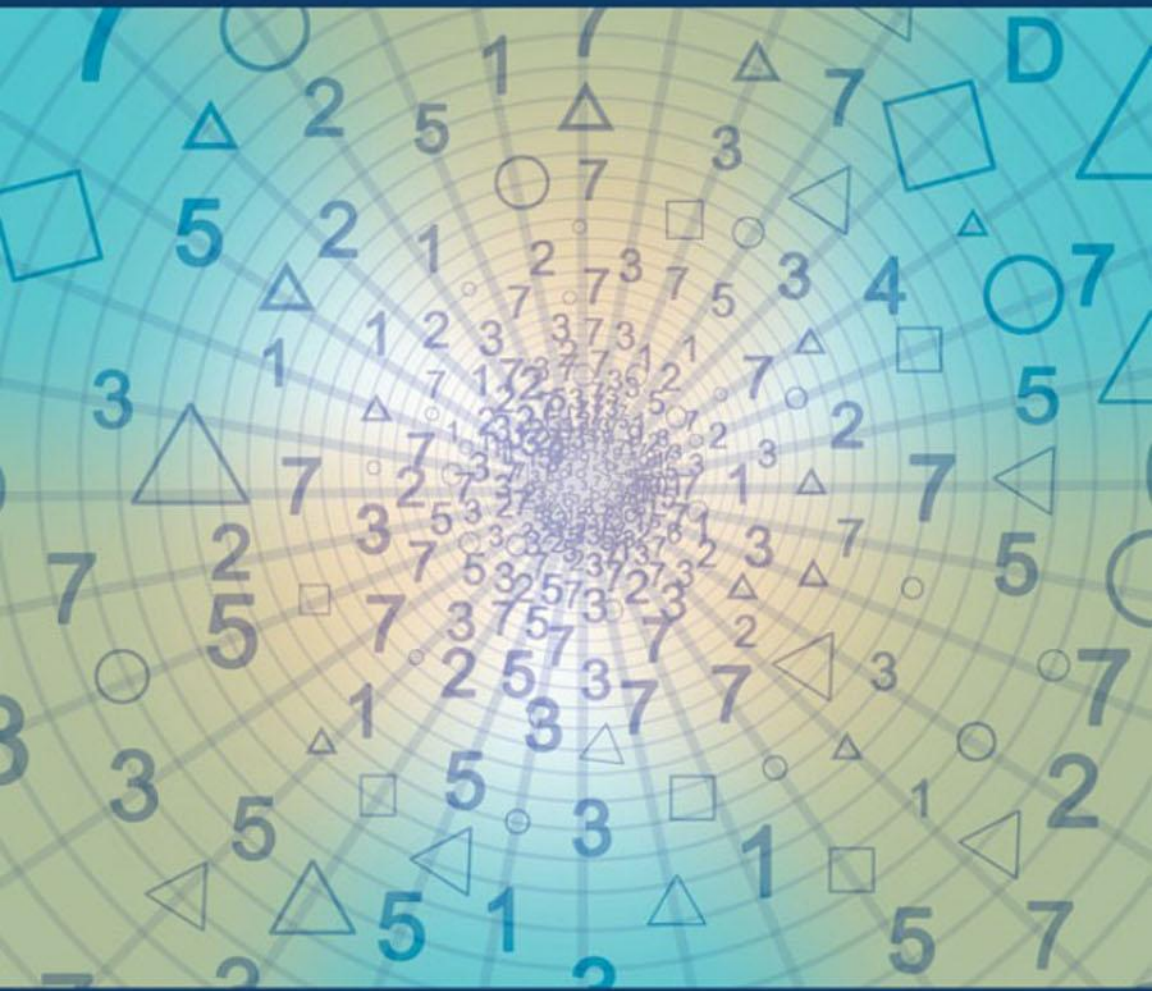
Simplicity Instinct

Why Prime Numbers Are Elusive!



Philip Gervase Jackson

SIMPLICITY INSTINCT

Why Prime Numbers Are Elusive!

by
Philip Gervase Jackson

Strategic Book Publishing and Rights Co.

Book Design/Layout by Kalpart team

Strategic Book Publishing and Rights Co.
12620 FM 1960, Suite A4-507
Houston, TX 77065
www.sbpra.com

ISBN: 978-1-62212-509-8

To my father whose passion was to find the principles in civil legislation he studied in order to serve his clients and his conscience which ensured that those that could not afford the best advice, still got it.

Table of Contents

Acknowledgments

Many thanks to so many people:

To Bernie Radford for pushing me to prepare my papers for publication on the Internet. Also to those people who listened to my message about simplicity with interest and provided me with additional ideas or encouragement.

To my friend Andrew Simpson for providing access to his network of contacts, which proved extremely invaluable for the resources it brought me in contact with and the people.

To John Gribbin, who, in his book Deep Simplicity, gave me the confidence that I was heading in the right direction as I developed and articulated a powerful problem-solving technique.

To those people, particularly at Microsoft, but also Apple and other software manufacturers, who overlooked the simple things that would have made their systems easier to use and focused on looks over substance.

And special thanks to mathematics researchers for concentrating on complexity and abstraction while leaving me to discover simple things that should have been known many years ago.

Introduction

The origins of my interest in simplicity can be traced back to when I was a small five-year-old sitting in a Sunday school classroom listening to a young lady teaching us about the Bible after attending a service at St John's Anglican Church in Roslyn, Dunedin, New Zealand. Even now, I can distinctly recollect the sudden epiphany I had that, as I was listening, something just didn't add up. Of course, the language that I used in my sudden thought wasn't quite as refined as that. By the time I was seven, I was regularly debating the existence of God with my father, sometimes going off for a furiously paced walk down Highgate in wintertime so that I could muster some new argument to take back to test on him.

Over the next six to seven years, I had many opportunities to debate this subject with roving born-again Christians, and looking back, I realize that I had started to identify the core principles of people's arguments. That proved very useful in rebutting their arguments quickly and effortlessly. On looking back at my atheism, I think that it is perfectly consistent with my interest in simplicity. A belief in a higher power along with all the trappings of religion requires complexity as an explanation. Many of my friends are Christians, and I enjoy their company and the odd debate. I can appreciate the relevance of religion to others and the valuable tenets religion has given to society.

Strong academic ambitions were never part of my character, so I more or less plodded through Otago Boys High School and left Otago University with an average degree in zoology. I followed this with a somewhat better performance in attending Polytechnic where I received a technical qualification, the New Zealand Certificate in Science. Sometimes, I had interesting debates, but mostly my brain lay dormant until I returned from living in Sydney, Australia with an Apple II+ computer in November of 1981. At an Apple users group meeting in the then new university Arts II building, I remember listening with great interest to a speaker describing how useful it would be to take information on paper and have some way of entering it into a computer quickly and accurately. This was many years before scanners and optical character recognition (OCR) became possible.

For some reason unknown to me, this statement snagged my imagination. Out the other end came an idea to make it easier for a programmer to type in lists of program listings that commonly appeared in the user magazine Apple Bytes. I quickly assessed that Applesoft Basic was not up to the task and realized that

only Assembler could handle the need for speed. Not far from my parents' home where I had come back to live was a very bright young man called Michael Hamel, who had competed with my younger brother all the way through primary school. He worked as a consultant of some sort at the university computing centre in Dunedin and was already fluent in Assembler. He showed me a screen utility that emulated the Star Trek warp drive visual effect of stars coming out of the screen toward the viewer.

That started an ambitious eighteen-month project. At the end, I had a very fast program editor that used macros and look-ahead techniques to take away the burden of sometimes having to type in listings or even maintaining their own programs. It could do syntax checking using a primitive form of recursion that I only discovered the name of afterward. Commercially, it was a total flop with only two sales, but the late nights and concentration had rewired the logical side of my brain, and what was previously difficult to totally grasp during high school or university programming modules in mathematics became something that now was considerably easier.

During the development of this product, I had been doing technical research work at the university and made a useful breakthrough that gave me some good life lessons. On a six-month project, I was to use medical abstracts to find a method for measuring a relatively obscure fatty acid called phytanic acid. It has multiple branches off the main carbon chain found in all fatty acids and had been difficult to isolate and measure. A little girl in Southland, New Zealand, had been born without the ability to break down this fatty acid, and it caused her to be become quite ill when it accumulated in her body. By measuring the levels of phytanic acid in different foods, it would be possible for her parents to avoid foods with high levels.

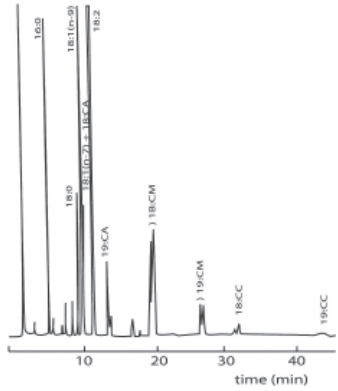
After several days, I found a method designed by a group in England that took two days for the analysis. Somehow, I was born with a laziness streak and that prompted me to think that there was no way that I was going to take two days to do an analysis; one day would suit me just fine. So, I looked more closely at their results, and to my surprise discovered that a professor, research fellows, and scientific officers had missed something that seemed obvious to me.

A gas chromatograph is a measuring device in which a gas is pushed through a special column packed with semi-absorbent material that retards the progress of different chemicals by different degrees. As the chemicals come out at the other end of the column, they are more or less separated, and you can see on the elution

profile (a record of the relative concentrations of different chemicals exiting the column) the peaks indicating various chemicals. By using known standard concentrations of these chemicals, you can identify which peaks correspond to which chemicals and how much approximately there is of each. An example of an elution profile shows what happens when similar fatty acids exit close together (see Picture 1).

On the two different columns using standard packing materials available at the time (e.g., DEGS—dioxy-ethylene glycol sulfate), this group had found that phytanic acid was superimposed over or exited at the same time as other fatty acids. What I realized was if I combined the two elution profiles in my mind, I could see phytanic acid coming out by itself. What I had done was to add together for each fatty acid the time to pass through each column, and I had concluded that phytanic acid would come out completely by itself. It was not physically possible to make a column that was twice as long and pack it halfway with one material and the remainder with the other material. Therefore, I made some columns with mixtures of the two packing materials. After experimenting with different proportions, I found a mix that did the job. Unfortunately, the chromatograph I had at my disposal was an old one and not capable of measuring the low levels found in different foods. My six-month job was rapidly coming to an end, and another research team convinced my supervisor that they needed me more than she did.

This project taught me two principles that are of immense value to all young people.



First, that it doesn't matter about the credentials of whom you may be competing against; it's what your competitors are doing, not what they have. Or, turned around, it is what you can do and not what you have in terms of qualifications. And, second, experts in their fields overlook opportunities. If you don't believe this, you will be less likely to make discoveries.

The new job in the surgery department of Dunedin Hospital under senior consultant Andre Van Rij was very stimulating. Although hired as a technician, I contributed to the design methodology, and through one of my observations, we were able to construct some equipment to help us measure a breakdown product of lipid peroxidation called malondialdehyde, or MDA.

After the government started to dry up the flow of funds to medical and other researchers, it was time to look to other employment. Computers seemed to be beckoning, and I applied for a job doing computer sales and software support at Whitcoulls in Dowling Street. One of my first challenges was to solve a long-standing problem with a customer who had bought Lotus Symphony, an integrated spreadsheet, word processing, and graphics package. He had problems printing on a printer with a tray of single sheets. The form feeder wasn't feeding a full page (the paper was feeding the incorrect distance), and no one else had managed to solve it. After about thirty minutes on site, I figured out what was happening and found a way around it. One problem solved, one happy customer, and I knew that computers were for me.

The branch I worked at was closed down when the company decided to do away with its business-computing store. Luckily for me, another computer store was looking for someone who could do sales and software support, and because I had started commercial programming, liked that as well. Within a short time though, I knew that Dunedin offered few opportunities in computing, and I decided to move to Auckland where I had a brother and other relatives living. It was with some sadness that I left the most character-filled city in New Zealand in search of greater employment opportunities and challenges.

Auckland exposed me to larger companies but also to business partners you had to be careful of. After one job installing Novell networks, I settled in working for a company in West Auckland where I wrote a large number of packages for customers in different industries. These varied customers exposed me to problems of many types whose solutions were simple ones. In 1987, just before the stock market crash in October, I wrote the first version of a futures and options back office trading package that specialized in the US market. One of the local users wanted to do margin trading on foreign exchange (FOREX or FX) deals, and the first problem that arose was how to define the relationships between currencies. The exchange rate as quoted between different currencies has a tradition in different markets to be used as either a multiplier or a divisor to exchange an amount in one currency to another. When I asked the brokers how this was defined, it was

immediately obvious that they had absolutely no idea. The worst possible solution was to have a two-dimensional table of all currency pair combinations with a high-maintenance problem of keeping it up-to-date.

Refusing to accept that approach, I looked for something simpler and intuitively came to a suggestion that I needed to have two tables of values, a ranking table and a relationship table. This worked a treat, and meant my customers only had to update a minimal number of rates between the USD and all other trading currencies, and the rest could be calculated from the tables.

My next big challenge was with a multilevel marketing company based in Dunedin who wanted a way of managing a binary tree (from the highest level, each descending branch splits into two) of members and their positions. That wasn't that difficult, but what was difficult was the need to be able to allocate blank positions in the membership database to new members who were sponsored by another member. Databases are designed to find records. In this case, I needed a database to help me find missing records beneath any location and the first position in the highest level available. This took two to three days to figure out, and what I ended up with was a bitmap representation of the database built up in blocks to represent groups of records in the membership database. Once the sponsoring member was located, I then proceeded from left to right a level at a time, with each subsequent level becoming wider, looking for the first available position. The company got into strife with the Commerce Commission on its similarity to pyramid selling schemes, and ceased trading, which left me out of pocket for several months' work but the problem-solving experience provided another step in the direction of building a simplicity instinct.

However, it was in politics that I found intellectual stimulation to develop the skills that made it possible to understand principles of things, and thus be able to develop an understanding of simplicity and the many things related to it.

In Auckland in 1987, I enrolled in a number of personal-development courses including public speaking, ballroom dancing, classical singing, and a philosophy course run by the School of Philosophy. This had been founded by a westerner who had spent some time in India in the presence of one of the Eastern philosophers, and when he returned, he set up a school based on the teachings he had encountered. A year was enough of this course, but it did expose me to ideas of practical philosophy, as well as the faulty thinking of their theoretical underpinnings. The course was supposed to help the average person gain more perceptive intelligence, but its arguments, although convincing for simpler souls, were unsatisfying, and

I decided to leave rather than keep on debating their ideas and spoiling it for the more passive participants.

At a pre-selection meeting in 1990 to decide the next candidate for the National Party for the Eden electorate, I meet Michael Bowden, a quietly reserved resident with a large rental property in the suburb. He was a couple of years older with similar interests in politics. We started talking regularly at a Friday-night meeting place in Balmoral, and one night we talked about a park that a local body council wanted to make a women-only sports park. We wrote a letter complaining about it, and due to the unacceptable response, I decided to look into this area further.

The further I looked, the more holes I found, and soon I realized that there was something wrong in the State of Political Correctness. After finding a swag of material, I decided there was enough for a small book. Once I embarked on some more lengthy analysis, I found some very flawed and unprincipled thinking involved in the Human Rights Act, the International Year of the Family (mainly anti-family), and all the little coteries driven to push their own peculiar wheelbarrows.

As I unbundled the principles from the emotive mires, I began to appreciate the massive levels of intellectual dishonesty inherent in the proponents of political correctness. Once I had self-published a book that sold around a thousand copies, enough to cover my costs, I retained my interest in this field but began to concentrate more on my own business, which had started after I left my business partner over his dishonesty.

In setting up my own business, I had taken clients with me, and they provided enough work to get started. Over the next five to eight years, I made numerous contacts and became involved with a package for the Flat Glass industry. Being asked to support a package with five to six megabytes of code is a big ask, and I needed a tool to allow me to look at that code both at an overall level, as well as being able to drill down to the lowest levels. No such tool existed around 1995, so I wrote my own tool, which I still use today.

Business trips took me to the States where I worked in several factories on production scheduling capabilities and gained valuable experience and contacts.

After a fall-out with yet another partner, this time due to his family, personal, and drinking problems and his sometimes unethical behavior, I left him aside and continued to develop a Windows package for the same industry. That set me off on the often arduous but fascinating journey that follows.

Chapter 1: Why Simplicity?

We live in a complex world where many things compete for our attention, whether they be advertisements, people we work with, noises, movement, colors, or smells. Although we have brains so powerful that even the largest supercomputer performs at a fraction of our natural computer's ability, we are unable to handle so much concurrent input.

We all have different capacities to handle all this input, but we also have our limits. The pace of technological development is resulting in a world that is getting increasingly more complex as it provides us with more options and more details. Consumer electronics and software are ubiquitous but hardly ever used to their full potential, because they are too difficult to use.

Suppliers do often make an attempt to produce easier-to-use equipment by using testers, but many others leave usability to a fashion-conscious designer cum-artist.

Simplification can help the world maintain this hectic pace of technological advancement. Different universities, including Massachusetts Institute of Technology (MIT), have projects on interface design,

but little obvious evidence exists that their research is making any difference yet.

We need a revolution that moves past the false assumptions made by many researchers and designers and assists individuals and groups in acquiring the skills that will take design to a new level. In the same way that Charles Dickens brought fiction to the newly educated middle class in nineteenth-century Britain, we need people who can bring technology to the newly educated lower class in the twenty-first century.

Complexity and simplicity are opposites. Complex systems abound around us and are the result of the interaction between two or more factors. Esoteric theories have been developed about complexity of which numerous commentators have been critical. Researchers have described closed and open structures, variety, and distinction without making any headway visible in the real world.

Modern (as opposed to Greek) Chaos theory has shown how simple behaviors can lead to complexity in visible steps. Deep Simplicity by John Gribbin, author of In Search of Schrödinger's Cat, is a worthwhile read as it covers the main researchers in this field. It demonstrates different areas of research that have used Chaos theory to give an understanding of how complexity in the real world can be the result of

very simple things interacting.

Taking this further, we have the universe, our knowledge of it, and its mysteries sitting on top of a hidden network of structures. Possibly everything we can see or record occurs above hidden layers of simple things interacting to produce what we see. The discovery of understanding comes from digging beneath what we see to try and find the support structures of all things around us, not just in physics and chemistry, but also in biology and behavior.

There are already organizations and individuals working in the area of complexity as it relates to businesses. American research companies such as the Standish Group, Gartner, and IDC already provide analysis of the costs of complexity. In January 2004, the MIT Media Laboratory with Professor John Maeda initiated a major research agenda focused on simplicity. The late Michael Dertouzos wrote The Unfinished Revolution on the need for users to make profound demands, while Donald Norman, the author of Design of Everyday Things, has also written The Invisible Computer, an argument for humane technology. Edward De Bono's Simplicity passionately drove the argument in its favor.

Unnecessary complexity imposes unnecessary costs, lowers productivity, and causes uncertainty; thus, higher error rates. Research has shown that in-

dividuals who have to change tasks often struggle mentally preparing for a change in role.1 That can also be taken as too many different ways of doing something amounts to overload.

Some companies do adopt simplicity, but they are rare in numbers. Vodafone (Independent, May 2005, UK) has spent a huge amount of money adding new features to its mobile phones, and now, having listened to its customers, has produced new models that make calls, text messages, and nothing else. Vodafone believes there is a big market for simplicity. Screen layouts have largely abandoned icons and replaced them with words. The models are called Vodafone Simply 1 and Vodafone Simply 2. A sure bet for success that will result in lower support costs. Vodafone has finally realized the competitive advantage of simplicity but still could have made it even simpler. My wife only occasionally uses my mobile phone, but every second time, she asks me which button to press to send a call that she has entered the number for. Like my mobile, the Vodafone one has a button with a little green phone icon (used by the little green man that in New Zealand we see on the electronic sign when we can cross the road at intersections) that is used to make the call. There must be many others who find icons awkward and unintelligible. A simple label with the caption of Call,

Dial Number, or Make Call would save her and others from having to ask the same question again and again.

Complexity with computers has given rise to new phobias: Technophobia and Computerphobia. These are real conditions brought about by technology that overwhelms its users. A good friend who is a senior manager at an up-market multi-branch jewelry store in Sydney, Australia won't touch a computer at all. Many young people seem prepared to put up with complexity, but many more others shun the complexity monster and lose opportunities in work and access to information.

Complexity has many disadvantages for companies, and they all affect profitability. Companies with complex products that are their mainstay are always going to be in a precarious position; when someone comes along with a simpler product, they will lose a market share. Soft Tech is a company in New Zealand that owns V6, a software package for designers of balustrades, storefronts, and many products that use glass and aluminum, wood, or plastic. It competes with three to four others worldwide and used to have the market in New Zealand sown up tightly. It is known as a complex and difficult-to-use product. Users are generally quite savvy and can be described as power users. In 2004, they lost the busi-

ness of a large number of customers to a product called Caliburn (renamed as Fusion for the local market), which was much easier to use, despite it coming, and having to be supported, from England.

New Zealand has a number of exported software packages competing in world markets, and some of these are even more complicated. Excellence's Dental Exact software stands out as an extremely complex product whose design coherency is clearly out of control despite it making some important sales overseas. Several times I have chatted with my dentist as I see him struggle with it, and he's no slouch. The strategic design of this package reveals that the designers have completely lost control of the complexity monster with menus emulating Windows menus on some dialog boxes and true Windows menus on others. That it is a monster is borne out by the time that has to be spent teaching users how to grapple with it.

Those individuals who seek to influence others, and there are many of them, are mostly losing out on their goals because they have not sought to understand how people assimilate ideas. Intellectual and academic obfuscation might go well with peers, but simplicity is the method that will connect with the masses. The art of writing simply is as difficult as the art of designing simply. It takes much training

and understanding of how others see complexity and how they respond to it.

Complexity is a very good hiding place and can appear to be as cavernous as Doctor Who's TARDIS. Not only does it hide simplicity, there is room for many other things. It is used wittingly to hide lies, deceptions, ulterior motives, negligence, irresponsibility, fraud, and a whole host of negative things. Unwittingly, it hides faulty thinking, dogma, and entrenched attitudes. If you pull out the simplicity, it is like pulling out the plug of a bath filled with many things hidden by murky water and suds. Soon all the previously hidden things are left high and dry and thus exposed for all to see. This is why simplicity is the enemy of so many things.

Simplicity is the White Knight charging against the Dark Knights of complexity, stripping them of their helmets and armor while revealing what lies within. Simplicity can be used to show that there are real truths and worthy principles that govern not only the nature of things but also the nature of society.

This book attempts to sink deep down into the quagmire of complexity, searching for its nature, and showing how it can be engaged in battle. The search for simplicity can be a difficult struggle, but it is well worth the effort. If it can be identified and articulated, great rewards lie waiting for those with the nec-

essary skills.

This book is aimed toward software developers, scientists, mathematicians, business owners, designers passionate about their consumers, venture capitalists, budding politicians, and anyone with an interest in using simplicity to give themselves an edge over others. Too often, project managers and owners give too much power to designers because they themselves are not technologically competent. In my own position, I was able to implement simplicity because I had the power and understanding of its importance, and now I wish to share the benefits of that with others. Dare to lay strict simplicity guidelines for your designers and developers if they don't have the simplicity skills, and hopefully with this book you can support this action.

No attempt is made to create arguments that would only survive in narrow university corridors or sealed-off newsgroups. Instead, ordinary language is chosen as the conveyor of ideas. Any person of average intelligence should have little trouble understanding them, but there will also be appeal for those interested in wanting to know how to adopt practical ways of identifying simplicity and complexity for specific scientific and business needs.

There is an emphasis on software in this book because that is one area that impacts most people

day-to-day. Usability and simplicity in programming courses and web site design courses are typically done after the basics have been taught. In my opinion, this is far too late. Usability and simplicity should be taught first and used as the base upon which everything else is built. It is often too late to introduce it later in a course when bad interface and programming techniques have already been assimilated.

Developers are highly capable individuals but mostly share a fierce independence and a belief in what they do as the best way. That makes it a struggle to bring them into line, and this will only happen when their employers are aware of what is needed. This book will help them acquire that information.

If someone truly understands simplicity, they can explain it to any rational, intelligent person. Like the sage who can express great wisdom in few words, the Simplitition is able to communicate the principles of something in few words and in a way that many people can understand. And, like the poetic French philosophers who love to hide their fuzzy thinking in colorful layers of emotional epithets, the non-simplitition is unable to communicate with anyone but those who share the same ideas or knowledge. Derrida and Foucalt were prodigious cloud makers, almost as if they were competing with Mother Nature itself. From a distance, clouds have shapes

that invoke our imaginations; they have context and color. But, if you fly an airplane into the middle of one, the shape disappears, and you don't find much at all, just some water vapor. Many of the grandiose ideas of these men and others like them are clouds. Get inside them, and you will often find them to be empty as they lose their apparent emotional magic.

Professor David Hilbert, a famous mathematician from the 19th and 20th centuries, once delivered a lecture before the International Congress of Mathematicians at Paris in 1900 on mathematical problems. He quotes the following in support of simplicity. "An old French mathematician said, 'A mathematical theory is not to be considered complete until you have made it so clear that you can explain it to the first man whom you meet on the street.'"

This clearness and ease of comprehension, here insisted on for a mathematical theory, I should still more demand for a mathematical problem if it is to be perfect; for what is clear and easily comprehended attracts; the complicated repels us.

Here is a link to his full speech: http://aleph0.clarku.edu/~djoyce/hilbert/problems.html

It is worth reading for it talks about the importance of solvable problems to mathematicians. His argument for understanding has largely failed to influence later researchers.

In my seminars on simplicity, I have seen people spellbound by the subject matter of simplicity but rarely overwhelmed or disenfranchised by it. In my deliveries, I am well aware of how straightforward the matter is. It almost feels trivial because it removes the mystique the emotion and the subjectivity, leaving simple ideas that can be easily digested. People can see I am passionate about simplicity because I find it fascinating, mentally uplifting, and exciting. Some of the ideas are well suited to expression using words of emotion, but the core ideas lack emotion.

Perhaps this is why many designers don't want simplicity. They want to imprint their emotions on their designs. They want uniqueness of design. They want to express the Monet within for the whole world to see. They want the use of their products to be an emotional experience. And that is often the case except the emotions can include frustration, anger, depression, and surrender to a classy design that fails to deliver on the functional level.

The minimalist art movement has embraced simplicity and so has Japanese culture with its tranquil manicured pebble gardens and bonsai trees. We need designers of all things to embrace it, so the consumer is able to enjoy the functional benefits that products were designed for instead of sacrificing function as a virgin on the altar of the goddesses of style and looks.

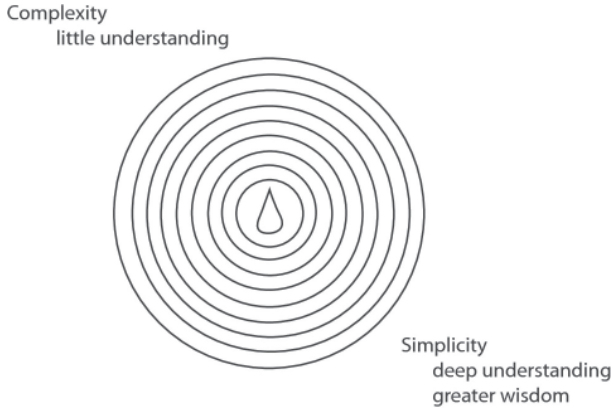
There is a famous orange squeezer built as a tripod with a central part tapering to a point below it that is supposed to direct orange juice into a waiting receptacle. It has become a collector's item for its looks, but as an orange squeezer, it failed the practical test. As a visual spectacle, it encourages a sense of an alien predator. It is the Superb Alessi Juicy Salif Juicer by Philippe Starck shown here.

The youth of today are busy developing personalities on the outside with tattoos, body piercing jewelry, cosmetic surgery, catchy hairstyles, and the trendiest fashions. These contributors of tomorrow will perpetuate form over substance unless society is able to focus on things that truly matter. Simplicity adds depth to life, work, and play because it gets to the bottom of things and reveals what matters. Simplicity can reveal the true nature of things and thus deliver great understanding. Complexity and superficiality reveal nothing but themselves.

There is an argument used to support the idea of further dimensions in space by using an imaginary group of beings called the Flatlanders who live in two-dimensional space. They are unable to see into

the third dimension, and, as far as they are concerned, it doesn't exist. We know it does exist, and the argument goes that we cannot be aware of higher dimensions also. It doesn't prove the existence of other dimensions, but it means to open the mind to the possibility. A complex world is like the second dimension, a world without depth. Simplicity adds the extra dimension in terms of understanding. Instead of seeing the surface of complexity, simplicity allows us to look beneath to see what lies further down. Think of the simple nature of things as the core of an apple and its complexity or expression in nature as the skin we can see, touch, and smell.

Consider the following picture, which represents a single drop of water in the center, surrounded by circles that represent the increasing complexity as drops of water coalesce when exiting from a tap. On the outside, it has become indescribable; yet on the inside it is simple.



If we are sitting on the outside of this relationship, we have little understanding of the nature of it. In attempting to describe it, we struggle and are forced therefore to describe it in verbose terms or create complexity in order to try to explain it. Pure mathematics and philosophy are salient examples of fields that sit on the outside of such objects. In mathematics, they invent great complexities and new vernacular in a clumsy attempt to find proofs while philosophers write numerous large books to explain their understanding.

As you penetrate the outer layers of complexity, your understanding and wisdom increases, and, as this occurs, your ability to articulate this to others improves, and you find that you need fewer words to do this. The greatest philosophers are ones with the deepest understanding of things, and they therefore are able to communicate this knowledge to others more efficiently and more effectively with fewer words. The Dalai Lama is one of the few individuals alive today who is able to convey great wisdom simply, and when this is done, it is barely contestable, which is much more than can be said for post-modernist "philosophers."

When I was a zoology student, I would often like to turn stones over to see what lay beneath them, hoping to find some unusual insect for my collec-

tion. Stones as we see them can represent the world. Finding out what lies beneath them represents understanding. In the real world, it is easy to turn over a stone or small rock, but, in the seemingly intangible world of simplicity, the turning over of metaphorical rocks to discover what lies beneath them is very difficult but worthwhile.

In the BBC's Hitchhikers Guide to the Galaxy, the secret of life was 42. That simply isn't true; it is simplicity itself. All the mysteries of the world have their simplicity shrouded by an intricate shroud of complexity. Revealing the simplicity brings understanding. Simplicity is the magic of life. It will explain our universe, provide us with design and operating advantages, supply an intimate knowledge of mankind, and give us what we need for more enjoyable and meaningful lives.

The happiest people are not the richest in a material sense; they are the ones who have rewarding lives from what they do, whether it is regularly mixing with a close circle of friends or family or their pursuits. Anecdotal comments talk about simple pleasures in life such as reading a book, gardening, or going for a walk. Even the busiest individuals need time out to relax and recharge and choose something simple such as lying on a beach in the tropics. Simplicity is the cure for complex lives and work.

New Zealand has had its share of poorly drafted legislation put together by those individuals who lack the simplicity skills to produce high-quality legislation. Business opportunities have been suppressed, unnecessary costs imposed, and constraints put in place without foresight. One such Act is the Health and Safety legislation that shares tests of accountability with sister legislation in Australia and Britain.

Numerous examples abound from these countries of local bodies and organizations being made accountable for actions that are either unpredictable or out of their control. In Australia, it has become so bad that the organizers of any public event are required to take out indemnity insurance before they can proceed. Community events have had to be cancelled indefinitely. In New Zealand, moves have been made to make charitable trusts accountable for accidents befalling their voluntary collectors, no matter where they are.

In Britain, the Metropolitan Police had to defend themselves against the HSE, the body responsible for pursuing cases of health and safety, when one of their policemen was killed while chasing a criminal. He fell through a roof.

The reason for these is because no one is sure any more of whom should be responsible for an accident. This isn't helped by those believing that all accidents

are the result of negligence, and therefore preventable.

The fault lies in the test for accountability itself. It places the responsibility on employers and authorities for ensuring that they take all practicable steps to ensure the safety of their employees or those under their care. This test makes no allowance for common sense as it explicitly says that anything that can be put into practice should be. Several years ago, an airliner leaving Auckland International Airport lost a small section of a wing that plummeted to the ground, making a hole in a factory roof but luckily missing the employees working in that area of their factory.

Any factory under the flight path of planes taking off or landing under this law should take all practicable steps to prevent accidents. If you interpret it literally, these factories should build fortress-style roofs, relocate elsewhere, or move underground. These can all be put into practice, but clearly, that is a ridiculous solution to a ridiculous challenge! Making the test practicable doesn't make allowance for controllability and predictability. If you have no control over something, and you cannot predict it, and it is also an unlikely event, why should anyone be held accountable? The simple solution to a complex problem is to remove the test and make it instead a test of controllability, predictability, and probability. It allows judges

to use their own common sense instead of coming out repeatedly with the statement (printed many times in our newspapers) that So and So Company failed to ensure the safety of their employees and therefore must be held accountable. Which employer could absolutely guarantee the safety of their employees and why do many magistrates parrot this nonsense when they are predominantly astute individuals?

Despite this test being around for at least ten years, no judges have commented about it, nor has any politician taken responsibility for the problem. Judges and politicians well versed in simplicity and principle detection would have never allowed such legislation to pass uncommented on or conceived such a test in the first place. Simplicity is obvious to those trained to look for it. Politicians have many ideals, and a strong attachment to them prevents them from achieving their full potential. New Zealand has had to suffer the results of decisions lacking simple insights through successive governments. We have sold our telephone network and cannot get true competition because governments continue to not want to take responsibility for their actions. We sold our railway lines for $1 and had to buy them back for tens of millions.

When I bought my first home in Mt Eden, Auckland, it was in the days of the New Zealand On

Air license fee. Avoiding having to pay this fee was relatively easy but just as easy was the identification of the absurdity of this charge. The intention of the collected fees was to provide funding for National Radio as well as for the national television Channel One and Channel Two. People only using radios were not charged the license fee; instead, television owners became liable even if they watched other channels. In essence, television owners were being charged for a television signal that was unsolicited, uncontrolled once it had been transmitted, and in no possession of anyone. Stretch yourself and think of any other situation where something matching these principles was being charged for. Even water is controlled, so that doesn't count, although charging people for using up oxygen belongs in the same ward in a sanatorium.

Articulating simple principles is a popular and successful way of selling ideas. Go into any bookstore and you will find a section containing books by Edward De Bono, Jack Welch, and others that give simple principles for management, marketing, and many other areas. No one willingly reads a tome of complexity to distil out the useful ideas. Presenting them simply allows someone to understand something quickly that could be useful to them. That is why such books as The 7 Habits of Highly Effective People by Stephen R Covey or 7 Hats by Edward de

Bono have done so well. Even if you are not planning to write a book, you may want to present seminars, lead a political campaign for change, or create a marketing strategy. In each of these, simple principles are going to be more effective than complex ideas when it comes to influencing others. Simple principles are easier to hold in short-term memory because they require less effort and are easier to repeat to others by way of word of mouth. But, most importantly, they are easier to digest.

Those people with long experience of solving performance or operating problems will be able to look back and see how a simple mistake can produce apparently unfathomable behavior. Software developers know that what seems irrational on the surface can be explained by simple mistakes beneath it. Solving such problems should focus on simple mistakes. Even though software can operate for months without problems, there can come a time when it operates in an unexpected way. As I write this, I have been solving a problem of this nature for a customer site a third of the way around the world. A software routine that had operated normally for six months had just started misbehaving. After a few hours, I found a + sign in a field that had no easy way of getting there, and this had stopped batching of orders for production. Initially, I had looked for something

complex, but now that I have the answer, I can explain the behavior. Upon reflection, I am able to see how I could have located the problem sooner, but, under the pressure of time, I made incorrect assumptions.

Operating problems produce unusual but observable behaviors, and those behaviors are clues to the causes. Any information produced contains manifestations of those causes.

If you have a list of items being processed, such as in batch processing, and not all items are being processed, there is a termination condition being met earlier than expected. In simple lists, this can be trapped easily, but when the result or pre-selected list of items is presented in a different sequence to the individual items, you need to work out which item in the result list is last in the processed input list. Once you have this, you can set a breakpoint inside the processing routine to stop when the last accepted item is seen and then step through the selection process looking for termination exception possibilities.

If a single item is being processed, you need to break down the processing steps and monitor completion of each step. In software, this can be done by saving stages reached to a text file or by showing screen prompts.

When a software language doesn't lend itself to

debugging, you could try adding audit information to the output list as well as showing the exit condition. If you have multiple conditions for exit, structure code so that it identifies which condition causes an early exit.

When software locks up, it has usually entered a loop where the termination condition is never met. In software that has been in use a long time, incorrect data entry may be the cause. Inspect all the data files and fields referred to in the problem routine, and write a program that identifies data outside the expected ranges or create exit conditions for loops that are executed when an excessive amount of time has been consumed.

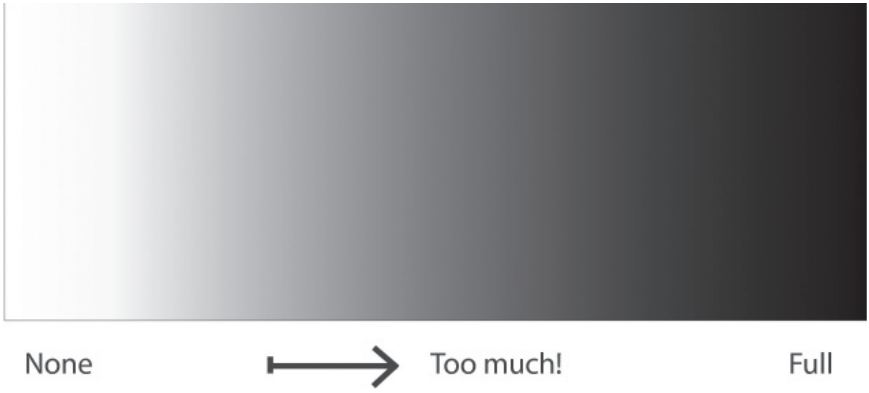
With all these problems, a simple approach looking for simple causes is going to prove more successful than anything else.

The world we live in has many problems and issues to contend with. Pollution, recycling, wildlife preservation, deforestation are just a few. Governments continue to produce bad legislation and impose policies that are dishonest. Judges have become more subjective and subject to poor principles. Social engineers impose policies that achieve their ideals but are divorced from common sense and cause structural damage to societies.

These types of problems need deeper simplic-

ity analysis to derive worthwhile principles and then to promote policies and action that are consistent with these principles wherever they are found. Inconsistency is the lowest level of integrity. Consistency using sound principles is the highest level.

The overriding reason software overwhelms many users is because it has too much non-data detail. This detail includes menus, buttons, icons, shadow lines, and other graphical elements that add to the visual loading of a screen or printed page, creating a competition with data for the eye. If we were to draw a continuum of competing detail from none to full/blackout, we would see something like the following.



There is a point that is probably similar for most of us beyond which it becomes noticeably irritating to look at material when there is a lot of background

detail. At some point further, it becomes very difficult before becoming impossible.

At a subconscious level, even before the point where we become conscious of competing detail, the mind is doing some filtering. You can prove this yourself by noticing how your sensitivity to background detail changes over time once you become aware of just what level of superfluous detail there exists on many software screens or written material. What is happening is that your conscious mind is gradually tapping into the subconscious. In myself, I have noticed that I am much more intolerant of this extra detail than I ever was, and it has helped me comprehend why some people don't like using computers at all.

Software is too difficult to use, not only in the way it looks but also in how you use it. Developers have styles of interface that make their controls stand out whereas users want their data to stand out. Software needs to be simpler in looks, and the controls need to be subordinate in looks to the data. It is not supposed to be a competition of focus, but controls should be complementary and present when required.

***

Buy the B&N ePub version at:-
http://www.barnesandnoble.com/w/simplicity-instinct-philip-gervase-jackson/1112041703?ean=2940014617482

Buy the Kindle version at:-
http://www.amazon.com/Simplicity-Instinct-Numbers-Elusive-ebook/dp/B008JGIVC6/ref