 feet tall and made of bronze.\footnote{Moore.} The exterior of the dome is seven rings of decreasing outer diameter, stacked one upon another. Thin brickwork is beautifully layered upon the exterior walls.\footnote{http://web.kyoto-inet.or.jp/org/orion/eng/hst/roma/pantheon.html}
Within the structure, there are two levels; the cornice separating the first and second levels divides the levels in the ratio of $\sqrt{2}$-to-one.\textsuperscript{124} The floor is open and made of marble with a circular and cubic pattern. It is contoured so that rain water admitted through the oculus drains away. The walls are also of marble, yellowish-brown, white, green, and reddish brown in color.\textsuperscript{125} Within the walls are many cavities and chambers, distributed amongst the levels perhaps to reduce weight and construction material.\textsuperscript{126} Also built into the walls are several larger niches which house the remains of Italian kings and popes. These wall openings have archways forming their upper perimeter to support the wall above the opening. The interior of the dome contains five rows of tiled coffers, 140 in total.

At the center of the dome is the oculus void, 5.9 meters in diameter with walls 1.4 meters thick.\textsuperscript{127} The oculus has at its perimeter a set of three rings which balance and distribute the

\textsuperscript{124} http://web.kyoto-inet.or.jp/org/orion/eng/hst/roma/pantheon.html
\textsuperscript{125} http://web.kyoto-inet.or.jp/org/orion/eng/hst/roma/pantheon.html
\textsuperscript{126} Moore.
\textsuperscript{127} Moore.
compressive forces concentrated here from the rest of the dome. Open to the exterior, the oculus admits light and rainwater. A spot of sunlight transverses the wall throughout the day, and at night the moon takes a similar charge, brightening the interior. Rainwater was viewed as the Romans as the "most wholesome of waters" and its entrance into the Pantheon was not viewed as an inconvenience.\textsuperscript{128} An exterior staircase leads to the oculus from the rings of the dome.

The material of construction for the walls and dome of the Roman Pantheon is unreinforced concrete. The fact that a span of this size can be constructed of such a material is truly phenomenal. The Romans wisely minimized the necessary materials by incorporating voids and niches into the walls and grading the wall thickness as the top of the dome was approached (from 5.9 meters at the base to about 1.5 meters at the oculus). Into their concrete, the Romans used lightweight stone as the filler, making the volcanic ash, hydrate lime, and rock mixture as light as possible.\textsuperscript{129} Arches and other curved surfaces were employed to reduce concentrations of stresses.\textsuperscript{130} The light stone arches create an open atmosphere, in contrast to Greek architecture which relied heavily upon large stone beams.\textsuperscript{131}

The Pantheon sits on unstable, marshy clay, creating a serious problem for the builders. To overcome this design issue, the Romans put a second ring around the first foundation to keep it intact during settling of the ground.\textsuperscript{132} Construction of the dome was a painstakingly slow process, stacking each ring upon the one below it and waiting enough time between stacking to ensure proper cure of the lower ring.


\textsuperscript{129} Mark, R. and Hutchinson, P; "On the Structure of Pantheon", \textit{Art Bulletin}, March 1986, p. 29. In their analysis, they show that if the density of the concrete were only about 35% greater, the Pantheon would be subjected to 80% greater stress.

\textsuperscript{130} Moore.

\textsuperscript{131} Billington, p. 195.

\textsuperscript{132} Moore.
The Roman Pantheon is an elegant marriage of materials, design, and art. The "building projects a solemn yet comforting image of engineering sophistication, longevity, and design harmony."\textsuperscript{133} It exemplifies "the utilian power of Romans so basic to the engineer's ideal of structural expression."\textsuperscript{134} In considering its purpose as a temple to all gods, it is rich with symbolism: "The structure is meant to be a microcosm of the cosmos ... The coffered dome symbolizes the span of the heavens. The ... oculus at the top represents the Eye of Jupiter and connects the earthly temple with the celestial ... realm of the gods."

As an architectural feat, the Roman Pantheon is an engineering marvel. Its shape is of perfect proportions and symmetry. "Without symmetry and proportion, there can be no principles in the design of any temple."\textsuperscript{135} In designing visual interfaces for computer systems, Mullet and Sano believe the relationships amongst the parts of a system are often more important than the parts themselves.\textsuperscript{136} Proportion is critical to harmonious design.

The Pantheon's acoustics follow the principles set by Vitruvius: "...the voice, uttered from the ... center, and spreading and striking against the cavities of the different vessels, as it comes in contact with them, will be increased in clearness of sound, and will wake an harmonious note in unison with itself."\textsuperscript{137} Its expanse is incredible for the materials and methods of the day. Despite failing many design principles held today, it stands as a testament to its creators, much

\textsuperscript{133} http://web.kyoto-inet.or.jp/org/orion/eng/hst/roma/pantheon.html
\textsuperscript{134} Billington, p. 195.
\textsuperscript{135} Vitruvius, p. 72.
\textsuperscript{136} Mullet and Sano, p. 51.
\textsuperscript{137} Vitruvius, p. 143.
to the dismay of many analysts who would have predicted its demise many centuries ago.\textsuperscript{138} As a temple to “all the gods”, the Roman Pantheon stands as an elegant marvel of man, perfectly shaped, of immense proportion, and wonderfully matched to its purpose.

\textsuperscript{138} Moore.
5. Characteristics of Elegant Designs and Architectures

The examples described above all share the traits common to all elegant solutions. Reiterating from Section 3.1, the two common threads of elegant solutions are:

1. the system must function according to its stated purpose; and
2. the design pressures constraining the system design must be simultaneously relieved.

Although these are useful criteria against which to evaluate a system’s elegance, there are other characteristics of many elegant designs and architectures that are worth more investigation. Although they may not be common to all elegant solutions, they are mutual and pervasive enough to warrant further discussion. One example of these characteristics is that elegant solutions are usually **enduring**. Counter examples could be found, but in general an elegant solution enjoys a long life of employment; although, the life may only be long relative to similar products. The Apple Desktop has been featured in computer systems for a long period relative to other operating systems but only a tiny portion of the time the Pantheon in Rome has stood.

The more important of these characteristics are summarized below.

### 5.1 Simplicity and Efficiency

*Tis the gift to be simple, tis the gift to be free
*tis the gift to come down where we ought to be
and when we find ourselves in the place just right
*Twill be in the valley of lore and delight

*When true simplicity is gained
To bow and to bend we shan’t be ashamed
To turn, turn will be our delight,
*Til by turning, turning we come out right.

-- Shaker Hymn, *Simple Gifts*

The fact that **simplicity** is overwhelmingly present in elegant solutions should come as no surprise from the second criterion of elegant solutions. Simplicity is a design pressure in nearly every system imaginable. This is a consequence of several factors. First, most systems are made of material components which must be fashioned and combined by conventional methods. These methods dictate a maximum level of precision and complexity which are attainable from the methods available. The number of parts in a car, for instance, is limited by the time and skill of the robots and humans available to assemble the vehicle. Secondly,
designs are the products of humans. Because of our limited memory and cognitive abilities, a design extending beyond these abilities becomes too complicated to comprehend at once. Hierarchy and abstraction are extremely powerful tools for dealing with this complexity, but they put additional strains on a system (e.g., additional manpower to design the architecture at the higher level of abstraction) and ultimately have their limits as well. Simplicity impacts nearly every aspect of the product’s life-cycle. Beyond the design and assembly, ease of use and maintenance are directly related to a system’s complexity. Simplicity is inarguably a desired trait of nearly all systems.

Unfortunately, the typical means to increasing a system’s function is to increase its complexity.\textsuperscript{139} When an engineer is tasked with adding a new feature to a product, the easiest path to accomplishing this is to add it directly on top of the existing product. Rarely are features or components removed (when’s the last time you heard of new release having fewer features?). Fortunately, “beauty is the ultimate defense against complexity.”\textsuperscript{140} The impact of an elegant solution on complexity and function is crudely shown in the schematic below:

As a given architecture is exploited, functions are added by increasing the complexity of the product. The Palm III version of the Palm Pilot is an excellent example. The original Palm Pilot was purposely simple. As future versions were released, additional features and add-ons have been added under basically the same architecture. Increased computing power has permitted these add-on, not a significantly improved architecture. When an elegant solution is introduced, however, the complexity drops and function is not compromised. Graffiti in the original Palm Pilot serves as a good example. In the handwriting-recognition software of previous PDAs, better performance only came from more sophisticated algorithms and increased computing power. The Graffiti breakthrough in the Palm Pilot greatly simplified this task but at the same time did not concede functional requirements.

\textsuperscript{139} Meyer and Lehnerd, p. 88.

\textsuperscript{140} Gelernter, p. 22.
Efficiency is a concept related to simplicity. It has at its core the concept of doing more with less. David Billington explains that efficiency deals with the amount of input required. In structural engineering, this might be the raw materials required; in space craft design, the consumed valuable resource might be the mass of the subsystem; in computer programming, runtime memory usage, processing time, compile-time resource utilization and the typing time of the programmer are some of the resources to be minimized. Elegance, on the other hand, involves minimizing the impact of all design constraints, not just system inputs. This fine point is missed by some authors who define elegance as getting the most output for the least input.\textsuperscript{141}

Simplicity is a trait that should always be pursued when seeking elegant solutions. Since simplicity is such a critical component of many elegant solutions and part of the intent of this thesis is to help an aspiring architect assess the elegance of his or her solution, a short discourse on the methods of defining complexity is warranted.

5.1.1 Defining Complexity

"Everything should be made as simple as possible, but no simpler."
– Albert Einstein

Intuitively, complexity is associated with things that we have difficulty understanding. In systems, we usually think of many components or intricate interfaces between the components as being the causes of complexity. Even with simple components, an incredible diversity of behavior is possible through the interaction of many of these components.\textsuperscript{142} The most simplistic measures of complexity, therefore, deal with the number of components and interaction. Boothroyd and Dewhurst, for example define a complexity factor, $F_c$, as:\textsuperscript{143}

$$F_c = 3 \sqrt{N_p + N_t + N_i}$$

where $N_p$ is the number of parts, $N_t$ is the number of types of parts, and $N_i$ is the number of interfaces in the system. Yates describes a slightly more complicated but similar model of complexity. He suggests there are five factors which contribute to a systems complexity:\textsuperscript{144} 1) A large number of parts or interactions, 2) Significant interactions, 3) Non-linearity, 4)
Asymmetry, and 5) Nonholonomic constraints.\textsuperscript{145} Such measures as these are adequate for many purposes but fall short in capturing all our perceptions of complexity in systems. For example, these authors may call ‘complex’ a system with many identical parts and identical interfaces. A large wall made up of many bricks is far less complex than, say, a small radio but these definitions fail to capture this intuition.

Other measures have been suggested which relate the complexity of a system or object to the length of time required to arrive at the solution as computed by a clever program independent of the programmer. Such a measure, called “computational complexity” is dependent upon the hardware used during the test and fails to explain complexity in the context of complex systems.\textsuperscript{146}

Algorithmic information content (AIC) is a slightly improved but still deficient method of describing complexity. It is defined as the length of a computer program that can print out the information being measured for its complexity.\textsuperscript{147} Long, random strings, therefore, have the greatest AIC; very short strings or repetitions of short strings have very low AIC. Although this measure begins to help describe the system, it also falls short of our preconceived notions of complexity in complex systems. A purely random bit string would not be considered complex in the world of systems. As an example, consider an industrious neighborhood boy who built a tree house from materials discarded from a renovation project. The resulting structure was the combination of many types and shapes of materials, causing one neighbor to wonder if the wood “had been blown up there by the last windstorm.” Despite its randomness, few would call the architecture of the tree house complex. AIC has the added disadvantage of being uncomputable: improved but yet unavailable compression algorithms might be able to further reduce the AIC.

Gell-Mann suggests a better measure of complexity, called the \textit{effective complexity}.\textsuperscript{148} An object’s effective complexity is a quantified measure of how long a message is needed to describe certain aspects of a system. More specifically, it is the length of a concise description of a system’s regularities.\textsuperscript{149} The total information in a system is made up of its randomness and its regularities. Effective complexity is a measure of the latter and coincides nicely with our perception of complexity in systems. Effective complexity is at its maximum at intermediate values of AIC. As the effective complexity of a system becomes smaller, shorter

\textsuperscript{145} This last component of complexity results when a part or several parts of a system have a high degree of freedom, and behavior or control of those parts can be difficult to predict.

\textsuperscript{146} Gell-Mann, p. 28.

\textsuperscript{147} Gell-Mann, p. 35.

\textsuperscript{148} Gell-Mann, p. 50.
and simpler descriptions (or schema) are capable of describing its regularities. A system containing many identical components with simple interfaces between the components, for example, would have low effective complexity. Redundant Array of Inexpensive Disks (RAID) technology, for instance, is an elegant, simple, and cheap solution which greatly increases the reliability of data storage in hard drives. As the name implies, however, the number of components is increased over previous methods.

The concept of effective complexity has been suggested by other authors. Herbert Simon, for example, provides an example illustrating effective complexity in which a matrix consisting of many elements can be compressed into a much smaller matrix by capitalizing on the redundancy in the original structure. In 1948, Weaver described three regions of complexity: organized simplicity, organized complexity (akin to effective complexity), and disorganized complexity (high AIC as in a random bit string). The relations between these measures are summarized below:

<table>
<thead>
<tr>
<th>Weaver:</th>
<th>organized simplicity ↔ organized complexity ↔ disorganized complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gell-Mann</td>
<td>low effective complexity ↔ high effective complexity ↔ low effective complexity</td>
</tr>
<tr>
<td>AIC:</td>
<td>low AIC ↔ mid AIC ↔ high AIC</td>
</tr>
</tbody>
</table>

Another important factor raised by Gell-Mann is that of granularity. If effective complexity is determined by the length of a concise description of its regularities, the detail at which those regularities are described will ultimately impact the length of the description. If, for example, one wishes to compare the effective complexities of the B707 and the B47, it is important that the level of abstraction is the same in both cases; it would be unfair to describe the B707 down to each nut and bolt but yet only describe the B47 at the system architecture level. This level of detail is granularity. Coarse granularity uses higher levels of abstraction; fine granularity considers more minute details.

The last important concept to be considered here is entropy. As mentioned above, total information for a system is made up of regularities and randomness. Effective complexity is a measure of the former; entropy is a measure of the latter. As more and more is learned

---

149 Gell-Mann, p. 50.
152 Gell-Mann, p. 29.
about the behavior of a fixed system, less of its behavior is attributed to random events and effective complexity increases (since more regularities can be described). In essence, "effective complexity measures knowledge."\textsuperscript{154} This is very evident in the scientific enterprise in which regularities are sought in an apparent sea of random behavior (see Section 5.1.4 for more discussion on this topic). Entropy, on the other hand, can be shown to be equal to ignorance within a constant of proportionality.\textsuperscript{155} A very elegant measure of ignorance, $I$, was defined by Shannon as (in units of bits):\textsuperscript{156}

$$I = -\sum_{r=1}^{n} p_r \log^2 p_r$$

where $p_r$ is the probability of event $r$. Since all the $p_r$ must sum to one, $I$ will be greatest as $n$ increases and as the probabilities become more equal (i.e., you do not know what might happen).

These concepts become useful when considering what constitutes complexity in systems, particularly when the attractiveness of two alternative systems is being evaluated. Simplicity is such a critical element in many elegant solutions, gaining some familiarity with its measures is a worthwhile investment. For two systems meeting the defined functional requirements of the system equally, the system with the least effective complexity is the more elegant one.

Simplicity has a different significance and a different quintessence depending upon the type of product or system being considered. Furthermore, disciplines have evolved at varying rates in trying to deal with complexity. Therefore, a brief but closer look at simplicity in Computer Systems, Commercial Products, the Scientific Enterprise, and Organizational Design is considered in the subsequent sections.

\textbf{5.1.2 Simplicity in Computer systems}

"Measuring software productivity by lines of code is like measuring progress on an airplane by how much it weighs."

-- Bill Gates, chairman and CEO of Microsoft Corporation

Engineers whose medium of creation is the physical world deal with the physical laws present there, and these laws help keep the system in check. When an overweight aircraft can not takeoff or a bloated ship can not navigate a port, the failures of these systems are immediately recognizable by the designer and all others. The domain of computer systems, however, is

\textsuperscript{154} Gell-Mann and Lloyd, p. 49.
\textsuperscript{155} Gell-Mann and Lloyd, p. 46.
\textsuperscript{156} Gell-Mann and Lloyd, p. 45.
without these physical constraints and, as Bill Gates' quote suggests, controls on a computer system often do not promote good design. Furthermore, a bloated piece of software is not immediately recognizable by the user nor perhaps even the designer. Elegance is usually not a consideration, nor is it rewarded. "Programmers do not strive for elegance; ... management encourages complexity."\(^{157}\)

The result is inelegant, overweight, inefficient software and systems that ultimately exasperate users. Nearly every user of computer systems has a story like Gelernter, who said about a particularly kludgy word processing package, "You wish software like this came with a rear end so you could kick it."\(^{158}\) Even before the product is released, things can get so out of control that 75% of all computer system development undertaken is either never completed or the resulting system is not used.\(^{159}\) Computer systems are in dire need of a healthy injection of elegance.

Computer systems do have their design pressures: bandwidth, disk space, power limitations, etc.,\(^{160}\) but because hardware has improved at such an incredible rate over the last two decades, these original limitations can be irrelevant once the developed system is ready for release. Brooks points out that many of the constraints he faced in 1975 were immaterial in 1987; virtual memory and cheap real memory, for example, had made obsolete the tough decisions on size of transient areas.\(^{161}\) Indeed, an important tradeoff considered in designing a computer system is whether to "invest in intellectual elegance or computer power."\(^{162}\) Undergraduate course in computer system engineering remind students that \(d(\text{technology})/dt\) is very large in this domain. As a result, "buying new technology [e.g., faster hardware] is more efficient than being clever."\(^{163}\) This is good advice in many circumstances, but it's not a valid excuse for kludgy software.

There are reasons for optimism, and elegance has its advocates. Researchers and practitioners like Tony Hoare, David Pensak, David Gelernter, and others have raised the expectations on computer systems to give more consideration to simplicity, elegance, and beauty in computer systems. Their urgings and teachings resound with the sentiments like the following:

\(^{157}\) Bentley.
\(^{158}\) Gelernter, p. 48.
\(^{160}\) Anthes.
\(^{161}\) Brooks, p. 238.
\(^{162}\) Schuett et al.
• "I gave desperate warnings against the obscurity, the complexity, and the over-ambition of the new design..."\(^{164}\)
• "The unavoidable price of reliability is simplicity."\(^{165}\)
• "If you build a big enough program, it is almost impossible to make it come out right."\(^{166}\)
• "The fastest code in the world to develop is the code you didn't have to write."\(^{167}\)

This is not to say that the domain of computer systems is without its elegant solutions. The Apple Desktop example given above is a refreshing model of elegance against which other operating systems should measure themselves. Some searching and sorting routines (e.g., the binary search and C.A.R. Hoare's Quicksort) are amazingly efficient and can reduce operational steps by four orders of magnitude on large databases.\(^{168}\) These simple examples can suggest elegant algorithms for solving larger problems.

Ethernet is also a refreshing detour into elegance. Designers of Ethernet "choose to achieve reliability through simplicity."\(^{169}\) As a result, this network medium has enjoyed widespread success. Another interesting, and maybe controversial, example of elegance in computer systems is the Internet worm that plagued systems throughout the Internet on November 2, 1988.\(^{170}\) This virus certainly achieved its goals (raising havoc and demonstrating the weakness of general security measures) and did so with remarkable grace. In less than 100 lines of code, the virus penetrated computer systems, users' accounts, searched for other hosts and users, replicated itself and finally retransmitted itself for wider spread. It did all this while concealing itself (by zeroing its argument vector and deleting all partially transmitted files) and protecting itself by immediately forking a copy of its process. Here, the design pressures are readily apparent: stealth, small size for quick transmission, fast execution, and compatibility in several environments.

Liberated by their environment, computer systems have accomplished a great array of diverse tasks and have grown in both function and complexity. The hardware explosion has further promoted this growth. A return to elegance is on the horizon, however, and it remains true that computer systems espousing the characteristics of elegant solutions are, in general, more successful and enduring than their kludgy counterparts.

\(^{164}\) Tony Hoare, Turing Award Lecture, 1980.
\(^{165}\) Tony Hoare, Turing Award Lecture, 1980.
\(^{166}\) Gelernter, p. 25.
\(^{167}\) Anthes.
\(^{168}\) Anthes; and Gelernter, p. 55.
5.1.3 Simplicity in Other Commercial Products

"An engineer is a man who can do for a dime what any fool can do for a dollar."

— Anonymous

Unlike computer systems, systems forced to operate in the real-world environment continually face design pressures. As a result, elegant solutions abound and many have at their core the concept of simplicity. "Low levels of complexity in product and process design are tangible measures of elegance. The virtues of reducing complexity, of course, go beyond aesthetics." Consider the conductive carrier in electrophotography given as an example above. It was a remarkably simple solution to a very difficult and elusive problem. Simplifying the handwriting-recognition software in the Palm Pilot and reducing its functions to only what was necessary were undoubtedly two of main reasons for its overwhelming success. Complexity, on the other hand, drives costs by increasing the number of parts, the different types of parts and required manufacturing processes. Complex products with human interfaces are less intuitive and can hide function from the user. The benefits of simplicity pervade every aspect of a product’s life-cycle.

Simple solutions almost always prevail in the end. In trying to claim the prize of the Longitude Act of 1713, for example, John Harrison developed in his chronometers much of the technology that is present in today’s analog watches. One particularly vexing problem was that of lubrication – when the lubricant heated up or cooled down, the change in viscosity would cause the chronometer to speed up or slow down. In one of his early clocks, the H-1, he solved this problem by making the clock largely out of wood (a material he knew well), and those parts that required lubrication he made out of lignum vitae, a wood which exudes its own grease. The clock, containing many other advances, kept time to within one second over an entire month, whereas most clocks of the era varied by more than one minute per day. Elegant ideas such as this ultimately led John Harrison to claim the prize which eluded the greatest scientists of the day for over sixty years.¹⁷²

There are real-world disciplines that suffer from a similar affliction as computer systems. In analog design theory, for example, Ioffe suggests “the theory that the simplest option is usually the best is not new.” However, he goes on to say, the “abundance of literature and inexpensive parts on the market often lulls designers into thinking they have found a solution to their design problem, when, in fact, they have simply overcrowded their designs with

¹⁷¹ Meyer and Lehnerd, p. 99
Designers of all systems must constantly be aware of the dangers of complexity and retain it only when no other solution is possible. Simplicity is so prevalent in nearly all elegant solutions that the system architect must pursue it relentlessly.

5.1.4 Elegance in the Scientific Enterprise

"Wonderful, but not incomprehensible."

– Simon Stevin after devising a proof of the law of sines.

Perhaps nowhere else is the pursuit of elegance more ardent than in the scientific enterprise. Scientists are, in essence, driven by this chase. "The central task of a natural science is to make the wonderful commonplace; to show that complexity, correctly viewed, is only a mask for simplicity; to find pattern hidden in apparent chaos." Elegance is the primary tenet of scientific synthesis. When a complex set of behaviors is explained by a simple, far reaching explanation that is readily understandable, the power of the scientific method comes to bear. To make these explanations more understandable and more intuitive but yet more powerful is the overruling duty of science. When it is attained, elegance brings to the investigator delight and contentment in the same moment. After finding a simple proof of the law of sines, Simon Stevin was so proud of the accomplishment he had a vignette engraved with the substance of the proof and the quote translated at the commencement of this section:

As seen in the proofs of the Pythagorean theorem, an elegant proof is powerful but condenses a lot of understanding into a relatively short description. This powerful compression is found in many of the great theories of science. Gell-Mann gives the theory of electromagnetism by

\[ F = \frac{1}{4} F_{\mu \nu} F^{\mu \nu} \]

\[ J^\mu = \partial_\mu A^\nu - \partial_\nu A^\mu \]

\[ A^\mu = e A^\mu_\alpha (\partial_\alpha + \omega_\alpha^\beta \epsilon_\beta^\mu) + \frac{e^2}{4} F_{\alpha \beta} (\partial_\alpha A^\beta - \partial_\beta A^\alpha) \]

\[ e A^\mu = \frac{1}{2} F_{\alpha \beta} (\partial_\alpha A^\beta - \partial_\beta A^\alpha) \]

James Clerk Maxwell in the 1850s and 1860s as an example. Combining the work and separate laws by Orsted, Faraday, and Coulomb, Maxwell was able to elegantly describe the entire scope of electromagnetism in just a few lines of equations. Similar compression was achieved by Einstein’s equation describing the general properties of gravitation in the general theory of relativity. As in the example of the stream function solution to fluid mechanics problems, these elegant theories make the apparently complex more understandable, more intuitive, and easier to visualize.

The concept of effective complexity captures nicely the idea of elegance in the scientific enterprise. The laws and relationships generated by scientists are schema, the length of which describes a theory’s effective complexity. By making a schema more thorough, it explains more behavior but also makes the schema more complicated. In this sense, it increases its effective complexity but decreases the entropy term since there are fewer possible entities possessing the attributes of the more descriptive schema. However, occasionally a more powerful schema comes along which reduces both the entropy term and the complexity of the schema. Maxwell’s equations in electromagnetics, Einstein’s theory of relativity, and Navier-Stokes equations in fluid mechanics are excellent examples of such powerful and elegant schema.

Given science’s long pursuit of elegance, designers and architects of complex systems would be wise to refresh themselves with the tenets of the scientific method. Science is fortunate that it (usually) is free of economic and political pressures compared to business products and programs. Undoubtedly this helps make its search for elegance a clear, unmistakable goal. To as great an extent as possible, this purity of pursuit should be replicated in commercial endeavors.

5.1.5 Simplicity in Organizational Design
Somewhat similar to computer systems, an organization faces fewer design pressures than the more typical engineering products such as automobiles, copiers, and photographic film. Perhaps because of this, little attention has been devoted to creating truly elegant designs of organizations until recently. Recent literature focuses on not only how to best design an organization, but also how to quickly change an organization’s design, how to equip the

175 Gell-Mann, p. 81.
176 Gell-Mann, p. 88.
177 Gell-Mann and Lloyd, p. 50.
178 Gell-Mann and Lloyd, p. 50.
organization to deal with this change, or when to optimally make the change. It might be that this focus on how to change an organization suggests that an elegant design is still lacking.

There are exceptions and elegant organizations do exist. Reddin gives the Toronto Symphony Orchestra as an example. With 100 to 1 aspect ratio (width to height of the organization structure), high job satisfaction, and beautiful products to its credit, it is difficult to argue that the TSO is not an elegant organization. Everyone understands the measure of success of the organization. “An emphasis on good design and clear outputs leads not to low job satisfaction but to a lively place in which to work.” Provost Joel Moses gives the catholic church as another example of an elegant organizational architecture. Tens of thousands of people within the worldwide organization are effectively managed with only four levels of hierarchy: deacons, priests, bishops, and the Pope.

Examples such as these and a greater focus on efficient organizations will lead researches and managers toward more elegant, simple designs. The inertia of most organizations, however, guarantees this to be a slow process.

### 5.2 Form Follows Function

“Well-designed products achieve elegance in both form and function. They are efficacious: They do what they are supposed to do and appear to the user to do so with ease.”

— Meyer and Lechner, Chapter 4 of The Power of Product Platforms

In his Axiomatic Design: Advances and Applications, Nam-Suh describes the process of design as the mapping from function to form. As a specific, desired function of a system or subsystem is realized, design creates the form which actualizes that function. When this form closely follows the function, products become simpler, operation of the products become more intuitive, and the products just seem to “make sense.” There are many ways form can be mapped to function, and the mapping is determined by the decomposition of both form and function. This mapping ultimately determines the realities of the design.

Great designers adhere to the “Form Follows Function” principle without fail. Dieter Rams, the celebrated chief designer for Braun, lists his fourth principle for good design as “Display the

---

179 For example, Ancona et al; D. Ackoff; and R. Henderson, Lecture to SDM ’97 class, Summer business trip, 1998.
182 Meyer and Lechner, p 83.
183 Crawley, System Architecture class notes, Fall of 1997.
The function of a product; its form follows its function.’ Rams goes on to say that the primary purpose of a good design is to fulfil the product’s primary function, which includes, according to Rams, appealing and communicating to the customer. Rechtin echoes a similar theme in his heuristics: “System structure should follow functional structure” and gives several examples in which a system was successful because of its adherence to this principle.

There are several benefits when the form of a product follows its function. A cleaner mapping makes coupled designs less likely, which in turn makes specification of the design simpler, more straightforward, and more likely to succeed. Secondly, the product clearly communicates its function to the user; the product does what it looks like it is supposed to do. This is another means by which the user can extract greater function from a product. Thirdly, Billington suggests that when form is chosen judiciously, the analysis of it can become “astonishingly simple”, making the design process a simpler, shorter task. Lastly, the function can be used to suggest forms of the solution. Elegant solutions many times have a corresponding structure with the original problem. Rechtin gives several examples in which this was the case. Solving recursive equations, for example, is generally very slow by typical means, but when binary recursive circuits are used which resemble the original structure of the equations, performance is significantly improved. The stream function example given in Section 4 is another excellent example. Isocontours of the stream function trace the paths of the fluid flow; mathematically incorporating the stream function also greatly simplifies an otherwise unruly problem.

---

185 Rechtin, p. 54.
186 Suh, Chapter 1.
187 MacLennan, p. 36.
188 Daniel Jackson, personal communication, October 19, 1998.
189 Rechtin, p. 54.
5.3 The "Wow" Factor

"The viewer who perceives elegance has the same aha! experience that the designer had when solving the original problem... When the viewer grasps the function of a design, all of a sudden it reorganizes itself and makes sense."  
— Robert Leff, MOT graduate  
Sloan School, 1997, MIT

The joy and excitement when recognizing a design or architecture as elegant is unmistakable. The rapture of sudden insight comes from the all-encompassing understanding after an elegant solution has been assimilated. What once seemed complicated and obscure instantly seems to make sense and becomes obvious.

In this vein, developing an elegant solution can be a double-edged sword. Michael Jackson relates the story of a programmer who was thought to have great potential but turned out to be a disappointment to her management. Despite being given what appeared to be difficult problems, her solutions always turned out to be quite simple, and management was less impressed with her than her colleagues who developed excruciatingly intricate solutions only they themselves could understand!

Often times, elegant solutions are revolutionary rather than evolutionary. In typical design practice, the current solution space is typically scanned to find the optimum in trying to relieve design pressures. Products succeed by operating near this optimum or perhaps customizing to fill certain market niches. When a truly elegant design is introduced which offers improved function while meeting all constraints, it is typically outside current design space. The nuclear submarine, which relieved the requirement for resurfacing once an hour for oxygen replenishment, offered notably greater function while operating within the constraints of an underwater vessel. Powering a submarine by nuclear power was clearly outside the preexisting solution space when the Nautilus left port in 1955. Rarely are truly elegant designs based on an existing, widely used platform. This novelty contributes to the feeling of awe and inspiration when an elegant solution is realized.

The S-shaped technology development model does a reasonably good job describing the evolution of an existing product and then its ultimate replacement by a new architecture. Typically, the replacement technology is inferior to the existing product when it is introduced, but, after some development, eventually supersedes and replaces the existing product. With an elegant solution, however, greater function is realized almost immediately. This is part of the cause for the sudden excitement often associated with elegant products. Shown pictorially, the

---

190 Leff, p. 25.
S-shaped technology development model is contrasted with an elegant solution in the figure below:

It should be obvious that the “wow” factor may be culturally and timing dependent. Many elegant solutions, the triode tube amplifier or the Roman Pantheon, for instance, retain the beauty of elegance throughout time and throughout most cultures. Others, however, become so commonplace that their beauty is lost in normality. The telephone or common paper clip might be victims of this neglect. Likewise, when a solution from one culture is introduced into another, the new culture could view the solution as revolutionary and, if it works better than the alternatives, perhaps elegant. In the founding culture, however, the solution is simply the way things have always been done. Engineers are as guilty as the rest of the population in falling into this type of ignorance. Rechtin cites a study of a group of practicing systems engineers who reviewed a list of engineering feats. The following observations were made: “Publicly perceived success seemed to be measured against what had happened before, rather than on an absolute basis. If little had existed before, then an order of magnitude improvement was a dramatic success, even though that improvement in absolute magnitude was much less than what came after.”192

192 Rechtin, p. 301.
5.4 Aesthetics

"Technology's single most important obligation is to get out of the way."\(^{193}\)
David Gelernter, Computer systems professor and author from Yale University

Aesthetics can be defined as "many auditory and visual perceptions that are accompanied by a certain feeling of value."\(^{194}\) Products that are aesthetic will naturally be more attractive to consumers than those that are not. Beyond this, aesthetic solutions are also appealing in some regard in that they appeal to the observer's emotion and intellect in addition to the usual physical aspects.

Clearly, just being aesthetic alone is not sufficient; a product must work and must work well. Steve Jobs is quoted as saying to his Macintosh design team, "Real Artists Ship." But aesthetics is certainly a component of most elegant solutions and an important one at that. Of the seven examples reviewed in Section 4, every one is aesthetically pleasing in some respect, usually in more than one aspect. Of the four social pursuits defined by the ancient Greeks, Ackoff believes aesthetics is the least understood but most critical.\(^{195}\) There is a stark dearth of literature on aesthetics in high technology subjects. So little is known about aesthetics that its discussion is sure to bring controversy.

There are schools of thought that believe aesthetics is far more important that just the advantages in appearance. Billington explains that the accident that occurred with the Tacoma Narrows Bridge, which self-destructed because of an aerodynamic instability, was a consequence of reliance on strictly theoretical analyses. "[T]he very problem that destroyed the Tacoma Narrows bridge had been anticipated and avoided a century before by bridge designers who were guided by aesthetic principles." The best designers "do not rely on mathematical analysis (although they do not abandon it altogether). Rather, their design activity is guided by a sense of elegance."\(^{196}\) Paraphrased, much of Billington's message is: Designs that look good probably are.\(^{197}\)

The attempts at measuring aesthetics are just as interesting as the concept itself. One of the most in depth studies was conducted by Birkhoff in 1933.\(^{198}\) Birkhoff found that aesthetics are related to the tension caused by the neural adjustments evoked during the act of perception.

\(^{193}\) Gelernter p. 50.
\(^{194}\) Flood and Carson, p. 36.
\(^{195}\) Ackoff, p. 39.
\(^{196}\) MacLennan
\(^{197}\) Billington, p. 10.
The more tension, the less aesthetically pleasing. Unfortunately, the psychoanalytical measures of aesthetics have not ventured much beyond this promising start.

### 5.5 Clear Goals

"The statement of scope will have enormous leverage on the success of the system."  
– Ed Crawley, Head of Aero-Astro department at MIT; Professor of Systems Architecture

Regardless of the discipline or the type of the system, all elegant solutions have clear purposes. This is true of all examples mentioned in this thesis. It is perhaps a consequence of the first part of the definition set for elegant designs and architectures: that they work. If there is no objective against which to assess whether or not they work, a system can not be described as elegant.

Beyond these semantics, clear goals and definitions are critical for an elegant system, and more will be discussed about this in Section 8.2.1. Simon portends that the Apollo moon missions were relatively simple because of their clear, undeniable goal. He makes the same argument for the creators of the U.S. Constitution. In each case, the objectives set for the system were clear and, although substantial, limited in scope. Bounded Rationality is the term he has coined for this type of goal setting. For the moon missions, Simon argues that they were difficult in only one regard, their technical aspects. This, he says, was a significant reason for the overall success of the program.

### 5.6 Consistency

"A good design is consistent right down to the details."  
– Dieter Rams, chief designer at Braun 8th Design Principle

A trait repeatedly mentioned about many elegant solutions is that they are consistent. This is particularly important when there is a human interface as part of the system. An elegant design or architecture should also look like it belongs in the environment in which it exists. These two factors of consistency will be considered in the following sections.

---

199 Crawley, E.; System Architecture class notes, Fall 1997.
200 Freeze, p. 12.
5.6.1 Consistency in Function and User Interface

"Conceptual integrity is the most important consideration in system design."

– Frederick P. Brooks, Professor of CS at UNC
and “father of the IBM System/360”

When people experience a product, they learn how it responds and expect it to consistently deliver results consistent with those observations. When a product fails to meet those expectations, the consumer is at first confused and ultimately loses confidence in the product. Products with significant human interaction (e.g., software products) are particularly vulnerable to losing a customer’s faith because of poor consistency. In addition to being entertaining reading, the *Unix Hater’s Handbook* warns aspiring architects and developers of the pitfalls from inconsistency. Chapter 7 deals with X-Windows and Ranum Marcus of the Digital Equipment Corporation is quoted as saying “If the designers of X-Windows built cars, there would be no fewer than five steering wheels hidden about the cockpit, none of which followed the same principles – but you’d be able to shift gears with your car stereo. Useful feature, that.”

Despite being powerful, X-windows fails to deliver consistency and can exasperate its users as a result.

It is imperative that the mental model presented to the user is coherent and consistently followed. This includes strategies for executing applications, user-interfaces to receive and supply data, and responses to the actions of the user. The Apple Desktop is an excellent example of this consistency, and its power is enhanced by ensuring that application writers follow similar guidelines. The Apple Desktop model adheres to the physical desktop metaphor almost unfailingly; some of these traits/objects include:

- overlay (not tiled) windows
- the ability to size and shape windows (which extends the desktop metaphor)
- lets user “put away papers” by minimizing them
- able to move objects by drag and drop
- point with the mouse
- use a trash can to dispose of objects
- document cutting, copying and pasting just as you could do with paper documents

The consistency is so pervasive that even a small deviation from the physical model is perceptible and even disdainful. Dragging the floppy disk to the trash can to eject the disk, for example, seems completely non-intuitive and even offensive in such a beautiful operating system. Visual ambiguity can be extremely detrimental to a product’s success; this is particularly true in user interfaces for computer systems.

---

201 *Unix Hater’s Handbook* #9, Chapter 7.
202 Brooks, p. 255.
Brooks teaches that the surest way to consistency of the interface is through "conceptual integrity." Conceptual integrity, he explains, is what results when a system is the product of one mind, and that it is conceptual integrity that determines the ease of use of the system. Through this one person, all interfaces with the user are managed so that consistency is guaranteed. A superb example of this is the Macintosh computer. While the majority of the Apple Company was developing and refining The Lisa, ultimately a huge flop for the company, less than a dozen people and a central architect came up with the Mac, an unequivocal success.

Products other than computer systems also benefit from consistency in interfaces. The Braun school of design teaches that interfaces with the customer must be "honest." By being honest, an interface will always produce the expected results from a user's action. This design philosophy is maintained throughout all their consumer products which have received many design awards for their simplicity, character, and honesty of design.

5.6.2 Consistency with Environment

"Nature is the main informant...."

— Steve Imrich, Building Architect

Elements just outside the boundary of a system will have the greatest impact on the system. This boundary might be the literal environment, as the void of space acts on a space craft, or it could be the more generic environment just outside the system boundary. In any case, the system must act consistent with its environment and be comfortable in it. System architecting is a combination of art and science, but Steve Imrich warns that it is more than just a purely creative process. The architect must consider where his or her system will exist and operate, the context of the system. A design must fit with its space and its surroundings.

An example can again be found in computer systems. The Unix operating system treats I/O devices as special files which can be read from or written to just like ordinary disk files. From this architectural decision, they gain significant leverage. Specifically, "file and device I/O are as similar as possible; file and device names have the same syntax and meaning, so that a

---

203 Brooks, p. 35, 44, 46.
204 Freeze, p. 12.
205 Guest speaker Steve Imrich, accomplished architect, speaking to the System Architecture Class, SDM97, Fall 1997.
207 Rechtin, p. 6.
208 Imrich, S.
program expecting a file name as a parameter can be passed a device name; finally, special files are subject to the same protection mechanism as regular files.210 In this way, Unix interacts with its external environment, in this case its I/O devices, in a manner consistent with its internal devices, its files.

Java programming language is based upon a simple but elegant premise and is another example of consistency with environment. Instead of writing programs for some particular computer, a programmer writes them for the Java virtual machine that every computer can become. In this way, the program is always consistent with its environment.211

Simon warns that complexity in the external environment creates complexity in the internal environment.212 In many instances, it is necessary for designers to try to “insulate” their products from the external environment, or, in other words, to make the system performance independent of the external environment. John Harrison’s chronometers, for example, were required to maintain their precision despite wild fluctuations in temperature, humidity, and the rolling of the ship’s deck. All of these factors in the external environment had to be considered before a successful product could be developed. In a similar fashion, the 707 had to cope with its external environment as it approached the speed of sound at high altitudes and, at the same time, had to consider the environment to which it exposed its passengers. These considerations had huge implications on the 707 architecture.

5.7 Capable of Redefining or Creating a Business

As discussed in the previous sections, an elegant solution provides the same or improved function while relieving design pressures that prior to the solution limited the system’s performance. In many systems, the elegant solution is simpler than the ones it replaced. Contrary to typical, S-shaped technology replacement, the elegant solution can leap-frog existing systems in performance. Because of these advantages, elegant designs and architectures often redefine the industry or discipline in which they operate. This aspect of elegant solutions is more of a consequence rather than a characteristic, but it is prevalent enough to warrant inclusion here and is motivation to continue the search for elegant designs and architectures.

209 Mullet and Sano, p. 51.
211 Gelernter p. 46.
212 Simon, p. 34.
Conductive carrier in electrophotography had such an impact on its industry when it was introduced in 1975. In a similar way, the 707 did not create the industry of public air-transport, but international air travel doubled in a short time after the 707 began service. In some cases, the elegant solution is such an innovation that it creates an entirely new industry or field of study. Although the PDA concept was not new when the Palm Pilot was introduced, this device redefined and, in essence, created a new market for PDA's. When Xerography was introduced, it completely displaced traditional silver halide techniques and created the multibillion dollar market for in-office document duplication. Many more examples can be constructed along these same lines.

6. The “Relativity” of Elegance

Even beyond the differences in individuals' styles, the concept of elegance is dependent upon other factors. If two people are looking at the same system but have different objectives in mind, they will almost certainly evaluate the utility of the system differently. Complexity, as has been defined above, is sensitive to the efficiency of the language in which the schema is being described. The concept of imaginary numbers, for example, is unfathomable to most grade-schoolers but pretty simple for a college engineering student. This difference in apparent complexity results in yet another reason what is elegant to one individual may not appear so to someone else. Lastly, timing and culture play an important factor in what we consider as ordinary or extraordinary so they, too, impact our relative definitions of elegance. These three factors will be considered in more detail in the next three sections.

6.1 Variations in Objectives

The two proofs of the Pythagorean Theorem given as elegant examples demonstrate how different objectives might impact what an individual refers to as elegant. If the intent of the proof is to make the Pythagorean Theorem as obvious and as intuitive to as many people as possible, the first example serves this purpose well, as do the many Java applets and some of the other, simple 2-d proofs. If, on the other hand, the intent is to create a powerful mathematical statement that works in any number of dimensions, the second proof using linear algebra arguments is clearly the winner.

This is also true for products as well. In software development, for example, there are many costs which make up the total development costs: design, execution (writing of the code), testing, user learning curve, system maintenance, etc. Depending upon the objectives set for
the product, minimizing different costs will assuredly result in different system architectures. How well the product meets these objectives will in turn determine its elegance (first part of the definition of elegance).

The F-16 fighter was originally required to be a Mach 2+ fighter. The actual F-16 would fail this objective miserably but actually is an excellent answer to the real, unstated need: a fighter which can quickly exit a dog-fight. The plane that was designed only attains a top speed of Mach 1.4 but, because of its high thrust-to-weight ratio, adequately meets the needs of the Department of Defense.213

6.2 Variations in Language

"The brain is an extremely complicated entity to a neurobiologist but, to a butcher, just another meat."214

– W. R. Ashby, expert in computer systems

Since simplicity is such a common design pressure on systems, how one evaluates a system’s complexity will determine how elegant that system appears in relation to its alternatives. As described in Section 5.1.1, effective complexity is a useful measure of a system’s complexity and is defined as the length of the description of its regularities. The language one uses to describe the schema, therefore, will determine the effective complexity of the system. It is helpful in this case to define ‘language’ as Mullet and Sano do in Designing Visual Interfaces. They describe a language as 1) defining a universal set of possible signs and symbols and 2) providing a set of rules for using those signs and symbols.215 It is imperative, therefore, that “the descriptive language … be previously agreed upon and not include special terms made up for the purpose.”216 Complexity is relative to the knowledge, notations, and methods at your disposal.217

Beyond the issues of semantics, language can determine the ability to grasp the elegance of a design or architecture. Only by understanding a discipline can one fully appreciate an elegant solution developed within it. Philip Johnson described it eloquently: “Whoever understands the dynamics of pitch in propeller blades or the distribution of forces in a ball bearing so that he can participate imaginatively in the action of mechanical function is likely to find this knowledge enhances the beauty of [those] objects.”218 This is clearly a factor in science.

213 Crawley, E.; System Architecture Class Notes, Fall 1997.
215 Mullet and Sano, p. 2.
216 Gell-Mann, p. 33.
217 Jackson, M; p. 154.
218 Gelernter, p. 8.
Consider the solution for stokes flow around a sphere given in Section 4.4. Someone not familiar with vector operators would have a great deal of difficulty following such a solution. Likewise, someone more familiar with vector calculus might prefer the solution presented by Landau and Lifshitz\(^{219}\) over the one presented in this thesis. These preferences result from differences in “languages.” Language and differences in language contribute to our different perceptions of elegant solutions. Improving one’s ‘vocabulary’ by increasing knowledge and understanding in multiple disciplines is a powerful tool in developing one’s appreciation of elegance and in creating elegant solutions.

6.3 Variations in Timing and Culture

“While math and science can be universally elegant, design is always viewed in the context of its time and culture”\(^{220}\)

—Patrick Brennan, Interior Designer

As described in section 5.3, an system that appears elegant in one culture might appear complete routine in another. In a similar fashion, some elegant solutions might lose some of their wonder as time passes and they become a part of everyday life. The mouse trap provides a good example of this. The typical, old-fashioned mousetrap by Victor is a familiar object to most people. It was patented in 1913 and has enjoyed enormous success since then, selling millions of units over the years.\(^{221}\) At the time, it would have been considered an elegant solution for rodent removal. Recent designs, such as the “The Better Mousetrap” have significantly reduced the number of parts, kinds of parts, the complexity of manufacturing, and have made it easier for a consumer to load, set, and clean the trap. Between the two designs, it is not hard to argue that “The Better Mousetrap” is a more elegant product. A valid question is does this detracts from the elegance of the Victor mousetrap?

As will be discussed in greater detail in Section 8.2, this relativity of elegance can be used by the architect to help generate elegant solutions. For example, in Shinya Iwata’s M.E. thesis, it is shown that home designs can be made more elegant by taking ideas from across cultures. In this case, Japanese kitchens and their efficiencies are incorporated into a new kitchen design for North America.\(^{222}\)

\(^{219}\) Landau and Lifshitz, p 64.
\(^{220}\) Jeff, p 18.
\(^{221}\) Meyer and Lehnerd, p. 84.
7. Gaining a Sense of Elegance

"Aesthetics is notoriously hard to teach..."223

– Bruce MacLennan, Author and Professor of Computer Science, Univ. of Tennessee

A prerequisite to building elegant solutions is the ability to recognize elegance. In many respects, this is the hardest part of creating elegance. The sense that a design or architecture is elegant is difficult to put into words and, as a result, difficult for mentors and teachers to instill in their understudies. A commonly held opinion by many that have considered this question is that considerable knowledge and experience is required to fully appreciate an elegant solution.

The first and foremost recommendation is to study the elegant designs that already exist. Bentley suggests that “elegance can be learned through imitation and practice” and suggests that programmers wishing to acquire a sense of elegance examine the code of the best programmers and study how they developed their code and styles.224 Rechtin similarly suggests an architect compare his or her creation to successful architectures.225 Petroski suggests a history course for engineering students. The nature of design, he argues, is essentially unchanged since its inception. A properly conceived history course would allow students to learn from past engineering successes and failures.226 Gelernter believes great works of the past should also be studied. He suggests that students of engineering concentrate some of their studies on the arts, the freest of pursuits, to develop a sense of beauty.227

Experience is the best teacher and there is no substitute for knowledge. Rechtin, for instance, suggests an architect gain at least ten years of experience before excellent designs and architectures can be expected.228 During development, Rechtin further suggests the budding architect be trained in an entrepreneurial atmosphere with increasing levels of responsibility and be encouraged to be creative.229 Brooks provides similar advice for cultivating designers capable of elegant solutions: 1) identify them early, 2) provide them mentorship, 3) have an identified career development plan, and 4) provide them with stimulating interaction.230 And most importantly of all, design obsessively.231

223 MacLennan, p. 37.
224 Anthes
225 Rechtin, p. 296.
227 Gelernter, p. 130.
228 Rechtin, p. 292.
229 Rechtin, p. 292.
230 Brooks, p. 203.
231 MacLennan, p. 37.
As mentioned previously, improving one's knowledge and understanding in multiple disciplines permits an individual to appreciate the beauty of elegant solutions in those disciplines. By increasing one's 'vocabulary' in this manner, elegance becomes easier to detect and eventually easier to create. This knowledge may come naturally through experience, as mentioned above, but it can also be effectively nurtured through study across disciplines and examinations of elegant examples in those disciplines.

To further cultivate the "elegance sense" of developing designer and architects, additional works such as this thesis should be encouraged to contemplate elegance in system design and architecture and to attempt to distill the common characteristics and true definition of elegance in complex systems. As more and more practitioners consider these questions, the answers can only be improved.

8. Creating Elegant Solutions

"One good question concerns whether most elegant ... designs are intentional or accidental."²³²

– Donald Christiansen, IEEE Spectrum Editor

If elegant designs and architectures arise purely out of inspiration, then the best one can hope to do is improve one's own sense of elegance so that it be detected when or if inspiration should strike. If this is the case, great architects and designers are born, and the rest of us will have to wait for them to fathom the next revolutionary design! It is most prudent to assume elegance can be taught and that proper application of perspiration with occasional inspiration will eventually yield an elegant solution.

To that end, some of the existing literature on creative problem solving is discussed as it relates to the development of elegant, complex systems. First, the more traditional approaches to creativity and innovation are briefly discussed in this light. Secondly, those aspects which seem to be most critical in the development of elegant solutions are given additional attention.

As a catalyst speeds the kinetics of a chemical reaction, the intent of these methods, procedures, and hints are to catalyze the process of creating elegant solutions.

²³² Christiansen
8.1 Traditional Approaches to Architecting, Creativity and Innovation

Nearly every text dealing with complex systems incorporates some survey of problem solving within its context. Complex systems create difficult problem requiring innovative and creative solutions. Through the years, people facing these difficult problems formulated their methodologies for dealing with them. The literature is rich with information on creative problem solving, and no attempt to survey it has been made here. Only how these techniques relate to elegant solutions will be proposed.

Those that teach creative processes implicitly believe that it can be taught, that it is not a magical process occurring simply by chance. That belief is postulated to extend to the creation of elegant systems as well, and, by learning these creative processes, elegant solutions can be more easily created.

Rechtin provides a taxonomy for the creative processes that can adequately frame this discussion. He proposes there are four methods for architecting systems: the normative, rational, argumentative, and heuristics methodologies. Each will be briefly discussed in turn. It should be noted that these methodologies are not mutually exclusive – they can and should be combined when it is appropriate.

8.1.1 Normative (Pronouncement) Methodology

In the normative method of architecting, an architect obtains his inspiration from a recognized expert in the discipline. The expert essentially dictates what constitutes a good design to his or her understudies, and by following these edicts, the disciples can create effective designs and architectures. The instructions laid out by Vitruvius in his ten books of architecture serve as an excellent example. Vitruvius specifies everything from the direction a temple should face to how the walks of a theater should be laid out to how the concrete for its floors and walls should be mixed. Bill Gates of Microsoft might be considered a modern-day maestro of the normative methodology.

One of the key advantages of the normative methods is that excellent results can be gained from entire schools of architects (even the less skilled ones) if the authority figure is very adept.

---

234 Vitruvius, p. 116.
235 Vitruvius, p. 155.
236 Vitruvius, pp. 42-51.
and can compose his process of architecting.\textsuperscript{238} However, because the normative methodology consists mostly of the “Do this, Do that” mandates from a central “guru”, creativity from the individual performing the architecture or design can be suppressed. In general, the dogma is not applicable to new, novel types of systems. Further, the quality of results is limited by the skill of the master.\textsuperscript{239}

Because the results are based upon the lessons learned from a previous architect and/or designer, it is difficult to believe that application of the Normative Methodology could consistently result in elegant solutions. The “out-of-the-box” approaches often required of elegant solutions are unlikely to result from the normative process, unless the guru himself or herself is doing the architecting.

\subsection*{8.1.2 Rational (Procedural) Methodology}

Whereas the normative approach focuses on the end result, the rational methodology focuses on the actual process of architecting.\textsuperscript{240} By following a thorough and effective process, it is assumed that good results will follow. The rational approach arose due to the limitations of the normative methodology in dealing with novel problems or very specific design issues not covered in the original dogma of the expert. The methods, and there are many, are more general than the normative methodology and also better capable of handling complexity. The processes are based on rational thought and rely upon the logical or mathematical structure of problems to ultimately result in a solution.\textsuperscript{241}

Altshullen’s “Theory of Inventive Problem Solving” (TIPS) is a good example of the rational methodology.\textsuperscript{242} Based upon commonalities and patterns found in many Russian patents, TIPS allows a user to locate a suggested technique or type of technology that was effective in relieving the same design conflicts faced by the user as was resolved in the Russian patents. The entire systems engineering (SE) process is an excellent example of the rational approach. By following an SE process, the systems engineer/architect can deal with extremely complex interactions, subsystems, and conflicting requirements to create a successful product. System Dynamics, invented by Jay Forrester around 1963, is yet another example. The tools of SD model transient, complex behavior of many independent but interacting actors; the results

\textsuperscript{237} Crawley, E.; System Architecture class notes, Fall '97.
\textsuperscript{238} Crawley, E.; System Architecture class notes, Fall '97.
\textsuperscript{239} Rechtin, p. 15.
\textsuperscript{240} Rechtin, p. 15.
\textsuperscript{241} Crawley, E.; System Architecture class notes, Fall '97.
\textsuperscript{242} Boppe, C.; Systems Engineering class notes, Summer '97.
suggest innovative ways to relieve conflict by viewing the system structure as the cause of that behavior. 243

Axiomatic Design by Nam Pyo Suh provides another example. Axiomatic Design seeks to create a science-based school of thought on the design process. 244 By specifying design axioms, Suh provides a rational for designers to make decisions during the design processes so that creativity is enhanced, trial-and-error is reduced, and the best design among the alternatives is chosen. 245 An interesting study in and of itself would be worthwhile to consider how the axioms of Axiomatic Design may or may not imbue elegance in system design.

The procedural methods focus on the procedure of creating an effective design; they stress the process and mechanics of design over pure creativity. 246 It must be said, though, that application of the various processes certainly requires creativity on the part of the practitioner. The rational approaches are superb at refining ideas, are more general than the normative approach, and are adaptable to various situations. 247 Elegant systems have and will continue to be contrived by procedural methods, but the lack of emphasis on creativity makes the next two approaches to system architecture and design the most promising for generating elegant solutions.

8.1.3 Argumentative (Noise) Approach

"...the principle obstruction between us and the future we most desire is ourselves." 248
– Russell Ackoff, Prof. of Management, The Wharton School, Univ. of Pennsylvania

Most constraints are self-imposed; this is true in many aspects of human life including complex system development. Those types of constraints have loopholes, but they typically are exceptionally difficult to recognize and appreciate. The argumentative approach to system design and architecture seeks to relieve these self-imposed constraints by introducing different perspectives into the architecting process. The technique hinges upon two separate components: 1) an architect, designer, or team open-minded to alternative solutions, and 2) a method for creating or suggesting alternative solutions. The first is the responsibility of the management of the organization; they must create an environment in which new ideas are valued and different perspectives are desired. It is also the responsibility of the architect or

244 Suh, p. 8.
245 Suh, p. 8.
246 Rechtin, p. 17.
247 Crawley, E.; System Architecture class notes, Fall '97.
248 Ackoff, p. 120.
designer who must eliminate pre-judgment and bias in order to make the familiar unfamiliar and as a result permit better designs and architectures to be considered. The second component is perhaps more interesting from a creative standpoint and will be the focus of the subsequent discussion.

In Section 3.2.1, the "fitness" concept was introduced as one possible way to view utility. Gell-Mann further describes the "fitness landscape" which is graphical way to picture a system's "fitness" within its design space. He makes the concept more intuitive by suggesting the fitness landscape is turned upside down so that lower values are considered "more fit"\textsuperscript{249}. Using this visual, a one-dimensional design space can be depicted on a chart like the following:

From the illustration above, it is easy to see that the current solution is not the best solution available. It is at "equilibrium," though, and unless there is some impetus to drive it out of the local optimum, it will remain there. Gell-Mann suggests that "noise" from the argumentative approach can be that impetus and shake and jostle the solution out of its local optimum\textsuperscript{250}. The noise might result in solution "A" (a less desirable alternative), but, if applied repeatedly, should ultimately produce solution "B". Lloyd warns that care must be taken that the noise not be too great, lest solution "B" be completely missed as the too-large "perturbations" convey the team directly to solution "C"\textsuperscript{251}. This visualization is so effective that I prefer to call this

\textsuperscript{249} Gell-Mann, p. 249.
\textsuperscript{250} Gell-Mann, p. 267.
\textsuperscript{251} Gell-Mann, p. 369.
methodology the "Noise Approach" rather than the "Argumentative Approach", which seems to have a slightly negative connotation.

There are many methods to introduce this "noise", and no attempt will be made here to provide a survey of the various methods. Brainstorming is perhaps the most popular of the methods as a way to introduce alternative perspectives from those who may be completely unfamiliar with the problem. In brainstorming, a critical principle is that no idea is considered implausible or unreasonable; all ideas are encouraged. A completely crazy idea could provoke someone else's thoughts to reshape the idea into something feasible or to invent a completely different solution. Another method recommended by Edward DeBono suggests using the last noun on the front page of today's newspaper to trigger a new wave of thought to a vexing problem.\textsuperscript{252} Similarly, just driving to work by a different route could provoke a new thought from the change in scenery.\textsuperscript{253} Ackoff suggests a form of brainstorming by first listing the desired properties of an idealized design. Such an idealized design is not constrained by money or development time or even technical feasibility. After generating idealized design, constraints are applied to develop a feasible system. This method has the added advantage that participants need not be experts to suggest ideal properties of the system.\textsuperscript{254} Unfortunately, Ackoff gives only rough clues on how to make the idealized design realizable. Synetics, search conferencing, and TKJ are other ways to promote creative thinking.\textsuperscript{255}

Aside from possibly generating the optimum solution, the noise approach can help populate a wider spread of the design space with possible solutions. As the solution space is expanded, it is more likely that the optimum rests within that space and can be generated by combinations, permutations, and hybrids of the existing solutions.\textsuperscript{256} As William Gordon suggests, the first step in innovation is to view a familiar problem in the context of strange, new concepts.\textsuperscript{257}

\textsuperscript{252} Gell-Mann, p. 267.
\textsuperscript{253} Chris Gangai, personal communication, January 6, 1999.
\textsuperscript{254} Ackoff, pp. 110 -111.
\textsuperscript{255} Ackoff, p. 178.
\textsuperscript{256} Billoy, S.; Principles of System Architecting, Fall 1997.
\textsuperscript{257} Leff, p. 30.
8.1.4 Heuristics

"Recurrent types of problem in design lead to elegant solutions in which the operation of consistent design principles may often be traced."\textsuperscript{258}  

– M. J. French, Prof. of Engineering, Lancaster University

Rechtin describes heuristics as generally accepted, qualitative statements that aid in decision making and problem solving.\textsuperscript{259} By-in-large, they are generalizations from the collective experience of architects who themselves have practiced the development of complex systems. In order to qualify as a heuristic, it must have several properties: \textsuperscript{260}

- the opposite statement of the heuristic is non-sensible or leads to certain failure;
- they must be applicable beyond the original context; and
- they must be able to be stated in a few sentences or short paragraph of text.

They may be either prescriptive or descriptive. The strengths of the heuristic approach are that it can bring method to a chaotic and ill-defined quandary. Heuristics can also stimulate ideas which can later be refined with the other methodologies, the procedural methods, for example.\textsuperscript{261} French believes that the most promising means of developing a discipline of design is the statement and study of principles, the development of new heuristics, and the refinement of existing ones.\textsuperscript{262}

There are several citable examples of heuristics. In "An Annotated List of Design Principles," French takes time to document forty-four design principles he has observed during his career in design, ranging from design specifications and constraints to thermodynamics.\textsuperscript{263} Rechtin's list is likely the most famous of heuristics and is mandatory reading for an aspiring system architect.\textsuperscript{264} In computer systems, Lampson from the Xerox Palo Alto Research Center has documented "Hints for Computer System Design."\textsuperscript{265} In this text, he gives helpful advice on many subjects including the definition of interfaces, the importance of continuity, the role of abstractions and hierarchy, and others. Brian Lim's SDM Thesis is a collection of system architecting principles distilled from the suggestions of the 1997 System Design and

\textsuperscript{259} Rechtin, p. 19.
\textsuperscript{260} Rechtin, p. 18.
\textsuperscript{261} Crawford, System Architecture class notes, Fall 1997.
\textsuperscript{262} French
\textsuperscript{263} French
\textsuperscript{264} Rechtin, pp. 311-319.
\textsuperscript{265} Lampson.
Management class. More documents such as these are needed so that architects might have similar and improved heuristics over a wider array of subject material.

The heuristic methodology is a powerful technique which can inspire creative solutions to well-defined problems. Computer systems supplied with a fairly comprehensive list of heuristics on a limited subject can produce surprising results. The AM system by D. Lenat, for example, is a computer that can solve problems without explicit goals. The system is provided:

1. criteria for determining what concepts are interesting (e.g., if the concept is related to other interesting concepts or if examples can be found but not too easily);
2. a set of heuristics, (e.g., after finding a concept, construct examples of it and generalize or specialize the heuristic depending upon how many examples can be found); and
3. elementary knowledge of task domain.

For example, given set theory, the AM system found the concept of natural numbers, addition, subtraction, multiplication and division. After discovering division, prime numbers came quickly and from that Goldbach's conjecture that any even number can be expressed as the sum of two primes. A somewhat similar program by P. Langley called Bacon performs similar functions in physics; it has found Kepler's law of planetary motion, Snell's law, Ohm's law, and other physical relationships.

To further illustrate the concept of heuristics, a few examples of heuristics for elegant solutions are provided below. There are many more, and, in retrospect, identification of heuristics leading to elegant solutions is probably a more noble and useful subject for a thesis than those covered here.

Example 1: Good Designs Look Good
David Billington makes this point clearly in his book, The Tower and the Bridge, and its significance was discussed in Section 5.4, Aesthetics.

Example 2: Stable Intermediate Configurations can Improve the Speed of Development
As stated in the introduction to Section 8, elegant solutions are difficult to synthesize and the thrust of this entire section has been to seek out catalysts for the development of elegant systems and designs. Herbert Simon, in his book The Sciences of the Artificial, makes a case that, if stable intermediates are constructed on the way toward elegant designs, the speed of development and the quality of the potential solutions are greatly improved. For an example, he gives the evolution of *homo sapiens* and cites our animal predecessors as the obvious stable intermediates.

---

267 Simon, p. 124.
268 Simon, p. 125.
269 Simon, p. 207.
Example 3: Capitalize on randomness

Despite the lack of discussion in the references studied for this thesis, randomness continuously arose as a valuable tool in many elegant solutions. I provide the following examples in defense of this heuristic:

- **Ethernet Collision Avoidance**: When a collision between packets is detected on the Ethernet, the client uses statistical arbitration to determine the next time for re-transmission. In this way, control is maintained on the local machine, greatly simplifying control of the entire network, and the possibility of another collision is minimized.\(^{270}\)

- **Monte-Carlo Simulation**: Given many initial conditions and the task of calculating the probable output of a system faced with these initial conditions, straightforward calculation of the result through a system model can be prohibitive if there are many conditions to be considered. Using the Monte Carlo simulation, some of these initial conditions are picked at random and, if necessary, the result is scaled appropriately. Gell-Mann cites using a Monte Carlo Simulation to add a very large set of numbers; picking a few of the numbers at random and scaling their sum gave a good approximation for the actual sum.\(^{271}\)

- **Combinatorial Optimization Problems**: In his Ph.D. thesis, Suresh Chari considers how randomness has become an integral part of theoretical computer science. “Randomization leads to solutions which are simple, elegant and more efficient than deterministic algorithms.” Randomness is so valued that random bits are treated as a resource.\(^{272}\)

- **Global Positioning System**: To reduce the required intensity of satellite transmission while still equipping GPS receivers with relatively small antennae, pseudo random code was used as the transmission. With sufficient averaging, an amazingly weak signal can give distance to the satellite with remarkable precision.\(^{273}\)

- **Random Reductions in Computer Science**: Rohatgi investigates the power and limits of randomness as it relates to random reductions, a recurring theme in randomized computation. “Randomness is widely accepted as a powerful computational resource because the most elegant and efficient solutions to several computational problems are randomized.” Rohatgi states that reducing harder problems to simpler ones is possible through these random reduction techniques.\(^{274}\)

Like the normative method, heuristics permit a system architect or designer to benefit from the experience of those who have faced complex problems in the past. It can also suffer the same disadvantage, however, in that a truly unique solution may not be realizable by simply following the advice of previous architects. Great care must be taken in the development and employment of heuristics to prevent them from limiting the solution space available to the engineer or architect. Furthermore, heuristics are purposefully general so that they can apply to a myriad of design problems. This can frequently result in vagueness, making the heuristic non-prescriptive or difficult to apply. This reaffirms that there is no substitute for studying the actual design problems and elegant solutions of the past first-hand. In that vein, the following

---

\(^{270}\) Metcalfe and Boggs, p. 397.

\(^{271}\) Gell-Mann, p. 45.


\(^{273}\) http://www.trimble.com/gps/howgps/gpsfram1.htm

section points out some features that appeared to be critical in the development of some of the
elegant solutions discussed in this text.

8.2 Critical Features in many Elegant Solutions

The methodologies reviewed above have all been demonstrated to produce workable systems.
Some of those systems would be considered elegant by the criteria put forth in this thesis.
There are some methodologies or heuristics which are so common to elegant products or
solutions that they warrant a more in depth discussion.

Many of these principles are covered within the four methodology-taxonomy coined by
Rechtin. For example, solving the right problem is a recurring theme in many of lists of
heuristics available (e.g., Rechtin, p. 19).

Beyond those listed below, it is perhaps redundant to remind the reader of the characteristics of
elegant solutions discussed in Section 5 and that those characteristics should be pursued if
elegance is desired. Constructing those characteristics into heuristics would be a fairly
straightforward exercise, for example:

- Seek simple solutions.
- Construct potential solutions that have a corresponding structure to the problem statement.
- Pursue consistency throughout the system, particularly in those interfaces with the user.

Once again, it is important to remember that there is no substitute for studying and learning
from elegant examples taken from a wide array of disciplines, businesses, and cultures.

8.2.1 Solve the real problem

"The really creative act in science involves initially formulating the correct questions rather than
providing the answers."

— Albert Einstein

The two guiding principles during the creation of the Palm Pilot resulted in simply redefining
(correctly defining) the need the product was to meet. This single step, along with some
innovative technology to realize those goals, was the critical factor in the Palm Pilot's success.
In the conductive carrier, rather than introduce an additional component into the toner
mixture or complicate the development station with more equipment to increase the electrical
field between the biased electrode and the photoconductor, Kasper and May eliminated the
problem that was reducing the field strength in the first place. By simply allowing the carrier
particles to bleed charge into the bias electrode, the problem was solved. Solving the correct
problem is the first phase in the development of nearly any elegant solution. As stated in
Section 5.5, nearly all elegant designs and architectures have clear goals; this is a result of correctly stating the need(s) to be filled by the solution. This is, of course, not unique to elegant solutions, and every competent list of heuristics has some form of Rechtin’s: “Don’t assume that the original statement of the problem is necessarily the best, or even right, one.”

Many times, application of this heuristic results in the problem having a more limited scope. The Ethernet provides an example of a solution in which the designers wisely limited the scope of what they intended to do. The result was an incredibly reliable communication medium for remarkably little complexity. It is not, however, completely robust, and higher level protocols are necessary to guarantee delivery of packets. This architectural decision allowed application designers to specify their desired level of reliability while providing all users a simple, robust communication medium. The example given as an inelegant proof of the Pythagorean theorem (Example #3) provides a different type of example. In its circuitous route to the solution, it introduced a wide array of inferences that, in the end, hardly seemed germane to the problem. Elegant solutions do not append superfluous baggage. This may not be a popular stance, but it must be enforced early in the development process. “Try to cut bells and whistles at the conceptual stage of a design. At first, certain features may seem useful and convenient (or good for marketing), but they often turn into a nightmare when producing, testing, or using a product.”

Contrarily, proper application of this heuristic may increase the scope of the problem. Ed Crawley suggests the K2 bike shock as an example in which the original problem statement was over-constrained. The original problem statement called for a piezo-electric “washer” to control the orifice in a viscous damper piston head. A better solution, however, was to build a bypass around the cylinder and use a piezo-electric valve.

Representation plays a strong role in forming the correct problem statement. It may be as simple as changing the frame of reference. This is done many times in problems of fluid mechanics; the results for a sphere falling through a viscous fluid are exactly as the same as the results shown in Section 4.4 for a fluid moving around a stationary sphere. A new representation may entail a completely different perspective. Isomorphism is a useful concept for taking a difficult problem and making it something much easier or more familiar. An example is the use of the magic square in number scrabble (a game which entails the task of collecting three numbers which add to 15 before your opponent) which transforms the

---

275 Rechtin, p. 54.
276 Metcalfe and Boggs, p. 398.
277 Joffe, p. 121.
278 Simon, p. 152.
complex task into a simple game of tic-tac-toe. Another example in which representation plays a key role is arithmetic: math with Arabic numerals is seemingly much more efficient than with Roman numerals. Yet another example is integration facilitated by complex numbers. The integral

$$\int e^{-x} \sin x \, dx$$

is relatively time consuming, requiring integration by parts. Representing $\sin(x)$ as the sum of two complex exponentials makes the integration trivial.

Because there is often a correspondence between the structure of an elegance solution and the original problem statement (see Section 5.2), solving the right problem is critical. One last example, the design of devices, called prehensors, for upper limb amputees that replace some of the function of hands, is used to cement this point. Previous prehensor designs created a fixed gripping force. This was an inconvenience to the user since a wide range of forces are desired when handling different objects. The simplest designs merely increased the number of rubber bands to increase the force during closing. Some improvements, therefore, looked at ways to facilitate removal or addition of a few rubber bands and most efforts were focused on this aspect. In the elegant solution, however, it was realized that, by changing the location of the fixed end of the elastomer, the normal force created on the rotating member could be easily adjustable across a wide range of forces. The result gave the widest range of closing forces available for any prehensor on the market and a force fairly constant with respect to opening angle.279

### 8.2.2 An Interdisciplinary, Systemic, and Holistic View

"To be an effective systems scientist we must at the same time be both a holist, looking at the system as a whole, and a reductionist, understanding the system in more detailed forms."280

-- R. Flood and E. Carson, in Chapter 1 of Dealing with Complexity

Complicated systems almost always span many disciplines. Complicated systems also, by definition, demonstrate complex behavior which is greater than the combined behavior of its subsystems. A study of a system’s components and subsystems does not reveal how the system-at-large will react or perform. These facts dictate that complex systems be considered across disciplines and from a holistic viewpoint. Failure to do so not only precludes elegance, but will also probably result in failure. Gell-Mann is emphatic about this point: “Today the network of relationships linking the human race to itself and to the rest of the biosphere is so complex that

---

279 Frey, Dan; personal communication September 25, 1998 and http://design.mit.edu/ddfrey/vector.htm
280 Flood and Carson, p. 17
all aspects affect all others to an extraordinary degree. Someone should be studying the whole system, however crudely that has to be done, because no gluing together of partial studies of a complex nonlinear system can give a good idea of the behavior of the whole.281

The system sciences arose in part due to WWII. Scientists were forced out of their laboratories (micro-environments designed to keep out the real environment) into the real world. When they arrived, they realized the problems there were very difficult and existing methods and techniques were insufficient to deal with the complexity. This led to the formation of interdisciplinary sciences, starting with Operations Research in the late 1930’s out of the British Military.282 The system sciences have expanded to include nearly every aspect of commerce and technology and their successes are well known.

The 707 serves as an example in which the entire system had to be co-optimized. Merely supplying the propulsion sufficient to accelerate a plane to near Mach 1 was insufficient: aerodynamics turned out to be the limiting factor. Once the fix to the poor L/D ratio was found, location of the engines and how they interacted with this new shape became a problem. When the cargo was to be people, another slew of interactions were thrown into the complex stew. By considering all of these factors (and more), an elegant design was found that met all the requirements of the aircraft. Without a systems approach, a success would have likely been the result of good luck rather than good engineering.

The submarine thermal nuclear reactor is an example of an interdisciplinary effort common to many complex systems. The work was split among many and varied disciplines: 30.8% metallurgical engineers, 28.1% ME, 11.3% EE, 11.2% physicists, 6.9% ChE and chemists, and 5.7% other.283

Slicing an orange vertically gives a very different view than slicing it horizontally, but both views are necessary to understand the orange as a whole.284 In a similar way, interdisciplinary, system views of a complex system are necessary to understand the system. Elegant solutions encompass the whole system.

281 Gell-Mann, p. xi.
283 Brennan, p. 32.
284 Ackoff, p. 20.
8.2.3 Looking to Other Disciplines, Times, Cultures, and the Arts

"The architect should be equipped with knowledge of many branches of study and varied kinds of learning, for it is by his judgment that all work done by the other arts is put to test... let him be educated, skillful with the pencil, instructed in geometry, know much history, have followed the philosophers with attention, understand music, have some knowledge of medicine, know the opinions of jurists, and be acquainted with astronomy and the theory of the heavens."285

— Vitruvius, Roman architect, on training of the architect

In the previous section it was demonstrated that complex systems often entail interdisciplinary efforts. The concept of conceptual integrity, as explained by Brooks, is also compelling. For an architect to maintain conceptual integrity, therefore, it is necessary, as Vitruvius observed over two millennia ago, that the architect be skilled in many disciplines.

Flashes of intuition result from acts of recognition.286 When an architect is familiar with a greater number of disciplines, recognition will come far more readily than to an architect with deep knowledge in one discipline but narrow breadth of understanding. When faced with the difficult problem of how to make a sensitive part in a coating-fluid-application easy to manufacture but precise and repeatable, a co-worker found the solution when he disassembled the carburetor to his lawnmower. Jeff Hawkin's fascination with cognition and neurobiology led to his invention of the Palm Pilot. In his book, Principles of Programming Languages: Design, Evaluation, and Implementation, MacLennan borrows extensively from the teaching methods in architecture, aircraft design and other disciplines. "These analogies can be very informative, and can serve as "intuition pumps" to enhance our creativity."287 He found that many of the ideas and insights transferred directly to the subject of designing programming languages. In this manner, new technologies can benefit from the wisdom of more mature disciplines.

Not only should other technical disciplines be consulted for inspiration, completely unrelated studies can provide the same. The arts and history are two areas that are particularly inspiring for many technologists. Einstein, for instance, could be heard playing Bach on his violin on occasion. Simon relates creating complex designs to painting in oil, and admonishes "technocentrics" to explore the arts.288 The inspiration for the Razor Pilot pen mentioned in the Introduction was inspired by the porous lotus root.289 Gutenberg's invention of the printing

285 Vitruvius, p. 5.
286 Simon, p. 105.
287 MacLennan
288 Simon, p. 187.
289 O'Connor
press was allegedly a combination of a wine press and a letter seal, and was inspired at a wine festival.290

In order to gain an appreciation for the arts, they should become part of the standard engineering curriculum. The arts and humanities taught in engineering today by most schools are not sufficient to create well rounded engineers capable of appreciating these subjects, let alone capable of applying them in developing complex systems. Gelernter suggests we “start teaching Velazquez, Degas, and Matisse to our young technologists now... Even technologists ought to study drawing design and art history.” Sim291on suggests greater interaction between engineers and artisans: “They [engineers and composers] could carry on a conversation about design, begin to perceive the common creative activity in which they are both engaged, and begin to share their experiences of the creative, professional design process.”292

Lastly, other cultures and history should be consulted for sources of good ideas. Since the “wow” factor has been shown to be time and culturally dependent, looking to other cultures and times can be a source of elegant inspiration. Bad ideas in the past can become good ideas in the present simply from the advancement of other technologies or a change in social acceptance. Likewise, normal, run-of-the-day ideas or solutions in other cultures could be break-through ideas in the architect’s culture.

There are many good solutions out there; an architect need only step out of his or her comfort zone to find them. This entails looking to other technical disciplines, the arts and history, as well as other cultures.

8.2.4 Tenacity

“Perfection must be reached by degrees; she requires the slow hand of time.”

— Voltaire

Elegant ideas do not necessarily come easily. They do come to those that persist. David Perkins, of the Harvard Graduate School of Education, has identified research indicating people who succeed in difficult problems consistently have several characteristics. They include, among others, a dedication to the task. This should come as no surprise to those who have faced difficult, complex problems.

290 Leff, p. 44.
291 Gelernter, p. 129.
292 Simon, p. 158.
That elegant solutions take time and tenacity can easily be shown by way of example. Chester Carlson labored for ten years before his photocopying machine became workable and eighteen years before it was able to be commercialized.\textsuperscript{293} The incredibly aggressive goal of laying the first transatlantic cable in 1857 required numerous elegant solutions to many problems, but the effort still met innumerable difficulties. In the summer of 1857 alone, the cable broke on six different occasions (once at a depth of 2000 fathoms and costing 300 miles worth of lost cable), the lines went dead repeatedly, unprecedented storms ravaged the deployment ships, and twice the ships had to return to England to get more cable.\textsuperscript{294} John Harrison was finally granted the Longitude prize in 1773 by King George III after 40 years of struggling with the problem of determining a ship’s longitude while at sea.\textsuperscript{295} There are a great many more examples in which elegance came about by more perspiration than inspiration than the reverse.

This difficulty makes elegance rare. Unfortunately, it also makes it unexpected and beyond the goals of most programs. In academia, for example, students are constantly permitted to cease when they have a working model.\textsuperscript{296} They are not encouraged nor given time to make their solution elegant. It’s easy to tell the difference between the work of a carpenter that habitually strives for perfection and one that just makes sure the structure doesn’t fall down. Eventually, the latter would not be able to make a beautiful house if he wanted.

\subsection*{8.2.5 Make mistakes!! It\textquotesingle s Human!!}

By now, it should come as no surprise that elegant solutions, though rare and difficult to create, are well worth the effort. In the process of creating elegant solutions, it stands to reason that there will be many detours and mistakes made along the way. Sometimes, these mistakes are the source of elegance.

Post-It\textsuperscript{TM} Notes are a popular, present-day example. There are many examples of elegant solutions that were at first mistakes:

- Palm Computing Inc.’s first product, Z Omar was a complete failure. But from that failure, the company learned what was critical for a good PDA and what was important to the customer. The result was a superior product that transformed the industry.\textsuperscript{297}
- In the search for the Longitude prize, most “knowledgeable” scientists believed the answer lie in a celestial clock, the most promising of which were the eclipses of the Galilean moons

\begin{footnotesize}
\textsuperscript{293} Schein, pp. 3-7.
\textsuperscript{294} Brennan, p. 87.
\textsuperscript{295} Sobel, p. 9.
\textsuperscript{296} Gelernter, p.130.
\textsuperscript{297} Dillon
\end{footnotesize}
of Jupiter. Unfortunately, even these “clocks” seemed to vary, the eclipses coming early during some parts of the year and late during others. Instead of ignoring these small errors, Roemer correctly concluded that the errors were due to the finite speed of light travel between Jupiter and the earth and the variable distance between the two depending upon the respective orbits. With this knowledge, Roemer calculated the speed of light to within a small error, a significant feat for the state of instrumentation of his day.298

- By a slip of the tongue during a lecture, Gell-Mann correctly identified the isotopic spin of a set of particles by mistake. Immediately following the slip, he stopped and knew instantly that imposing this new spin would be consistent with all other observations and must be the correct spin.299

These stories do not minimize the “incubation” time or effort required for these difficult problems. In fact, they may be an important part of the process. Gell-Mann proposes creative ideas are birthed in a three step process: 1) working for very long periods on the problem; 2) taking a break from the problem; and 3) the solution comes when the mind was considering other things.300

Given the likelihood of making mistakes, it is important that organizations and reward systems be structured to encourage risk taking and look upon mistakes as opportunities to learn. The “Noise” approach discussed in Section 8.1.3 by nature will suggest many unfeasible solutions. Following the processes, procedures, and heuristics discussed in this section will clearly result in some innovative but unrestrained concepts. An organization that is not afraid to embrace and fairly evaluate them will be rewarded in the end. The Boeing corporation, for example, self-financed the entire development of the 707. Even after development, the company tooled for manufacturing capacity, at great expense, despite not having a single order from a commercial airline.301 How many companies today would be willing to take that same risk?

9. Conclusions

Although truly elegant solutions are somewhat rare in complex systems, enough examples exist to suggest the importance and relevance of elegance in system architecture and design. Based upon the advantages imparted to all stakeholders, elegance is clearly a desirable trait in complex systems. Although elegance is not a guarantee of commercial success, it is a strong impetus in that direction. Elegant solutions are unique in that they simultaneously relieve the design pressures that restrained the optimum performance of the system. Whereas traditional engineering solutions seek to find the optimum solution by making intelligent tradeoffs within the design space, an elegant solution forces no such compromise, yet meets the requirements set for it.

298 Sobel, p. 29.
299 Gell-Mann, p. 263.
300 Gell-Mann, p. 263.
301 Http://www.boeing.com/companyoffices/history/boeing/dash80.htm!
Elegant solutions share many characteristics which, although they may not be universal, are remarkably pervasive:

- Simplicity is a resounding theme throughout elegant designs and architectures. Since complexity is a design pressure for nearly every realizable system, the fact that elegant solutions are simple comes as no surprise.
- In elegant solutions there is often a corresponding structure between the problem statement and the solution such that the form of the solution follows the function desired of the design.
- Elegance can induce an emotional response in the observer who is able to appreciate the elegance of the system. Nearly instant awareness and understanding are accompanied by the recognition of an apparently obvious solution, despite the fact that elegant solutions are typically outside the usual design space.
- Aesthetics play an important role in elegance and could be at least partly responsible for the undeniable attraction of elegant solutions.
- Nearly all elegant systems benefited from clear goals at the inception.
- Elegant systems are consistent with their model representations, in their interfaces (especially to the user), and to their environment.
- Finally, the advantages imparted by an elegant solution are often so pervasive that, as a consequence, a business can be redefined or even created by the introduction of an elegant system.

Intrinsically subjective, elegance as defined even by the criteria given in this thesis can not be absolute. Elegance is relative to the objectives of the viewer. It is also dependent upon the "language" or depth of knowledge the observer has with the subject and can be dependent upon the culture and time in history of the observer.

Gaining a sense of elegance is a critical requirement if elegant solutions are to be developed. Some of this sense comes from experience, but it can also be learned by studying the elegant solutions of the past and the processes that created them. There has been much written on the creative process and much or possibly all of that is applicable in trying to create elegant solutions. Beyond the traditional approaches to creativity, elegant solutions consistently result from good, accurate problem statements. It is critical that the architect be familiar with the many disciplines the system will span and that a holistic view is maintained during the development of the system. Elegant solutions often seem to result from inspiration injected from other disciplines and even history and the arts. Architects and designers wishing to create elegant solutions would be wise to reacquaint themselves with great works of art, literature, and the history of technologies and cultures.

It is suggested that additional works considering some of the same questions touched on here be encouraged. These works should contemplate elegance in system design and architecture and continue to distill the common characteristics of elegant solutions to uncover the true
definition of elegance in complex systems. Heuristics capable of yielding elegant designs and architectures should be articulated. Answers to the questions surrounding elegance will only improve as researchers and practitioners ponder them in greater detail.
10. Bibliography


Crawley, E.; System Architecture class notes, Fall 1997.


Eves H., Great Moments in Mathematics before 1650, MAA, 1983.


Gardner, K.; “Less is More, nu?”, Trustee, 50(2), February 1997


Jackson, “A Plea for Leaner Software”, Computer, 28(2), February 1995, pp. 64-68


Kennedy, R.; “Elegant Kludge”, Byte, 20(8), August 1995, pp. 54-60


Lang, Serge; Introduction to Linear Algebra, Springer-Verlag, New York, 1986, 2nd Ed.


