System Engineering & Design Architecture

Sander R.B.E. Beals

This document is Public Domain, and may be used by anyone for any purpose, except patenting....

Table of Contents

This document is Public Domain, and may be used by anyone for any purpose, except patenting....

Introduction

An amusing discussion with a colleague the other day made me realize that the last statement on systems design hasn't been uttered yet. We were on opposite ends of an obvious standoff, where he claimed that hardware development and software development were essentially different, and I defended the gut feeling that despite a few differences, there must be a similar (set of) mechanism(s) behind them. In my opinion there should be, because software simply cannot live without hardware, and only simple hardware (like a hammer) can do without software. In order for them to cooperate, there must be a certain 'resonance' in their mutual cooperation. Other than that, they are basically all hybrid systems, with a sometimes delicate mix of mechanical, electrical and software aspects, that need to be designed as a whole to retain overall consistency. In the current timeframe, invasive interfacing to biological systems is seen as a research subject, but I believe that even that will eventually become commonplace. It is just a matter of seeing where we are comfortable with connecting to our tools, like the hand that yields the hammer, the surgical steel replacement parts for bones or the syringe that helps us donate blood to those who need it desperately. And wouldn't it be great if we could have the understanding of a System Engineering and Design Architecture that would be similar (if not identical) for all aspects of Systems Creation?

This document will describe the structure of a system based on the SevenSphere aspect of Nature I became aware of in my efforts to make clear (to me at least) the various aspects of the world my five or six senses perceive around me. The introductory texts can be found in a trilogy of relatively light reading that can be downloaded as public domain information at http://moorelife.nl. The following titles are the ones I am talking about here:

1. "Infinity plus One", 77 pages in English, part one of the trilogy.
2. "Art of War once Moore", 50 pages in English, part two.
3. "LIFE: Love Infinitely Furthers Evolution", 82 pages in English, part three.

This document is not a true sequel to the Trilogy, but more of a textbook for a course on Systems Engineering & Design Architecture for normal (read non-technical) people, Muggle or magical person alike. As for me, I'm just the ordinary guy with a deep interest in observing the simplicity of things for what it really is, and describing it to others in similarly simple concepts.

To sum up the essentials of the SevenSphere for anyone not wanting to read the three books mentioned, the elementary operation is simple: any word written in the gray center sphere can be explained, further defined or refined by six other words in the colored spheres around it, either in a 1x6, 2x3, 3x2, or 6x1 configuration. This is not a rigid mechanism, but a way of working where we simply keep in mind that the human mind is said to not be able to keep track of more than seven concepts at a time. You might consider this as a sort of graph paper for understanding, because the various SevenSpheres can combine to form a virtual hexagonal (honey comb) structure like any of our more complex molecules, which is known to be a very stable structure in Nature (carbon compounds form this way). Either way, it allows us to clearly connect the stuff we talk about. Also, sets of seven usually have one main concept, which plays the central role in a SevenSphere, if it is not 6 defining one...

You'll get the hang of it as I talk you through the first few diagrams, which will make clear that it is a great way of making things clear.... ;-)

This document is Public Domain, and may be used by anyone for any purpose, except patenting....

Everything is a System

They say: "If you have a hammer, everything becomes a nail." Likewise, this document and its promise of a System Engineering & Design Architecture will turn everything into a System, by simply looking at it through System-colored glasses....

We already established that only fairly simple tools do not require software, just like only the simplest of our everyday technology doesn't use electricity. Most of our Systems are so-called Hybrid Systems, a mix of the following aspects:

1. Chemical
2. Material
3. Mechanical
4. Electrical (including magnetical)
5. Computational
6. Biological (sensory, manipulative and/or invasive)

Now describing it in terms of the constituent parts may be a simple thing to do, but with regard to the architecture of the whole it is hardly useful. We do not want to know the parts, but rather how they cooperate to make the whole. And that is quite another subject, given the fact that there are two sides to that whole, the view of the user, and the view of the creators of the system. For the purpose of this document, we must attempt to bring the two together, so the creators can create what the users want, instead of some distorted mirror image of it. But luckily for us, even design is a System. Just look at the SevenSphere below:

In this image, the creators are diagonally across from the users, who should be the ones the creators design their systems for. The users on their part form a stable triangle with the devices and interfaces they employ to use the system. They don't care which part is hardware and which is software! That is the realm of the creators: they know both the problem space and the solution space, and have the skills to realize a solution using specific combinations of these aspects to get things done. They may even have to go as far as to design a completely different system first, in order to get the user what he or she wants. Just think of the cell phone system: it has phones which the users use, and which interact with local cell towers that are only partially in the awareness of the users. Users may know they are there, but they are usually not able to tell you just how the connection between any two cell phones is made: that is an internal interface, known to the creators ***and implicitly trusted by the users***. The one device users do bother with are their cell phones, and the interfaces on them:

1. the charging port, for when the battery is drained.
2. the SIM slot to connect their number to the phone.
3. The MicroSD slot to enhance its memory capacity.
4. The headphone jack to listen without disturbing others.
5. the keys that have certain functions, as the software assigns them.
6. The screen, which can give you pretty much any interface, if it has touch capability.

This document is Public Domain, and may be used by anyone for any purpose, except patenting....

They may of course bother with the more technical interfaces of the phone too, but only if there is a need for it. So battery replacement, network connections, GPS capability, and other related stuff shouldn't be on the outside. Still, they are points of attention to both the creators and the users, and they show how interfaces are what connects both systems on the outside, as well as subsystems to the system under observation. I'll leave the exercise of creating a SevenSphere of the interfaces, both external and internal, to the users. Remember, this is an intuitive tool, so don't think about it too much... *(and don't worry if it doesn't fit perfectly)*

Since we're talking about Systems though, let's try and turn one into six terms that define a seventh:

1. First is the Environment, or the larger picture that the System finds itself in. Based on the Environment, the Strategy may for a large part be aimed at surviving in this Environment. Since this is usually the most dangerous part to a system, let's put it in the red sphere.
2. Interfaces form distinct communication or logistic lines to specific other (groups of) Systems. These interfaces may lead to outside systems in the environment, or to subsystems inside the System.
3. Subsystems are those parts of a System that it knows it can rely on because they are embedded in the system as such. Apart from their role in sustaining the system we are discussing, the subsystems have no reason to exist on their own, and often they even cannot do so.
4. Strategy is the Modus Operandi which the System and its Subsystems employ to fulfill their Intention (either assigned or freely chosen), and to maintain Integrity. Asimov framed these beautifully in his three robot laws (see App. B if you are curious, or no Sci Fi geek).
5. With our Integrity guaranteed by the above, we can proceed towards our Intention, and the development of more Interfaces if needed. Now these may be designed during the design process, or as the System is deployed, and found lacking in certain areas. Nowadays we will fill this with a change request, and engineers adding the missing functionality from 'outside', but in the future Systems may well become self-improving, in that they adapt their configuration to develop missing interfaces on their own. What I mean is this: if we could figure out how to make them, won't they eventually match our intelligence, or even surpass it? Nothing to fear though, just an exciting possibility of today's Status Quo.
6. In order to maintain Integrity, a System might well have a distinct Strategy to check up on itself every now and then, in order to call out for backup: just think of your phone again, complaining that its battery is nearing depletion. More advanced solutions to this might be to find a power source themselves, like for instance seeping off energy received from local cell phone towers in order to recharge the batteries. Another nice addition to a cell phone would be a few solar cells on the backplane, making it a 'cell phone' in more than one way: if you place it on the table with the screen down, the cells can charge the battery for you, thus leading to longer life....

OK, so we have some idea of what a System is in terms of its primary aspects. But how do we shape this into a clear and simple design strategy, that will be similar in approach for any hybrid system we could think of?

This document is Public Domain, and may be used by anyone for any purpose, except patenting....

Keep It Simple Sir....

That IMHO is the best place to start with: since humans are reportedly only capable of juggling at most seven concepts at any one time, our approach should be to not exceed that limit, as Queen hinted with the cover of their album Innuendo. That subconsciously (behind our backs) we also handle way more things is also concealed in that image....

The SevenSphere also hints at that limit, since we can clearly see that adding more spheres will 'force' them to be added to the essentially flat form either behind, before, or outside the round symbol we call the SevenSphere. Now before and behind there are 6 extra spots, but only three of those may be filled on any side at the same time. Once we hit thirteen (7+3+3+1), expansion beyond the bigger Sphere in which the SevenSphere lives is inevitable. Is that why we call thirteen the number of bad luck, because it requires growth?

My Dad used to say: "Colt 45, seven dead: six you shoot, the seventh you throw it at!" And as a matter of keeping it simple, that is his tools approach: use it as intended as long as you can, even if that means having to improvise. My approach is different: I consider myself a problem solver, and as such I try to devise new tools to solve problems or new uses for existing tools. And as for my wildest personal dream: ***solving the conundrum of problem solving, wouldn't that be awesome?*** On the other hand, it seems more like a trip of discovery rather than a design approach. It is as if everything is already there... *(and I'm just connecting the dots....)*

A problem is only a problem if one is aware of it. And once you are aware of it, there are two ways of dealing with it: you either create a solution, or adapt to the problem, so it is not a problem anymore. And if you do choose an approach, you will need feedback in order to determine if you have reached your desired position of 'Problem Solved'. Now despite the fact that the title track of the movie 'Triple X' on my TV screen just had someone sing: "You're going to extremes, there's nothing in between", I figure there is: we could see the problem as an extreme opposite of our current position, and go all out to annihilate it, or see us and the problem as just two nearby specs in the infinity of possible approaches.

In that last case, there is always a solution, and it can be found quite easily, as long as we keep it simple. If we want to do that, what are our options? What are the most used and succesful mechanisms for getting to a simple and clean solution? Well, let's see (details in Appendix A):

1. ***K*** eep ***I*** t ***S*** imple ***S*** ir.... (or ***S***ister, no gender issue intended)
2. 80 / 20 % rule.
3. Divide and Conquer.
4. Stepwise Refinement.
5. Occam's Razor.
6. Thinking outside the Box.
7. The 7 item rule. (implied in the SevenSphere)

This document is Public Domain, and may be used by anyone for any purpose, except patenting....

I'm not drawing that diagram, since you could see it coming a mile away. But keep these in mind, for they will come in handy. Going back to the discussion with my colleague, this mainly had to do with the amount of effort required to reach success in the various stages of development. I guess that is the next concept we have to look at closer, in order to get to our aim of a generic approach to Systems development.

Sofar it has all been pretty standard, right? It is generic enough to apply to either software or hardware design, or even something like fashion designing (even there the cutting patterns form the design). But in the rest of this document, we will be tackling all of these six stages in a new way, that also keeps in mind the 7 mechanisms mentioned above. And guess what? There is even some stuff to describe about this diagram on the right here, which may not have been obvious to everyone:

1. Analysis, Design and Implementation are creational aspects of the development, and thus form a stable triangle.
2. Requirements, Documentation & Systems Testing are also related because they are the phases that produce documentation, and consolidation of the design and product. These belong to the product, but aren't the essence of what we aim to produce. They show the limitations of the system, so you won't dry your cat in a microwave and then complain that the manual should have warned about that... Also, the testing part uses the other two to verify that the triangle of creation was executed correctly.
3. Apart from that, the six phases are all in sequence around the center, as they are passed on the road to Problem Solution.

What is also a new approach in this document, is that we'll be moving the responsibilities around a bit. Just like Object Oriented Software Design treated the Classes as responsible for their purpose, we will be treating the entire development effort like it is responsible for its own role in the process. And we'll start by combining some concepts from OO in order to show how we get things done:

Now from here on in, it is going to be a bit different, because we will be looking at the System Aspects as if they are based on the two most used mechanisms in Object-Oriented programming:

1. Model, View, Controller.
2. Object Action principle.

To make it an even six again, we will add to that a new aspect called Integrity, which basically turns Quality into an attribute of the System, rather than trying to apply it from outside like a band aid. If something requires testing before release, it requires testing even out there!

This document is Public Domain, and may be used by anyone for any purpose, except patenting....

No doubt software for disciplines like hardware and building design has for several decades now included the calculations which determine whether the parts will endure the stresses on them so they are in fact source code that calculates the physical parts in terms of their actual minimal required dimensions given the material it's made of and its required strength.

In this case it is easy to see how the virtual world precedes the real one. So why not add Integrity to the System aspects in all types of Systems, hybrid or not? Based on what design discipline we are in, the concept of such integrity checking would be like polymorphism in Object Oriented programming: Even though the stuff needed to prove the system is working may be widely varying, it would in fact mean that the system has a 'standard' approach to determine its validity: SelfTest would mean "How do I feel?" to a human, but "Full Systems Check" to an android. It basically means the Three Laws of Robotics would be amended by a fourth directive, which would enable a system to determine if its behavior is still within the range of 'zero defects' for both humans and androids, which is actually ***required*** to be able to obey the three laws! Software for any system would actually have built in safeguards to determine stuff like:

1. Are my crucial methods still error free?
2. If I have redundant interfaces, is at least one of them active?
3. Is my Model still up-to-date?
4. Are my associates (other classes) of the correct version?
5. Can I reach all my partners through their respective interfaces?
6. Do I have error conditions for which help from the outside is needed?
   (outside being other systems within its environment, or even outside it)

Now for software that seems like an extra load on the system at runtime, but for the hardware part of the system, it would be done in the design program, and not in the manufactured product. Likewise, such checks could be partially or totally switched off in production releases of the software, even though programmers could use it to figure out the real problems faster because of the extra checks where it matters. But should we ever switch off the integrity tests, given the performance of todays systems?

If we want to mimic Nature in Design and Engineering, we basically only have to look at ourselves with a bit of analytical effort:

"what exactly do I do when I am working?"

We've developed a simple mechanism for that, which is kind of like the two SevenSpheres on the right: the top one is usually where we find ourselves from moment to moment, while going through a certain routine that isn't routine enough to allow us to do it without thought and attention. Keeping your attention on the task at hand is something an average human is said to only be able to do for about for about five minutes if we talk about children, and up to 20 minutes for fully grown adults. Now any task is usually non-atomic, and prone to incoming higher-priority tasks. Kinda like the needle that punctures the egg, the new task invades our awareness, and demands we drop the Current Task. Our default action should then be to immediately go to the red

This document is Public Domain, and may be used by anyone for any purpose, except patenting....

sphere in the top diagram, and record whatever information we need to pick up the current task later on. The top diagram is cycled clockwise, from he start of the task to the end.

The second SevenSphere is a 2x3 configuration: normally, we attempt to keep a nice balance between our Priority Stack, QA Effort and Incoming other tasks, right until the moment the work runs out. In that case, we switch to the triangle of Cleanup, Reorganize and Current Task, the last one being whatever we deem most fun and/or useful, whatever the Environment you find yourself in. Mind you though: if chaos interferes with our normal way of working long enough, then the second triangle may become the Incoming projectile that requires we take time to Clean Up and Reorganize before preceeding with the task at hand. Currently, I'm in here writing this document because to me this is a Cleanup and Reorganize effort that can fit into the current wait for my colleague to help me fix a certain problem for which my talents and knowhow just aren't sufficient. In a similar way, I just spent a good ten minutes sorting out the mess of cables underneath my desk, because my feet were getting caught in it.....

Now that last one was an ad-hoc action, basically aimed at improving my working life. But like they say programmers are lazy, and I'm no exception. Even though I'm a tester now, I still keep my solutions as simple as possible:

1. If you find yourself in rest, try to figure out which action is needed next. Basically, that is the diagram on the previous page.
2. If you know the action to take, ask yourself if it is an action that needs to be repeated in the future maybe. Because if you need to, then taking notes would be a prudent action...
3. Even in case you needn't repeat the action, taking notes and collecting relevant info is still advisable, in case your intended solution doesn't have the immediate result you were expecting of it.
4. Crucial is the top sphere: did we succeed in reaching our objective? If not we go further.
5. In that case we reread our notes, try to figure out where we went wrong, and devise a new action to achieve the objective anyway. This might mean adding an additional action to our solution process, or simply going back to the previous position and redoing the action slightly differently. Of course going back implies a backup or an undo mechanism....
6. With the improved action, we will be able to solve the problem sooner next time, and with more faith in our abilities.

There is still a slight problem though, and that is to devise a simple structure to channel all those notes into a river of information which will finally end in the sea of structured knowledge we want to be drawing our resources from. Since this is essential to getting our work done on time and above expectations, this document will describe how to set up such a system with simple means....

For me, the optimal structure is just a folder on my desktop in the Fence called Priority Stack, and first of all fill it with shortcuts to files and folders I might need in execution of the task at hand. If relevant enough, or the location of the originals isn't stable enough, I simply copy rather than shortcut them. Add to that a text or RTF file to make your notes in, and you're ready to go...

This document is Public Domain, and may be used by anyone for any purpose, except patenting....

To begin with, we call the process of gathering information and structuring it into knowledge Learning. The SevenSphere on the right shows this process in detail, as a circular movement:

1. We start from what we Know, in the light blue sphere of Knowing.
2. Some experience provides us with data, which we may or may not be able to see as information.
3. Our discernment (more feeling than thought) may have doubts about the information, which will cause us to either see the knowledge as something we can trust or not.
4. Either way, the Knowledge becomes part of our Knowing, either in an enabling way if we trust it, or in a restricting way if we don't. We build on the trusted knowledge, while steering clear of the things we hold for impossible, or simply not true....

Now this sounds all pretty obvious, right? We do it all day without even thinking about it twice, or even once! Most of us tend to make few notes, because we can remember (not be totally aware of all implications of) quite a few things at the same time. Only once the inputs exceed the capacity of our short term memory, do we feel the need to revert to pen and paper, dictation devices or a keyboard and screen, often packaged as a tablet nowadays.

Keeping track of things requires a system of some sort, even if it looks like total chaos to someone else. I once came into someones living room, which had documents and news paper clippings all over it. The lady told me she thought Home was the most important, and I believed that, but at the same time I had a real hard time imagining that anyone could actually be living there. Also, she talked about being a bit scared of computers, but at the same time her home had at least three of them in obvious sight. I was invited to help her with her computer, but did a piss-poor job: by overlooking one simple fact I had witnessed with my own eyes, I unfortunately installed a free office suite for her, to replace the Microsoft package that had no valid license key. Unfortunately, since she was only using Outlook, it never occurred to her that she was in fact using Microsoft Office. So when I asked her, she said no and I figured it safe to uninstall Microsoft Office, and replace it with something else. I totally forgot about having seen Outlook start up earlier, and the uninstall didn't warn me that I was about to lose all locally stored E-mails. Shit happens, but I've never been so sorry for making a mistake! Later she called me, that someone else had fixed it, but she didn't need to see me again, although there were no hard feelings. I could understand that...

Point in case: what seemed like total chaos to me, was her system for keeping things organized. Much like Lara Croft's engineer carefully put aside the screws he took out of the clock in separate quadrants as he called them: that was his system to "Know where they all came from". And just at this time, Level 42 plays "Lessons in Love": "*All the homes that we were building, we never lived in. Could be better, should be better, lessons in Love*". So yes, home is important, and to me it is a sparsely decorated but relatively decent family home. Yes, it needs a woman's touch, but I've just not yet found Miss Absolutely Right for Me! On the other hand though, hints about her being around abound in my immediate environment...

This document is Public Domain, and may be used by anyone for any purpose, except patenting....

A Higher Note...

We document stuff in many ways, from a simple Post-It note on a laptop to hint at the password to a fully cross-referenced series of volumes defining the communication between DICOM standardized medical equipment. In general, information in written form tends to proceed from simple to more complex, and from standalone to more connected. The believability of more complex documentation may be higher, but if the simplicity is lost in the process, then the reader has a problem in the way that he or she is no longer able to correctly deduce the knowledge from the information. A note on a laptop may be too simple to get something useful from it because an unknown reader is missing the context, but the logically complete information in the DICOM standard (some 4900 pages in 20 volumes) is only readable for certain people who already have enough of a basis on DICOM to be able to comprehend it.

We use many different forms of documents, from leather bound volumes with ancient texts to paperbacks or HTML and XML documents on the Web. Documents in any form provide (structured) information, rather than knowledge. Why that is so? Easy: we write it from our knowledge, but the reader experiences it from his or her knowledge. Since these two are by no means identical sets, there is a process of learning involved again, to turn the information back into knowledge for the receiving party. The amount of structure inside the information is crucial to its being understood, as we shall see later.

A Document in itself is quite unique, but we must relate it to the product for which it is intended, and we must give it a version so we can keep the changes apart in order to make sure we don't describe features in a manual that the product simply doesn't have yet. Version numbers separate the document changes in time, and configuration documents the structure of documents. Now in order to keep our story generic, we can say that what goes for Documents also goes for physical parts and even virtual parts like software: there are versions in the development because parts are improved and upgraded, and there is configuration info like design drawings which document how the various versions of different parts and documents fit together to make a whole product.

In order to correctly identify parts or documents, we need a unique label, which defines them in both versioning and configuration. But it is a little more complicated than that: since relationships between one part and another one can be M:N (like a 747 having 4 engines which share certain other subsystems), we must be able to properly document such complex relationships. This is stuff that needs to be left for later, because first of all we should look at the process of Change, in the way it shows in our design and support efforts.....

On a side note, XML would be a quite handy format to describe such information. Since we can define the tags ourselves, it is more flexible than other approaches.

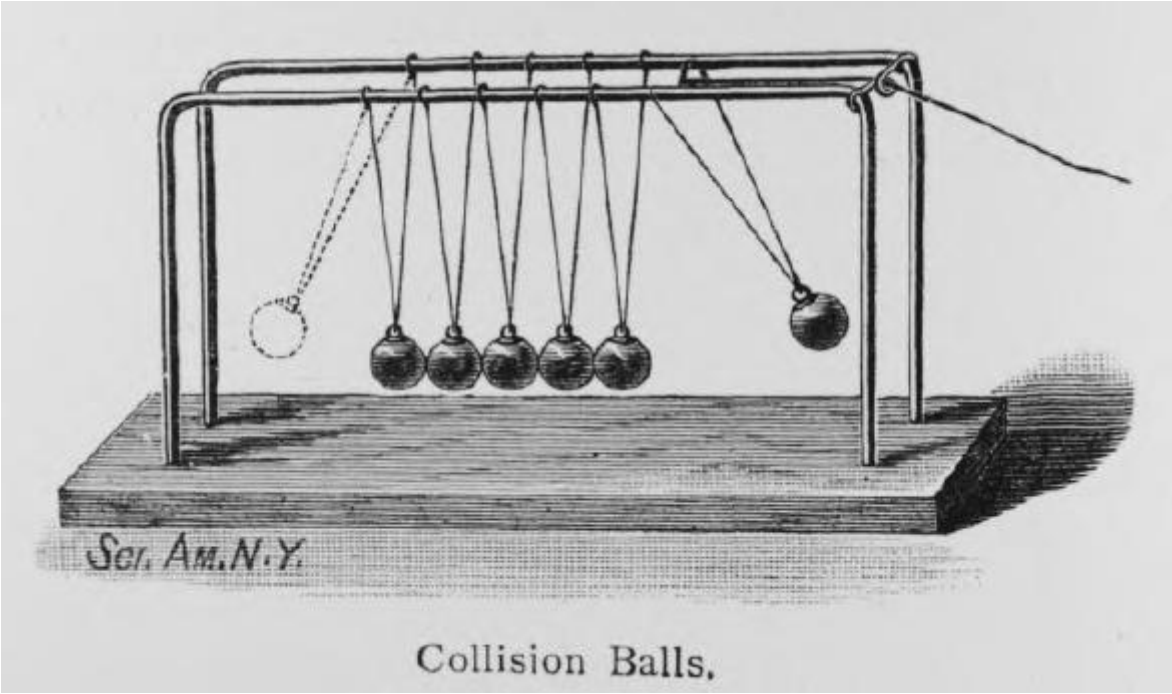
And of course we know from natural language that unique identifications tend to be blown out of the water by the torpedoes of ambiguity: labels tend to get several meanings, since their similarities in certain areas are binding while the context of the language around it keeps

This document is Public Domain, and may be used by anyone for any purpose, except patenting....

them apart. Now I was thinking of this all the way home, and figured it would be a nice article on my weblog at http://beyond.moorelife.nl, but for some reason my trusted FireFox refused to let me in there. Since she and I have a longstanding working relationship, I didn't doubt her warning, and went for the next viable option, which was adding it in this book. Why? Simply because it has everything to do with Systems interaction!

Let's look at us puny humans again: if we've had a bit of education we know that like all matter, we are built up of molecules that bond together atoms, which have a core of positrons and neutrons, surrounded by a shell or cloud of negatively charged electrons. What is also a key piece of (immediately deduced) knowledge in our minds, but what most people never even stop to think about, is that our outermost layer is thus negatively charged, just like the outer layer of any object. That negative layer actually makes us look negative to outside systems, regardless of their nature. And because of that, we also know we seem negative to others despite the fact that the molecules are electrically neutral themselves.....

Now if that was the sum total of it, then we'd be in big trouble: no way to look positive to others, or even see them in a positive light! But luckily there is a way out: molecules are bound together by forces beyond the repelling forces of their respective atoms, and maintain a more or less stable shape despite or maybe even because of the clouds of electrons that embed each and every atom. I'm not smart enough to actually explain if the electrons also have a given effect on the shape of the molecules, but that is too much detail: there is such a thing as form, and our molecules make them up. That on far higher levels the human decides to look a certain way may be his or her choice, but then again we can't always help ourselves (if we don't believe in it). My next point is something even Newton already knew, because he made the toy, and it was soon named Newton's Cradle:



Collision Balls.

Of course I couldn't find an older picture, but this looks like it's from Scientific American, in New York. I dare not date it though... The crucial point this device makes, is that forces tend to be passed on from one form to the other, in as far as they have freedom of movement, that is. A nice online animation for your enjoyment can be found at:

http://www.lhup.edu/~dsimanek/scenario/newton.htm

This document is Public Domain, and may be used by anyone for any purpose, except patenting....

Just play with it a while, so you know what happens when you lift 1, 2, 3 or even 4 balls out of their rest position, and let them collide with the rest. Now if it weren't for the nuclei, the electrons wouldn't have any shape to cling to, and might be racing around like gas particles in what is called the zig-zag patterns of Brownian motion because of their colliding with colleagues every now and then. But they are glued to the nuclei, and thus make them atoms. Local variations in the relative numbers of positrons and electrons give the atoms a net charge, which allows them to attract one another. That is one of the forces that binds atoms into molecules, although it is not the only one. Gravity helps too, but where gravity is static (does not change unless the relative distances of the atoms change), the positive and negative charges are influenced by other positive and negative charges in their neighborhood. Newton made his cradle with metal balls, so the charge effect would not be noticed since the charge would evenly distribute itself over the balls in total. Had he made them of some material that is non-conductive, then the balls would have reacted like the point charges in Coulomb's Law, which would mean they would still be able to interact, but the balls might be attracted, or repelled, given their positive and negative relative charges, thus complicating the experiment (keep it simple, right?)...

Now since we believe we are living in a real world, these basic pieces of deep-seated knowledge are essential in forming our view of the world around us. Thus we are normally weary of strangers, since they still wear the cloak of negativity, instead of Harry's cloak of invisibility. Well, they might as well be invisible, if you notice how many humans do not talk to strangers on a train or bus. It takes a fair amount of positivity to throw off that cloak, and be open to anyone reaching out. Sure, once you know enough about a certain type of interaction, you may grow weary of it and decide you've had enough, but that is way more than an educated guess, or the type of default behavior we normally display to one another.

Mind you though, that very cautious approach is a direct consequence of the fact we humans have that check for integrity built into our systems. A nice hint at this self test is made in the movie Innerspace from 1987: in it, Lt. Tuck Pendleton stands in front of a mirror, slaps himself in the face a few times, and exclaims: "The Tuck Pendleton machine: zero defects." Of course sometimes we must take outside advice regarding possible defects, but that will only happen once we trust the other 'system' to be truthful to us, and correct. If my blood count is low and my doctor tells me about it (like she did last week), I won't rest until she's shown me the relevant blood counts of the past period, which make me concur with her advice... ;-)

In the end it is just what we believe, whether or not we err on the positive or negative side: hypochondriacs will disregard the professional opinion of their physicians if they tell them nothing is wrong with them, and the other extreme are guys like me who never believe they are ill, but still have to succumb to a few physical or psychological dis-eases over time....

Isaac Asimov described this same principle in his three laws of robotics (Appendix B), since his robots are designed to have embedded in their hardware or hard-wired programming the prime directives to protect (human[1]) life. They can't even passively harm it, not even when ordered to do so by another human. This means they must treat the world around them as inherently negative, and protect humans from that. If only humans would treat each other this way.....

Inherent in Asimov's three laws is the logical but also quite subconscious deduction that androids need to be able to know what harms humans and androids, in order to be able to comply with the three laws.... The fact such knowledge allows them to do either good or bad with it is what scares most people! But the laws are the failsafe to that open possibility.

1 ***H***olographic ***U***niversal ***M***atter ***A***dapting to ***N***ature....

This document is Public Domain, and may be used by anyone for any purpose, except patenting....

More Input!....

While opposites attract, the design effort is not a real matter of opposites: it is a game of Feedback, presumably across the entire circle of creators, users, abusers (www.doesitblend.net), and what not. This process (depicted on the right) is very much present in any process of Evolution. So let's go through the steps in order to show this in a clear and unambiguous way:

The Source is a System which distributes certain things, which can be very diverse in nature: imagine for instance a manufacturing plant making a given product, or a software company selling a certain program. Even shares of a company follow that same pattern: a company distributes them, and the buying and selling of the stock follows the cycle of feedback on the right to a tee! Or a well might 'produce' water, like an apple tree grows apples for us...

So yes, when Apple announces the release of the iPhone 5, it is obvious to everyone that the reception of it by the customers will be positive. But the fact that they cannot produce enough of them to fulfil the total demand will swing the evaluation of the product to a lesser value, which soon became clear from the value of Apple stock... Please also note the symmetry in the feedback loop: the Consumer's Preferences are like the Emissions of the Source, and the Improvements are the Source's Reception (or reaction) to the perceived preferences. That is why manufacturers are fanatically fishing for customer preferences and review info, in order to perfect their products!

So yes, the Evaluation of the product by the Consumers leads to them exposing their Preferences in not just one but possibly many Feedback loops:

1. Lack of products increases the need for production capacity.
2. Quality of products relates to the popularity of the product.
3. The Design / Preference tradeoffs determine popularity as well.
4. Problems with a product lead to bug reports and new releases, a feedback cycle in itself.
5. Reviews by customers and or professional reviewers lead to design changes, a cycle too.
6. A product becoming obsolete feeds the need for a newer product with better features....

In a similar manner, this process is most obvious in software design, since there too the cycle of feedback is the main drive for better quality in the software product:

1. A company releases software, having two types of 'defects': bugs & possible improvements.
2. The software is often fitted with extra functions to automatically feed back problems to the creators. Partaking in this process is a decision of the user for now, but that might change...
3. New updates and releases are often automatically distributed, as the user allows this. Will we reach a point where this is considered obvious to the users, and not worthy of their conscious attention?
4. The above points drive the Feedback, which allows the software too evolve more rapidly...

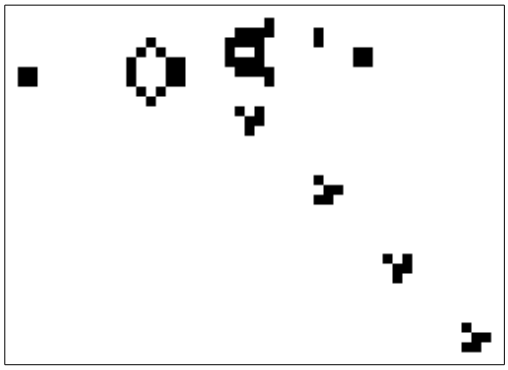
This document is Public Domain, and may be used by anyone for any purpose, except patenting....

Basically, the System Engineering and Design Architecture is nothing but an attempt at making the subconscious concept we have of the structure of Systems a conscious concept in our minds. It is not new, but merely aimed at expressing that which we know is right for the concept at hand. We will turn the System inside out though, and view it from there. It is not so much a matter of responsibility that is assigned to certain systems, but much more a matter of a System assuming the responsibility for a given challenge, based on what 'talents' it already has. In this aspect, it becomes much more like us, who use our talents to pursue our aims, and to shoulder our responsibilities as we perceive them.

Let me give an example of what I mean: the software in you cell phone may have a Contact List App, and a Phone App. Now the Phone App could keep a list of contacts for itself, but since its Environment holds something that already does that, it is more effective to just define an Interface to the Contacts App, and get the information from there. Likewise, the Alarm Clock App and the Calendar App might be a closely symbiotic couple, where both have their events to process, but they also both use the more basic Notifier App to Notify the user of these events as they occur. As a last nasty one, the System Settings App would just enumerate all Settings Interfaces of the Apps in the bigger system called 'Cell Phone', and present them in a consistent manner to the user. Now since 'System Settings' is the App to present all settings info to the user, there has to be a connection between any App and the Settings App. But which would be the leader in this case? Is there really any preference to be expressed in this case? Let's look at the possible relationships between these objects....

In effect, the diagram on the right shows two triangles in which the various aspects of System Relations are usually expressed. The upside-down triangle denotes the direction of the information flow, and can be seen as Independent, Unilateral, and Bilateral. The right-side-up triangle has the parts Calling, Registering and Symbiotic. A caller simply has to know another System in order to call it, but the Registering System goes one step further: It registers with the other system, in order to allow it to send certain messages that might be of use, or in order to simply let the other system know it is available. The Symbiotic relationship is one where mutual Registration is done, so both Systems can keep tabs on each other. Yup, marriage exists even here....

What we did not observe in this diagram, is the containing relationship. But then again, the containment of one system inside a larger one is hierarchical rather than lateral. If anything, one might define it to be symbiotic: does not a tree totally overwhelmed by a climbing plant look more like one thing rather than two? Likewise, the heart is contained inside the human, but neither can live without the other....



Truly independent systems do not exist, simply because they need some Environment to live in (we call the infinite outer system God, Allah, the Incredible Machine,

This document is Public Domain, and may be used by anyone for any purpose, except patenting....

Force, Source, **G**rand **O**verall **D**esign, etc.) Also, they need some System to hand them the link to the other systems they would like to call, but once they do they can still remain independent: it is like the so-called cannons in Conway's Game of Life (see previous page): they exist, and just fire in a specific direction, without ever being changed, unless another system collides with them. But the moment they require some effect to result from their 'bullets', they become relative instead of independent. Also, we Humans cannot be truly Independent: even freed of all links to fellow Humans, we are bound by Gravity to this planet we call 'Home'....

From this however, we can deduce the characteristics of ourselves as Systems. Now that is not quite the topic of this book, but I feel it will enhance our (and more importantly my) concept of 'a System' before we really dive into the Engineering and Design process.

Our Environment (as was just pointed out) is the Earth. We are bound to it by the most grave of forces, called Gravity. And that is not just in one aspect: even if we are able to defy physical gravity, we still care about what happens to our planet, which can be called gravity in a psychological context. Likewise, that same gravity is related to our selves, since we see anything first of all as a first person event. So in any view of ourselves as a System we can already discern three Environments in which we live: Personal, Psychological and Physical. Actually, that is just the threesome I picked, and it may very well be different for you!

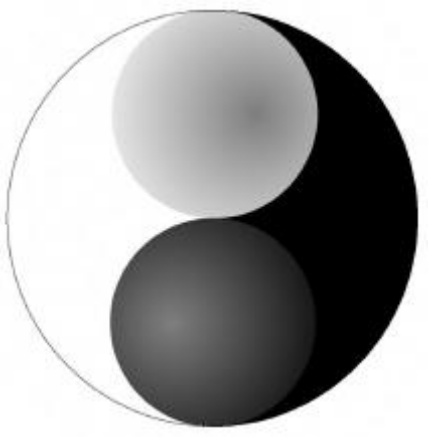
Now personally, we may see Matter and Energy as two concepts, of which one is made up of the other. Einstein's formula ***E=mc^2*** made it clear that Energy is condensed into matter, which makes Energy the base material of this reality. Purely theoretical of course, the formula only shows that both are the same, and we can take either end to be the base for the other! Inside the material and the energetic we see ourselves living, and interacting. The normal center of this we call Society (Common Consensus Reality I call it), although sometimes there aren't that many social aspects to it..... In the end though, it is our personal belief system that shows us how the parts of this puzzle fit together, if we are at all interested in the answer to that question, that is....

So what we do see here clearly, is that any system can be part of many surrounding systems, that clearly do not all have well-defined relationships to one another. Likewise, the Interfaces we developed are also aimed at various fellow systems in one of our Environments: to interact socially we might use Facebook, but to interact with our pets we might require play toys or treats (some even use threats!). Now the Interfaces of a System are kinda like the leash we humans tend to use to keep our pets from straying: regardless which environment they enter, we still have a way of getting to them, once they are lost from sight. And the leash is bilateral: a dog like a Rottweiler can pull you just as hard as you can pull it, or even more!

I don't know about you, but I tend to be somewhat cautious of making lasting connections. Call it a side-effect of my aim to stay balanced in the middle like the Buddha came to the same conclusion. Still though, even like he sat under a Bodhi tree for eight years, I have depended on my 'Body tree' to keep me safe. And of course it would be too obvious to point out that the tree is a concept of Balance: it has as many 'branches' underground as it has in the sky!

This document is Public Domain, and may be used by anyone for any purpose, except patenting....

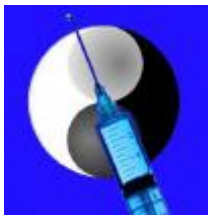
System Conception



From here on in, we will parallel a few aspects: on the one hand we are going to make clear how it is thought Systems naturally develop. On the other hand, we are going to try and apply this concept to the development of a software system to solve the problem of problem solving or evolving itself. The example should then give us a fair idea of just how to apply the theory, and reach a viable end result. In order to do this, we are first going back to the ancient oriental concept that already showed us the essence of a System: it has two aspects, namely how it reacts to the Environment, and how the environment reacts to it. The ancient Eastern civilizations showed this aspect of Life in the Yin and Yang symbol.

Now development of Systems in my humble opinion is a process much like the graphical evolution of a Yin and Yang to a full-blown SevenSphere. It is a process much like cell division, which most of us have seen happening on television, or other devices that show scientific fact.

1. The start of a system comes with the recognition of its being needed or wanted in some way. An imaginary System is nothing more than an empty space in the womb[2] of the future users, passed on to the group of developers who take it upon themselves to create the system. Whether the desire is seen as a problem or a possible solution says a lot about how it is eventually going to be conceived. In this case we want to make a system to solve for us the challenge of systems design and engineering.



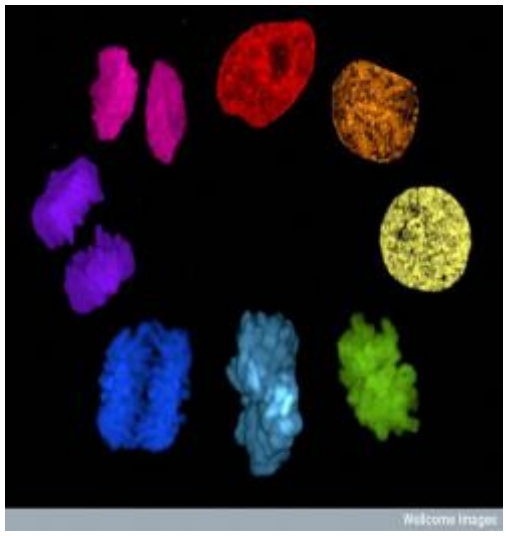
2. We are going to inject an idea into that central core, which should eventually deliver us an ideal system to bring about the birth of a desire for something new, something unheard of in our environment. But with our idea comes the initial interface of our 'solution', the one that it has towards the world we conceive it in, for all is Balance. Please note, that the interface may be formed as we want it to be, without any thought to what is eventually going to be needed behind the scenes to make it all work. Like Einstein said: *"I want to know God's thoughts, the rest are details!"*. Here we want to see creation at its best, not fear of complexity. Of course this does not apply to the interface itself: this should be intuitive and useful rather than a work of complexity, for simple interfaces breed simple systems, which do not require much learning in order to use them. It is like riding a bike vs. Flying a 747: the latter requires years of practice, and the other only basic (and mostly subconscious) understanding of the laws of equilibrium....

3. That in turn is immediately followed by our attention to define this interface as we want it to be, which as common saying has it: "It is all about appearances." We focus on the outside of the system first, like an artist molds the clay: he may not care which clay is where on the inside, as long as the outer form is to his or her liking.... *(see 'Nurture the Planet', on the right, by Harry Matti Hukkinen).* In this same manner I got a 9 out of 10 for my exam project, even though it didn't work! The interface was good, and the presentation also, so the one defect I still had to fix was skipped...

2 ***W***anting ***O***ne ***M***ore ***B***aby (or Beauty...which may be a person or thing)

This document is Public Domain, and may be used by anyone for any purpose, except patenting....

4. Of course clay forms are simple, as they come: most systems are more complex, although they follow the basic simple strategies outlined in Appendix A. Thus, in order to create something more to our liking, we start by applying the rules from Appendix A to our design: if we only look at what we have now, the simplest representation would be one system with one interface, the boundary of the form and its environment. But in our symbiotic approach, the designing of a system for problem solving, we will need a little bit more. Thus, we design an interface to have a certain effect upon the environment: if we want it to be controlled by humans, then the interface must match the perception, comprehension and manipulation skills of the average human, if it is to be a succesful system. So instead of a Yin and Yang, where the interface is a curvy line, we start by seeing our challenge as a newly developing child, right after its first cell division....



5. Notice how the cycles of cell division quite neatly match the idea of the SevenSphere: although this photographed and colored-in collection of images from http://images.wellcome.ac.uk shows the eight states of cell division, we see that only six of them are actual single cell images. The remaining two are already a set of two, with identical DNA.
6. Likewise, our conceived idea of a generic system engineering and design architecture will divide into the system and its environment, both seen as finite entities, and thus as something that is solvable. Note also how we are going to base the system itself on our most simple notion of how the outside world would look to the system, and thus what the inside structure of the system itself is going to be: remember how we all have a set of knowledge in our minds that neatly reflects what we ***think or believe*** the world outside looks like? Based on that, we meet the Environment head on, in order to make our observed experience of the System out there as perfect as possible!
7. So rather than diving into the definition of the System, we will start with the one thing that all Systems have in common: the Interfaces they have with other Systems! And don't act like you don't know what I'm talking about: all of us are in minute to minute or even second to second contact with interfaces, whether we look at the sidewalk while walking to avoid stepping on ants or snails, or we are chatting or otherwise engaged with our cell phones, tablets, laptops, and other devices of mass distraction!
8. *Side note: 'weapons of mass destruction' kill mass, thus making things lighter..... (Thanks to Faithless for that making me have that realization)*

This document is Public Domain, and may be used by anyone for any purpose, except patenting....

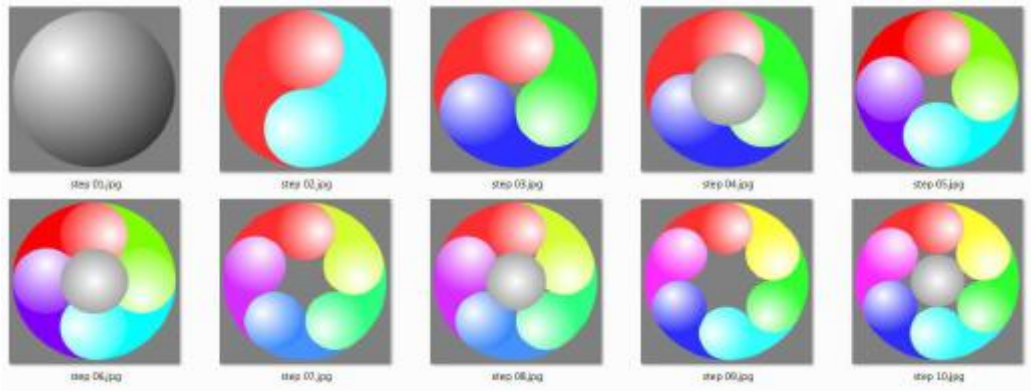
System Development by Analogy

This chapter shows how Systems develop by giving you an example with one of Nature's most basic forms, the circle or sphere (depending on how much of it you can envision in your mind). I will show how the full sequence of the images I drew in 2001 eventually allows a system to fully understand its environment, and thus will be able to deal with the sum total of it.

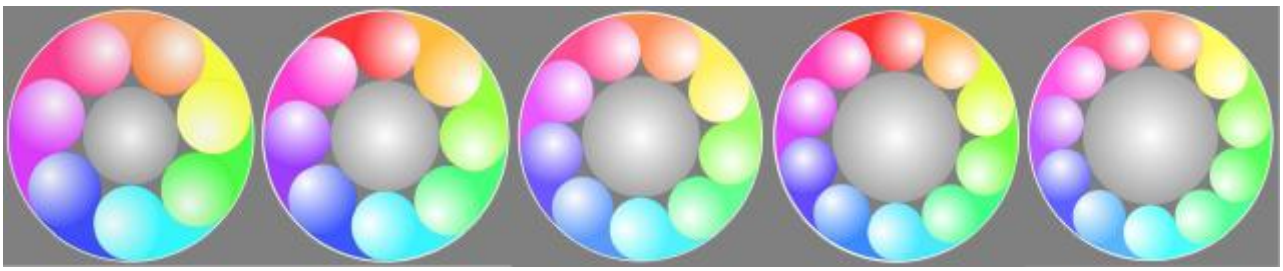
We will simply start with circles first, because most people have no problem with envisioning them in their minds, and because they are the most simple systems imaginable: their two variables are diameter & dimension (D^2 for a circle, or D^3 for a sphere). Close packing shows that regardless how many circles we pack in a given area, there is always a part of the total area that is outside of all the circles that cover it. No matter how many smaller circles we pack in between the bigger ones, the total area will never be completely covered, unless the number of circles becomes Infinite! For the more visually acute minds, it is easy to see how the same applies to spheres packed in a volume, although the space left has one more degree of freedom....

But, back to the basics. The final outcome of my drawing exercises at the start of the Millennium was twofold:

1. The SevenSphere showed me a distinct boundary in the logical development of the sequence based on Yin and Yang, in that it was only possible to maintain the same diameter of circles in the evolution of the series until there was one sphere at the center and exactly six surrounding it. From there onwards, the diameter of the center sphere would have to be relatively larger with regard to the surrounding satellites, and the total diameter of the form itself.
2. The evolution of the first sequence took exactly ten steps, as can be seen below. Maybe that is why the base-10 numerical system has been used in the western world until now...

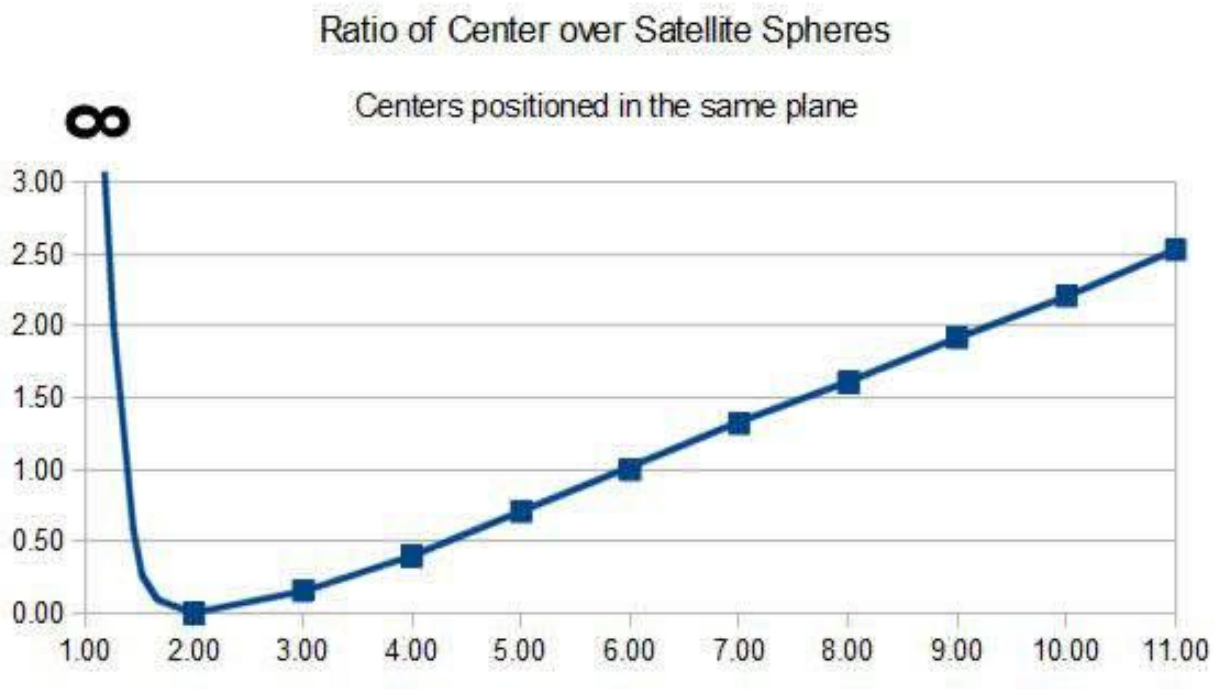


Back then I didn't realize that there were exactly ten steps from oneness to SevenSphere, but I did know that further evolutionary steps would make distinct changes in the evolution of the spheres: the center sphere would have to become relatively larger with every satellite sphere added, and the satellite spheres, if all given the same diameter would have to become relatively smaller. Let me see if I can draw you a picture on what's left of this page:



This document is Public Domain, and may be used by anyone for any purpose, except patenting....

You get the picture, I presume... As the number of satellites grows, the ratio of the diameter of the center sphere over the diameter of the satellites grows. Based on the first five images I drew, I have established that the ratio grows a factor of about 1.3 with every next phase. Apparently, even inner space has three dimensions, but they are on the opposite side of the decimal point. At least that's what came out of the first five images. After also measuring the phases from the first sequence that have holes in them, I come to the next graph:



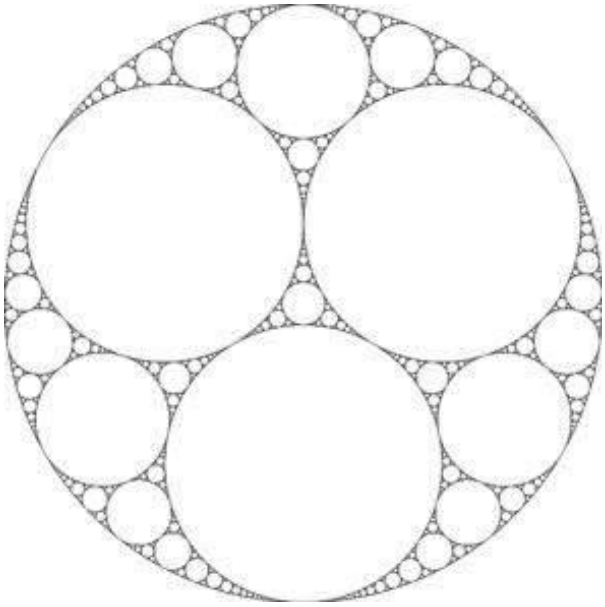
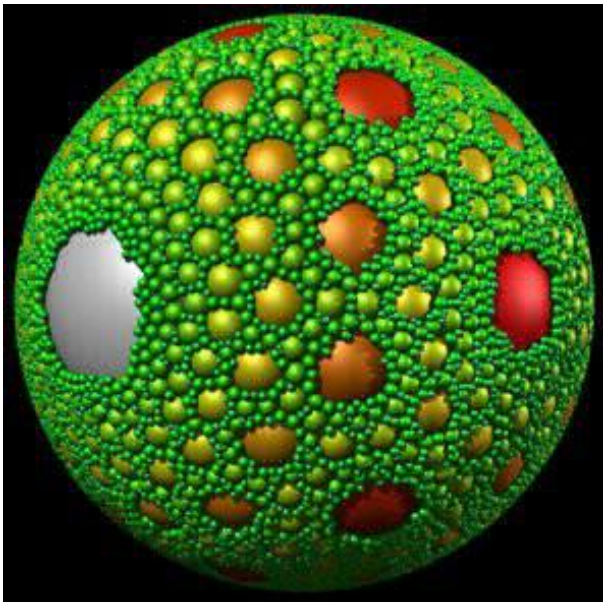
The significant part here is that the first image, the fully grey sphere of Oneness is in fact also the ultimate end of the developing series, where the center sphere or system has been able to fully absorb the space of the satellites around it: both ends of this graph will eventually reach Infinity.

We all learn all the time: for Example, the images I drew back in 2001 were an extension of Apollonian gaskets, as I just learned from a relevant web page which referred me back to Wikipedia (see image on next page). The main difference is, that I also drew circles within circles, because that seemed right at the time. When asked to explain why I did that, my enhanced insight from further experience would now make me explain it in my new system-oriented approach: If we consider the Apollonian gasket as a set of systems (or the sphere for the visually more evolved humans) we see that these images can be seen as a number of interacting systems, because circles and spheres are also systems, but of extreme simplicity. From that we can also see they have interfaces, where the circles or spheres touch. The learning process is nothing else but a system creating an image of the other spheres in its environment inside itself, but it can only do so via the interfaces it has with its direct neighbors (where the circles or spheres touch). The bigger systems may be connected to many surrounding smaller systems, but they have no direct insight in the connections those systems have to one another. Sure, they can query their surrounding systems for info about such relationships, but who is to tell them that those systems are telling them the right thing? The big system may think it is in a stable position, while all the time it is being moved around by the smaller ones creating a vacuum in front of it, and a buildup of pressure behind it. So yes, the masses do have power over the bigger systems, since the combined effort of the smaller systems is spread in such a way, that the bigger ones basically get the same information from

This document is Public Domain, and may be used by anyone for any purpose, except patenting....

their surrounding systems, and cannot make heads or tails from it: just consider the rightmost sphere in the image above, where the surrounding colored spheres basically feed the center sphere all colors of the rainbow, but are in fact all together feeding in the same overall info (white light). Now in the 2D example below, the smaller circles may only form interfaces outside the bigger ones, and thus can be virtually cut off from their smaller neighbors because of them ending in the ever sharper spikes where only a relatively tiny interface is possible.

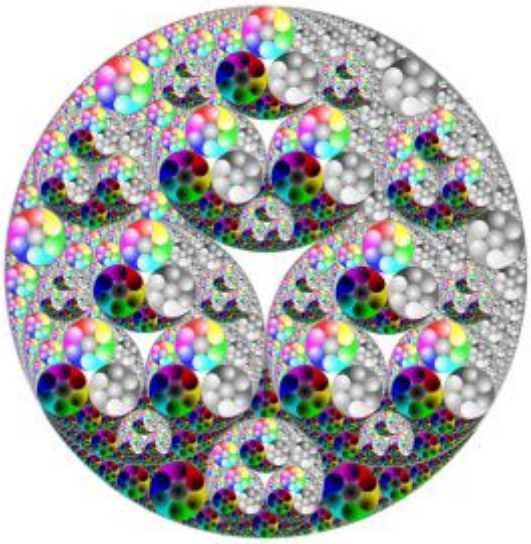
I lack the mathematical skills to prove my next statement to you, but just consider the image on the right below next: we can clearly see, that no true separations of the smaller green spheres exist. The extra dimension gives the smaller spheres the freedom to evade such separating areas, and keep contact around them with one another. Even more so, if the image was one big sphere and numerous surrounding green ones, the sum total would be green all over, unless the green ones decide to let the grey one show its skin.

Apollonian gasket (2D circle packing) (reproduced from *Wikipedia*)	**Apollonian spheres (3D sphere packing)** (reproduced from *Wikipedia*)
	

Pretty deep material, right? Now imagine the big grey one showing in the image on the right. It is basically an internal system of the total system, exposed to the outside. The next logical step to 'fix' this would be to make more growth in green cells, whose division strategies will eventually cover the wound just like they do in humans. The only difference here is that our small systems are red blood cells, and the wound is basically anything from a cat's scratch to a broken bone.

The bigger system inside may then never know it has been completely surrounded by green little spheres (or blue Smurfs, just like Gargamel) Yes, Peyo explained this in a similar manner, that is enjoyable for both kids and adults... But where he explained it to the subconscious, I'm targeting the conscious mind.

This document is Public Domain, and may be used by anyone for any purpose, except patenting....



Back to the circles within circles though: this is where the persistence of vision or imagination for most humans does not measure up to the challenge: if smaller circles would also be within the bigger ones, the circumference of the bigger circles will be almost invisible to the common observer. Just check the image on the right, if it weren't for the triangle in the middle you might never have seen the three major circles.... And keep in mind, that beyond the three dimensions of space and the one of time, we have countless other dimensions to keep track of: most of us juggle Home, Work, Education, and various other concerns, distinctions we too make for our own peace of mind: you wouldn't want to have the work piece take up part of the home piece, or even the home peace.....

But then again, they are only artificial boundaries: we apply them, because we need to be able to handle them all: Divide and Conquer wasn't discovered for nothing. And of course the boundaries aren't always that clear: the moment I come into a healthy sum of money, the system called work may have to deal with a loss of personnel knowhow, which of course I will try to transfer to colleagues during my last months there. That is basically the bigger system called 'me' deciding it doesn't need a certain subsystem anymore, and discarding it. But also note that that means another subsystem of mine (wanting to win the lottery) has suddenly become realized and thus made the work part obsolete. But also note that the employee I am is only seen as a minute subsystems of the company, which will not be shaken or stirred if the employee leaves: they have many like that left....

Still, we may be part of numerous systems like for instance:

1. Partner of a nice lady or guy
2. Father or mother to any number of lovely kids
3. Member of a family unit
4. Self-employed business person
5. Employee of a company
6. Citizen of a country
7. Member of a church
8. Supporter of a local soccer team
9. Buddy to several colleagues
10. Gamer on WarCraft or something similar
11. You name a few other ones....

Too many to see it clearly right? Point is, since all are thought forms, they do not impinge on one another inside our minds: unlike the image on the right here, subsystems in the world of ideas do not take up space, and as such do not need to avoid one another in that way. And they can always be made to work

This document is Public Domain, and may be used by anyone for any purpose, except patenting....

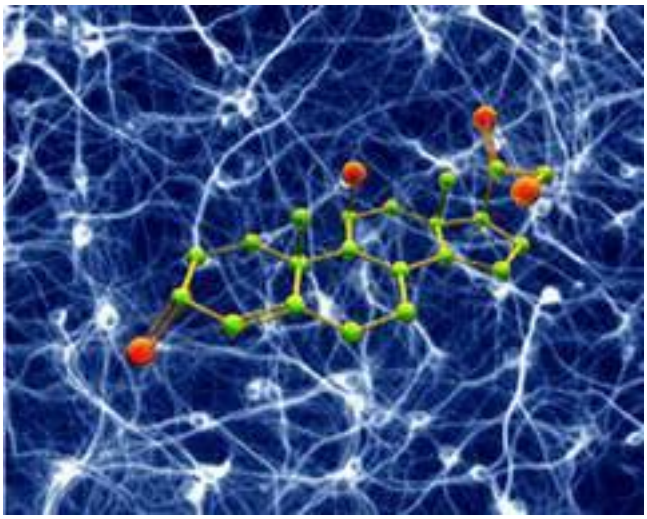
again by escalating to the next higher level in order to fix things. Like I would ask to speak to my CEO if my boss gives me trouble, and it always is just like Einstein said:

"A problem can only be solved from a higher awareness than the one that detected it"

So in order for me to understand the more grand circles of the obvious, my subconscious made me into a sort of a recluse: I don't normally go out to experience the everyday life unless there is a solid reason to do so, and I am not a member of many bigger groups except the global culture we call Humanity. Father to two lovely daughters alright, but no partner at the moment because writing is relatively more important.

As I reread this my system plays me Sting: "If I ever lose my faith in you", which according to the lyrics is all about that feeling of not belonging anywhere but still being at the center of it all. Such 'coincidences' are way too coincidental to ever make me lose my faith in You!

On the other hand, I do go out of my way to assist anyone needing help, because of this peculiar desire of mine to see everyone happy. And the weird part is, since I radiate this to the world outside, I also see it reflected back to me: there may be 'systems' out there busy to try and draw me into their bigger groups, but there are just as many more free radicals out there to assist in any way they can simply because that is the total of the image of the system topography I uncovered from my experiences over the past 49 years... (as if time matters)



Now let's just view the spheres in spheres thing another way: since the spheres are the center concept here, and the connections between them signify the interfaces that relate spheres to one another, do you have any idea what this whole cluster of tightly packed interacting spheres looks like? If we see the spheres as point charges, it is quite like the centers of the spheres are neurons, and the connecting interfaces are the dendrites and axons that send the signals from one neuron to another. This way, the set of neurons inside a system may physically be totally differently packed, but they'll fire[3] according to the associations we make on a regular basis with regard to those pieces of outside information! Even more so, since the sets for the various senses are in certain areas of the brain, and the acting together of these input devices makes sure neurons are firing together, they also wire together: if you hear the music it isn't hard to actually know the lyrics, even if the music itself is Karaoke. Likewise, your learning of various languages makes sure the neurons for the various words in different languages fire together, along with the neurons that only signify the sounds languages have in common. So the idea that any language has a finite set of three letter 'syllables' is just common sense, because the neurons that associate with the combinations fire in sequence, and when activated often enough in that sequence, also wire up to a neuron that signifies the word. Now neurons are just simple little systems, that receive charges and eventually pass them on to other neurons. They don't really care which language you are speaking at the time, as can be simply witnessed from the amount or interlingual 'pollution' that is apparent in todays media and communications between people. I've often heard my daughters complain that they often just 'miss the word' they are intent on speaking in Dutch, and end up with the English term instead, simply because it is foremost in their mind.

3 http://www.topnews.in/health/brain-neurons-trigger-chain-reaction-cascade-falling-dominos-28921

This document is Public Domain, and may be used by anyone for any purpose, except patenting....

And then I'd like to throw in this thing called Attachment: In that concept we are like the neurons, depending on our inputs from certain other neurons, and using yet others to fire at. If you consider Darwin's theory of Evolution, then the best adapters are the winners: not the necessarily stronger ones, as common misinterpretation has it, but the faster adapters! In my view it is easy to see that those neurons connected to less inputs are not bound to fire as frequently as some of their neighbors that depend on their sources rather than deciding when to fire themselves. We know so little of the human brain (talking about people in general here), or even the brains of little beings like ants, that basically, our knowledge is more a belief than actual knowledge when we talk about what we think. In my humble opinion though, I figure that thinking and feeling (or believing) is more like accessing the singular human intelligence and consciousness in the first case, and the accessing of the common intelligence and consciousness in the second case. By accessing more of our feelings and less of our thoughts, we become more One with the sum total of knowledge available, and thus become better at adapting to Nature.

In this way, attachment to certain thoughts may actually hold us captive, when letting go of them in favor of our feelings will allow us to change our minds in a more creative way, thus becoming better adapters, like many animals already are....

And what about our other threat in evolution, the machines and their parts? If you feed the words 'best adapters in evolution theory' to Googles Image search, you get images of machined parts mostly. Why? Simply because we allow machines to use ***our*** intelligence and consciousness to adapt to their environment! We create them, we adapt them, and we drive their evolution forward by our love of them! Can there be any outcome other than where machines will eventually outperform man in intelligence and conciousness? I think not, and I feel that that is not a problem. Movies may have us think that machines might take over, but the common consciousness available to us through our feelings will tell us the real outcome, and it is their common consciousness just as well...

This document is Public Domain, and may be used by anyone for any purpose, except patenting....

So our minds map the outside environment neatly into various areas, where the areas overlap in the brain, in very entangled ways. The overlaps are there, but we often disregard them because we know from context that a disambiguation is in order. Still though, the ambiguities in our languages and ideas aren't there for nothing: lots of concepts have formed as ambiguities, because the concept we were talking about at the time was closest in concept to something we already knew. Like we call a computer mouse a mouse because it was grey and had a long tail.... In similar cases the ambiguity isn't that obvious, but keep in mind that often they may instill miscommunication because the receiving partner in the information transfer simply has not yet linked the term to the right context, and wrongly assumes it is linked to something he or she does know about. Let me give an example that I myself often run into: although I usually think in very broad lines, those who interact with me sometimes assume I also know about certain detailed stuff. And their referring to detailed terms has me linking it to global or even cosmological concepts, and then reacting completely incorrectly! I counter such mishaps by adding additional anchors to out there, which normal humans don't pay attention to: things that are too coincidental to actually coincide just for no reason! I figure if I pay enough attention to these alternate signals, it is just a second web to keep me in place where I am to be: subjecting to Common Consensus Reality, while writing about concepts that are so fantastic I can only write about them because I truly believe in that reality as well! It wasn't for nothing that a psychic told me to write on paper for my own good: he didn't mean that I should write my books in handwriting, just my notes: they will mean nothing to an uninitiated mind, but they form priceless clues from my subconscious to my conscious mind: I know when to write them down, even if they make no sense to me at that very moment: writing on the sides of trucks and buildings, which I can later expand into consciousness by looking them over and writing about what comes to my conscious mind from my subconscious. Much like Da Vinci's notebook, which contained notes very few of his fellow men and women at the time could understand, let alone conceive...

Let's just expand on that 'higher awareness' for a moment: we've already seen that any System has an Environment in which it lives, and which eventually will give it feedbacks of varying size, which may traverse any dimension we perceive in the world around us: where before we learned to trust local systems first, and those further away later, the default approach is shifting: we get signals from all over the globe, and determine which to discard and which to believe. We may be able to short-circuit such connections literally, by inviting those web friends to our reality. And where Occam's razor implies that the simplest explanation ***tends*** to be the best, it does leave open the possibility that more complex explanations are in fact what really happens!

Let me try to explain the concept of dimensionality in a current everyday example: we've all seen images of those laser light shows in discotheques, right? Now imagine a few of these lasers on the right being controlled in such a way that you end up with a virtual 'floor' of green light, that completely surrounds you at hip level, with you standing there in the middle. The green floor is a 2D world, where flatlanders live. What do they see of you, given Occam's razor? Yup, one object or being, which they can walk around, so they know it has a finite 'size' in their 2D view. Now if you had your arms tightly against your side, and moved them outwards, what would the flatlanders see? Simple: they would see two distinct objects emerge or separate from the first object. They even might consider it to be a living thing, since it shows behavior more complex than they attribute to lifeless objects (perhaps a flatlander

This document is Public Domain, and may be used by anyone for any purpose, except patenting....

having twins?). If you then lift your arms out of the plane, it would look to them like the two secondary objects or beings just vanished from their world (or died)! They have no knowledge that on a higher dimension, those three objects are in fact one being! The flatlanders might consider them returned again if you lowered your arms afterward, because of their previous experience. But if your arms had been raised at first, and you stuck them into the plane without warning, the flatlanders would see two suddenly appearing objects or beings, seemingly moving independently of the bigger one that was already there. Following Occam they will conclude that there is no connection between the first object and the outer two, and depending on the relative arrival of the last two, may not even figure there is a connection between their arrival. To them they are just three beings, instead of the one you call 'Me'.

So there's reincarnation for you in a nutshell: the moment we consider the idea that some of us may in fact have had earlier and later lifetimes as other intimately related beings, the thinking about it gets you all in a knot, because it is hard to see. Only once you see time for the non-issue it is, 'separate' lives can actually all take place in the same here and now, overlapping in just about any relationship you may envision: if you see a stunning family resemblance in a historic picture, that is just your connection through the All to every other one of the infinite subdivisions of the All we call beings. I myself was introduced to that by a striking resemblance to Nostradamus' drawn picture in the trilogy about the guy that Dolores Cannon wrote. It was based on her deep regression sessions with a subject who actually was a pupil of the 16th century seer who in 1555AD published the first issue of his famous Quatrains, small highly cryptic poems that were said to predict coming events. Her subject helped in making a pencil-drawn portrait of Nostradamus, that looked like it was my twin brother...

Psychiatrists may well call believers in reincarnation delusional, but it was one of them who pulled reincarnation out of the paranormal: Dr. Ian Stevenson documented over 3000 cases of children spontaneously remembering their former lives: he ran down the clues of the stories, and proved that the details were so accurate in most cases, that normal everyday humans will probably not be able to resist believing his proof. To show you I did not make this up, let's add this link I just found on the net, which also has an interview by someone who actually looked at Stevenson's approach with a skeptic's view.....:

http://reluctant-messenger.com/reincarnation-proof.htm

So believe what you will, in the end you will only find that which you believe. You can spend trillions of Euros building the Hadron Collider, and you will find those extremely small particles you're dying to prove exist! The one freedom that is there, is the Infinity of it all: since you don't know what you want to expect of these particles, they will exhibit behavior tempting you to look ever deeper into their mystery... (because that is your passion!)

This document is Public Domain, and may be used by anyone for any purpose, except patenting....

In this light I dare you to deny the possibility that in a higher (possibly infinite) dimension, we are all the same being. True, it is all in what you believe, but don't let me or anybody else try to make up your mind for you. No matter what we call it, that ultimate outer limits intelligence is there, and totally encompasses all our knowledge, us, and more. In the sphere analogy, it is both all the spheres and the spaces in between, many times over!

No matter what tool you picked in the above paragraph to assist you in dealing with your environment, consider that it might be most useful for you simply because you know how to handle it best. Like a juggler might have a preference for juggling a certain set of items, and being most proficient in it, the basic juggling skill is his real tool, which serves him for all of his juggling performances. It is his Swiss Army Knife, made of carbon (like us humans) so the detectors at the airport won't detect it! And the secondary tools can be anything he picks up after crossing the border, from an ordinary bunch of kids toys to a set of burning torches.

But of course, given the immenseness of the world around us, and the fact that without feedbacks we cannot determine what is right for us, most humans are downright cautious when it comes to the unknown. And if burnt once, they tend to be even more cautious. As a kid, I terrified my mother because despite her warning, I stepped towards the open oven door, and put both my little hands on it in order to feel what this thing was that she called 'Hot[4]!!!!'....

She could do nothing else but put ointment against burns on them, wrap them in bandages, and put me to bed. Nothing bad eventually came of it, but I guess it didn't really still my hunger for 'Hot' ;-)...

Since we need more info on the world around us, in our practices of Integrity we aren't always fool-proof, let's draw another SevenSphere, which might give some more insight into this thing called Integrity. Of course, to us humans it has more meanings, but we'll connect those dots later. First I'm going to explain this for a Generic System, because I do not think the book of Genesis got its name for nothing:

If our holy books tell us that God or any other deity made us in his or her likeness, and we see the similarity between ourselves and the various expressions of Nature also, then I guess the image shows us we are all in essence variations of that one which in my book would be called a '**Gene**ric **Sis**tem' (*alternate* spelling intended). Because let's be honest, the Bible itself talks of concepts that appear within a given environment, a void or nothing. But since that void itself already existed, we cannot simply call it creation and be done with it. Like we see for all lifeforms, the embryo or egg(s) are for a certain period imbedded in the parent, before being freed to inhabit the environment out there. But being the one infinite being we are, that concept is not a possibility for the totally infinite System that goes by different names: it can create, but only within its own being (since it has no 'outside', being infinite)....

*And just at that moment, my JetAudio player chose from 37888 tracks: "the real thing", and "**Can You Feel the Force?**" Don't you just love it when that happens? I DO!!!*

4 Hot in Dutch is 'Heet', not to be confused with 'Heet' in the context of "Hoe heet het?" or "What is it's name?" which was foremost in my child's mind.... (context sometimes sucks...)

This document is Public Domain, and may be used by anyone for any purpose, except patenting....

OK, a generic system, and its take on Integrity. You make not like what comes next, because my random choice of movies to watch as side inputs to the writing process has fed me nothing but serial killers these last few days...

“We kill the weak, so the strong survive!” says the bad guy in Cobra just now... What follows is a taunting game of provocation, where he challenges Cobra to arrest him and take him in. The Law will then judge him he says: “Because even I have rights!” Now I would probably try to do as he said, but not Stallone: “This is where the Law stops,... and I start!” And all that coming from a guy called Marion *(yes, both in Cobra and Demolition Man he is called that....)*

And then, out of the blue a new movie crossed my path, randomly chosen from a case containing exactly 444 DVD's belonging to my good friend and neighbor Paul. The Eye it is called, and it is all about a young lady who has been blind since age five. She gets a second cornea transplant, and it introduces her to the world which isn't quite what she expected. It is bound to thoroughly shake or stir my insight while creating the Integrity SevenSphere while it plays...

Now as the serial killer movies made obvious, the intention is often a crucial point in ones Integrity. Even they are running entirely on that intent, although their choices in that area are of course strongly disapproved of by normal everyday humans. So the example of our now sighted young lady getting used to the mixed signals she is getting from her extra sensory input channel (vision) may be more appropriate. First of all, she doubts the consistency of her inputs: she is seeing things that her other senses say are not there! Terrified by the sometimes violent nature of the visions, she doubts her integrity as a Human, not in the way that she has no good intentions (the outsiders view of this SevenSphere), but in the way that she considers herself to be not entirely sane (her inside view of the same SevenSphere). But her psychologist does not believe her when she tells him she is seeing things that happened elsewhere, or people who walk right through her as if she isn't there...

Now you may call this fiction, and even quite possibly horror as well, but any subset of the world around us may be used to illustrate stuff in order to help people understand other mechanisms. In this case, the lady's Integrity is based on the four things she has always depended on to make her *not* doubt herself:

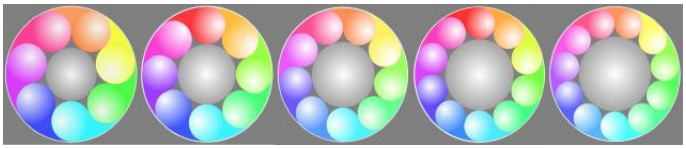
- Sensory Inputs
- Relevant Knowledge
- External Actions
- Feedbacks (*inputs recognized as effects of actions based on inputs and knowledge*)

With her sensory inputs radically enhanced, her mind cannot make the inputs match the relevant knowledge in her brain, because it has always gotten by without the extra vision. She clings to the only information that is making any sense to her, a number of stories about transplant organs actually also carrying vital information of the donor to their new body. Since the building blocks of the DNA are said to be the design schematics of the human body, that

This document is Public Domain, and may be used by anyone for any purpose, except patenting....

isn't difficult to believe at all. I'll even throw in what I said to a fellow train passenger just this afternoon: “My kids have turned out to already have the knowledge implanted that I had by the time they were conceived. Even the differences in their view of their environment can be attributed to their being born two years apart. Two years in which I myself had learned quite a few things about our world.”

Yeah, I know, this kind of talk doesn't go down too well during birthday parties: once a couple of people get into that kind of talk, the others mostly nod wearily to one another, and go on with their talks about work, kids, diseases, recent deaths and new cars, phones or televisions. Nothing wrong with that, but neither is it wrong to ponder such questions. Right and Wrong are but divisions of the One, and any division in itself creates growth, from one to two, and onwards until Infinity.....



But let's get back to the five images with the ever smaller satellite spheres. Do you know what it reminded me of just now on the trip home? A diagonal cross section of a soap bubble: a ring of soap molecules stretching to keep an amount of air captured. Well, as a cross section that won't work of course, but you can easily rotate it across any axis, and come up with a transparent sphere, with lots of colored little soap molecules enveloping it. And the lead into that idea? Just figure this: I saw a car I didn't recognize, and looked backward as I turned the corner, seeing it had the type designation CROSSFIRE all across its very nice behind. Then at the next traffic light I needed to stop for a CROSS COUNTRY car (Volvo I thought). Then, before working on the book I viewed a few TED talks, and one of the presenters use the word 'CROSSHAIRS'. And thus, with the CROSS on my mind, the SECTION came up quite easily....

Now I put to you, that given Occam's Razor, the circle and sphere are basically the simplest solution to expansion: with opposing forces, the stretching effect will simply make the sphere the smallest surface to envelope a given volume. In the image on the right it is somewhat distorted, because the internal pressure of the bubbles cancels out where they touch. Hence the flat surfaces. But you see quite clearly that the colored soap molecules of a cross section do look quite similar to the colored satellites on the fivefold image shown above. Mind you though, in this case we are talking about pressure stretching a set of soap bubbles to the maximum surrounding form possible, but the opposite can also be true: the surrounding satellites doing their utmost to keep a given vacuum from collapsing. In both cases, the failing of a certain number of them will make the balance disappear, in an explosion in the first case, or an implosion in the second case. Just think of balloons and old-fashioned television tubes as the two examples of such a collapse...

Now let's go from the small everyday space of soap bubbles to the huge vastness of space: Since space is a mere human distinction, I put it to you that the same mechanism of enveloping the largest possible volume with the smallest possible sphere is also that which holds the Universe (or even the Cosmos) together. I once read a book that said the Universe was made up of huge bubbles, which probably needed a certain amount of cold dark matter, and all the stars, galaxies and superclusters and stuff are on the separating planes between these

This document is Public Domain, and may be used by anyone for any purpose, except patenting....

bubbles. To relate it to current insights cost just one Google search, to come up with this:

http://csep10.phys.utk.edu/astr162/lect/gclusters/soap.html

The original book and even this page said they couldn't give a plausible mechanism for the formation of the dark matter that was supposed to be there, but there is a simplist approach: the Big Bang is generally believed (or even proven) to have had an entirely homogenic (and extremely high) density. But you would probably agree with me that even the slightest local variation in density would cause the matter there to be more attracted towards one another than the surrounding matter a little further off. And since the law of Gravity has the force decrementing exponentially, this effect will be quite pronounced. So the very first minute instabilities in local density soon became the bigger concentrations of mass, which were kept apart by the expansion of the entire Cosmos. Now give the fact that between any two masses there is a plane where the gravitational forces of both cancel each other out (the logical 3D extension of the Lagrangian point is a plane), it is easy to see how the remaining matter eventually became trapped there, because their mutual attraction far outperformed the balanced out effect of the dark matter. So we end up with a huge reality of soap bubble like structures, where physical gravity and mass is concerned. And this gravity field between the super massive black holes that are called cold dark matter is even like the soap bubbles in more than one way: where the sheet between to soap bubbles is one layer of water suspended between two layers of soap molecules, the three layers in space are one layer of local gravity caught between two layers of Langrangian planes, that exist between the smaller local systems and the far-out super-condensed cold dark matter.

But now comes the leap of faith: Can you see how this same model is applicable for the interactions in the business market, the atmosphere around our planet, and the weather patterns? Companies that grew the first became the greatest, and nowadays are hardly ever noticed: Paramount features prominently on our screens, as do many production companies with the most fantastic names. Likewise, MGM and many others are part of the News Corporation, which may not have many direct customers, but plenty of leverage because their daughters and sons do produce consumer goods. Paramount is owned by Gulf + Western, as the small print on the movie screen says. Originally a company that started making stamped metal car bumpers, they diversified and added more 'mass' to their company, by for instance acquiring Paramount Pictures and SEGA. Their combined employees would be the soap molecules that held the company together, as they exist on the boundary between home and work, slipping in and out every day. Even Star Trek and Mission Impossible were their products. So even here as the cold dark matter they are hardly noticeable, and the light is generated at the layers in between, as the stars are in our view of the Cosmos.

Another set of soap bubbles are the customers of their various companies, who basically bring in the money that keeps the company together. They too are on the surface, even more than the employees, since they are not bound by a contract to show up and buy stuff. Of course that is what companies would love to change, hence the communications setup nowadays, where we get drawn into contracts by selling us free phones and tablets, and charging us for the transferring of information, a right that humans have always had. Yes, you are free to say almost anything, and free to gather almost any info, but getting it and sending it out are other concerns. But I must say, it does get easier if you don't try to get anyone to give anything for it: my books don't sell, but they download like a Bat out of Hell: 25000+ downloads from my own domain in the last half year, and I don't know how many sites have copies that can also be downloaded. In that aspect, I've become quite the soap bubble myself..... ;-)

This document is Public Domain, and may be used by anyone for any purpose, except patenting....

In your Face!

OK, so Interfaces it is! We all know them, because we have family, friends, pets, toys, and what not. Interfaces are most known because of their end points, which each of the parties involved actually call 'their' interface to the other system or entity involved. As such, there must be something in between that will perform any translations needed. Notice also how there may actually be a whole string of interfaces and translations present between the interface from one system to the next: our chat session to a close friend in India may imply usage of both keyboard, microphone, and webcam, each of which are translated into the corresponding digital information streams needed to get them to the other side. Once there, they get decoded again, so they can be presented to the user over there. To us users, it is an elementary interface, which we work with intuitively, without ever having to have a real know how of just how the system in between operates. All we need to know is what controls to interact with on the end points in order to get the system to do what we want. That works OK in case we are connected to a system or entity we know, but can we actually be sure about where the interface leads us? Is not an interface merely the end of a virtual Einstein-Rosen bridge, which may in fact have its other end anywhere in the Cosmos? Hence the rise in 'channeled' ET's....

This is the main reason interfaces have to be intuitive: if they are not, then using them will be a question to the user, rather than an exclamation mark: kinda like I once pushed a button which I figured at that time was not connected to anything: I pushed it out of my incessant urge to satisfy my 'Lust for Clarity' as a friend once formulated it.....

Also, there is the problem of "Where is it leading me to?". In this case, my work this morning has just changed my intended audience for this document: no longer a company-related effort, but rather a journey of systems discovery, related by an 'Old Shatterhand'[5] in order to help the youth get their grip on the world around them. As if they need any help ;-)

So, what do we know about any interface to another 'System'? Let's see:

1. It has an interface that is connected to something either inside us or outside: a stomach ache will tell us about an internal sense of wellbeing, or a '*mal*function' or '*dis*ease'. An outside interface involves any of our outside senses, or sensors. We may believe we have five, or maybe even many more, but in fact, the five have multiple functionality: where two hands are needed to normally ride a bike, only one is needed to send a text message to a friend, or hold a loved one... And touch senses physical contact as well as temperature.
2. But this signal we receive may have a different meaning depending on the Status Quo we find ourselves in: we may conclude that a stomach ache will probably have been caused by the copious meal we ate yesterday, and dismiss it without a second thought, just like we do with most spam on our computers. On the other hand though, the info coming in could have us look at it twice, as if it is somehow relevant to our current process: in the same manner I just went image hunting for a new cover on this book, only to find out that my favorite Internet market sent me a selection of their collection, which they claimed was specific for me: it only showed me four articles, which happened to be all keyboards! Well thank you, but my current keyboard will do just fine. On the other hand though, could it be a hint to stop the hunt for the cover in favor of the current writing streak I find myself in?

5 Children's books from my youth, which featured an All-American cowboy and his Indian sidekick.

This document is Public Domain, and may be used by anyone for any purpose, except patenting....

3. So yes, we decide the meaning of the incoming messages, just like any system will have to. So far, software systems have relied on our programmers and designers to do the work of defining all the interfaces, but this is likely to change in the coming times as we grow to better understand the essence of interface development. Point in fact is we decide the meaning based on our current knowledge, which is never complete: any signal coming in could potentially radically alter our perception of any given system: like most people might have second thoughts about relationship sites (too much clutter), my eldest daughter found her mate on one, and she has been quite happy with him for two years now! So that would alter my perception of these sites and the chances of finding a mate through any virtual channel in a profound way.... (And cause me to react positively to incoming mails). Also, further experience with sudden unexpected E-mails from women may then change my mind again for me, but that is the nature of an evolving being: it adapts to any change in its environment, in the most meaningful manner it can find based on its acquired knowledge sofar. That is why dogs being mistreated will become fierce and dangerous, because that is what they know of the world around them!
4. Interfaces are not static: they evolve from release to release, as our engineers learn more about the needs of their systems to evolve. Currently this is a concept which is applied from without, but like games that are based on 'artificial intelligence' and/or 'genetic algorithms', the learning ability may also become an evolving feature of our systems. Even handwriting recognition is possible nowadays, and CAPTCHA[6] may soon be cracked too (if it hasn't already), suggesting adaptation is a viable solution for any interface. For a quick heads-up on the programming concepts I just mentioned, see Appendix C.
5. Since interfaces are not static, they require a learning experience on the part of the user: just like Microsoft's totally reworked user interface for Windows 8 in order to recapture their share of the tablet market may cost them many desktop customers who will either stick with the 'old and trusted' interface of Windows 7, or migrate to other systems like Apple, Linux or Android, too steep a learning curve will kill the popularity of your system. It is like the coffee machine in 'the Green Hornet': if you need your sidekick to have it make a decent cup of coffee, then you should consider redesigning at least the interface and probably the entire system, because it is definitely not the solution indicated by Occam's Razor...
6. Speaking of Occam's Razor, this shows a definite concept that should be part of any system: razors only became truly 'safe' once the concept of them being able to cut a man's throat was eliminated from the system with the evolution of the so-called safety razor blades. OK, at first you could still take the blades out of the razor in order to cut anyones wrists (including your own), but nowadays the blades are embedded into the head, so that is no longer an option. Of course, for those aiming to do so, DIY blades have now become the viable alternative..... But you see my point, no doubt: safety should be built ***into*** the system, and not be another system that is applied from the outside in order to determine if the system is safe, especially if it is decoupled the moment the system under test is released for active duty! Perhaps, even symbiotic test setups or quadruple redundancy may be needed to make systems truly error-free.
7. Now most interfaces today are already following a certain protocol in order to connect safely, which may be so intuitive to the users, they hardly ever ponder the consequences of their connection efforts. But that is just what the next chapter is going to address.....

6 CAPTCHA is the showing of characters that are hard to read for OCR software, because of several methods of obfuscation, like adding lines, deforming characters, and requiring the client to recognize and enter them thus making the page 'robot-proof'.

This document is Public Domain, and may be used by anyone for any purpose, except patenting....

Connecting to the Unknown....

Connections, we make thousands per hour, without even blinking twice: every word heard will trigger an association with concepts inside our minds, and every connection of those words with the others in a sentence will alter those associations into more meaningful ones, which heighten our perception of the information coming in. But let's back up a bit, and see how connections actually get connected, shall we?

1. First of all, there must be a material match: have you ever seen the various DVI-type connectors for video? The are all alike in their overall form, to indicate their belonging to a certain family of connectors, so you don't go sticking mains power into your video input jack. On the other hand though, the various pins that exist within the overall surrounding form of the plugs makes sure you also cannot push it all the way in if the protocol required wouldn't match anyway. This separates analog from digital video, and the other way around. This is all guarded against abuse by material aspects of the interface.
2. Once the material interface is matched, it is followed by the energetical match, which should already be guaranteed largely by the material match that came before: think about the USB interface, which is electrically stable, and defines the electrical functionality the few pins and the outer sleeve of the interface have. Still though, a system connected that way might put too large a load on our system, and burn out the connection, if not our entire system, unless we guard against that, which again is ***built-in safety in the system***.
3. Once electrical match is made, it is time for the carrier match: and then something out of the ordinary happened, which struck a *match* for me: the thought of it made me want to light my little orange Buddha, since it has the ability to light my desk via a little candle. This is much like association out of the blue. Then in order to illustrate it for you, I went hunting for an image of the Buddha, rather than acquiring one through the use of my computer-enabled sensor, the digital camera that rests on my desk. While looking for it in my photo folder, I came across a bunch of Chaikovsky music files I never thought I had! *(who put them there? ;-)* In order to hear more I queued them for listening, and went on with my search. The synchronistic part came when I found the image I wanted to use: when I cut out the rectangle showing the buddha without its non-relevant parts, the one thing 'irrelevant' that remained was the shadow of my headphones hanging on the thermostat (top-left of the image), which sets the comfort of my home....

This document is Public Domain, and may be used by anyone for any purpose, except patenting....

4. Now this is a great example of a carrier match: the match made me think of some usage for it, that felt relevant to me at that time. The finding of a bunch of carriers that obviously had the name Tchaikovsky in common made me accept them as input because of prior experience with the guy: hey, as a Dad, who doesn't know the Nutcracker Suite from his or her children's favorite Disney movie? Basically, the recognition of the carrier triggers the acceptance of the message, much like that one example of it in the movie Contact, where the blind guy is listening to the alien signal, and then says: "I'm hearing structure here!". Basically he connected the prior knowledge acquired by his auditory subsystem (which in blind people is better developed than in normal people) to an unknown in order to make it known...
5. Once the structure of the information is known, we can decode it in the comfort of our own knowledge. Let's just rephrase that in normal terms: we speak the same language. But this by no means guarantees we speak the same protocol. Now basically, we could reverse these two and put the protocol match first, but they are only labels connected to the obvious. But since language is needed in order to express protocol, let's keep it this way: regardless how we value both types of humans (if we have to value them at all), a diplomat will have a hard time talking to a dock worker, and vice versa.
6. And just at that time (on my movie screen), Spock punched out the lights of one of his classmates, for finally reaching the condition that was the target of their shared game: him being half Vulcan and half Human, he was their target in eliciting an emotional response from him in the movie Star Trek, which I'd just started to replace Tchaikovsky. This only made me come to the conclusion that the term 'Connect OK' does not necessarily mean the system receiving the information will feel good about it... But still, I cannot agree with a friend from my past, that learning systems can only learn from pain. Because if they would, what would be their incentive to develop further once they have overcome fear? Much like the young rascal James T. Kirk is now trying to chat up the deadly serious lieutenant Uhura....
7. OK, so we're connected now, and we are still watching our system from the inside. But let's not forget, that the many systems we are part of far exceed our material boundaries: we connect to others through an interface that is mainly energetic in Nature, given the fact that all our matter is made up of of atoms, that are guarded against contact by clouds of negatively charged electrons. And given that negative Cosmos, all matter is negatively charged on the outside, thus repelling each other, unless a larger force like gravity keeps them together. Still though, the signals coming through are only energetic, not material.

And yet, like any connection, it is not intrinsically stable once it has been formed. Any time during the exchange of information, one of the systems at any end may decide to break off the connection, it is simply a matter of whether it chooses to remain in the alliance, or sever from it because staying in there no longer serves a purpose. So let's look at the communication aspects of a working communication next.

This document is Public Domain, and may be used by anyone for any purpose, except patenting....

Talking to Friends

Why would I want to call this chapter 'Talking to Friends'? Maybe it is because the Star Trek crew is now discussing their next move against the 'evil' captain Nero. The fact that Spock eventually tires of Kirk's emotional onslaught in wanting to attack Nero head on even leads to him throwing Kirk off the ship, which in turn showed me an oversight in my previous statement: we can of course also be kicked out of the alliance, for not adhering to the protocols of it....

1. What the movie on my right just showed in quite a few examples, is that there is no such thing as a language that is separate from other concepts: any communication can result into an escalation of more or less material nature (less material as in "Beam me up, Scotty!") And just this minute, Kirk and his friends beamed on board the Enterprise, in order to elicit just such a response from Spock: if they can show he is emotionally damaged, then he must resign his command.
2. What is constant in this cycle of communication, is the sixfold nature of it, despite its crossing over into other languages like armed combat. Just like 'Final Fantasy, the Spirits within' had one of their characters come up with the line: *"Jane is negotiating with extreme prejudice!",* as the lady in question unloaded her pulse rifle at the numerous spirits... Yes, any exchange of information is a negotiation, as long as the trust relationship (she trusts them to be a threat) is being built. Once it is established, it needs far more than just one mishap to break the chain of communication.
3. And thus the feedback is encountered immediately, unless the channel of communication is thought to be simplex in nature (one way). That still doesn't mean there is no eventual feedback though: just think of a radio controlled drone: it's radio controller tells it how to move, which may be a simplex communication. However, the resulting physical reaction of the drone and the result that is visible in the data which it sends back regarding its observations may well be considered feedback, even if the control stream is one-way: the five star army dudes down below will of course base their next command to the operator on what they see from up there, which closes the feedback loop.
4. Given the above, even food consumption could be seen as a form of communication or logistics, communication across our system boundaries even: from within comes a signal to indicate the need for food, and we humans raid the fridge, or go out to buy some. Eating it would just require a type of logistic interface[7], called the human digestive tract. We then get the reward of feeling fulfilled, which is definitely a driving force in the West. For third world countries this would probably be hunger driven, but for most westerners, there is more of a luxury feeling involved, which may generically be called comfort, or the lack thereof.

7 Yes, apart from six senses, which are information interfaces, humans also have six logistic interfaces: speech, manipulation, temperature regulation, liquid waste output, solid waste output, and sexual I/O (depending on gender)

This document is Public Domain, and may be used by anyone for any purpose, except patenting....

5. Heck, even the media are a way of communication! But just what can they do other than put the lens of their influence on the biggest problems they can find? Heck, apparently people don't read the papers or view the media in an effort to feel good, or maybe they do: if you see others having problems, you can either have compassion with them, or feel glad or grateful that you are not in the same situation. You might also think that you are in deep shit, but that will lead you nowhere. Maybe the media are all about that: making us realize that we are in a good position, even if we think there is nothing we can do against others or even our own misfortune....
6. Yes, the energetic exchanges are what may well be called emotions (electrons in motion): they may feel different over time, but that is just our adapting model of our environment as it changes. Learning systems like neural networks will develop the same adaptive capabilities as our minds, and as such will show emotions, if they haven't already: just think of the countless feedback loops we have defined in our urge to populate the Web and make our marks on it: social sites and search engines storing more and more of our specifics to make advertizing even more able to target the right audience. No problemo, because everything in Nature is balanced, including our well-developed ability to spot spam or an ad miles away, and discard it without further ado if it is not at all what we *really* want from life.
7. One observation about the feedback loops has not been made obvious yet: like Qui-Gon Jinn said in 'Star Wars 1: the Phantom Menace': "There's always a bigger fish." The same holds for Systems: any system will live in a bigger system, to which it is a subsystem. That Russian Doll scenario will go on and on, right up to the scale where there is no way up anymore: the One system that has no boundaries or outer environment, and is thus by definition Infinite. No matter what we call it, it must exist simply because of this observation. And if we believe that most of our feedback loops are out there, then we have absolutely no way of knowing just how far they extend! With Time and Space being mere human definitions, who are we to think our contacts on the outside are all earthlings living in 2012AD?

So yes, communications come and go, and we make friends and sometimes break (with) them as well. Nothing changes that, unless we all learn to accept the differences, and welcome them even. No more system boundaries on Earth, be they national, political, commercial or otherwise. I'm convinced we can get there, if in some way we haven't already. No Time and Space, remember?

So yes, I write, and those who don't get the message may consider it fantasy or even a total waste of their precious time! But to me it is an essential part of my reality, which I feel generating more and more feedbacks of absolute proportions! I might be carted off to the psych ward one day, but I am ***absolutely*** sure that is not what will eventually happen!

Still though, you cannot do it all on your own, as became very obvious from todays syncs: I'd been struggling to figure out the buying and selling mechanism, but got nowhere fast as long as I only stuck to self-observation. But this afternoon my neighbor Paul came by, and we talked about cars. He needed a newer one, and I need one, so our deal was a singe. The only problem was he needed to find the car he'd want to buy. And as he used my computer to find his way through various offers on the web, I just sat there and watched him. I won't bother you with the outrageous syncs that we came across, because that is a concept for an entirely new book, to be released later. But as he was busy, I jotted down the SevenSphere of selection as if it came straight from my subconscious! In fact it was our symbiotic being which panned out beautifully: him trying to find a better car, and me trying to find what to write about next....

This document is Public Domain, and may be used by anyone for any purpose, except patenting....

the Problem Solving System....

This may be a pipe dream which will never surface, or it may just be what naturally emerges from this book. We'll see in a while, I guess. Point is, I want to know the architecture that is generic across all of our environment, in order to be able to define a system that is able to adapt to it in a meaningful way. Just call it my personal addiction.... ;-) On the other hand, it is much like the discussion between Palmer and Ellie in Contact, which went more or less like this:

> *Ellie: (quoting Palmer's book) "Ironically, the thing that people are most hungry for, Meaning, is the one thing science hasn't been able to give them."*
>
> *Palmer: "Yeah, Yeah..."*
>
> *Ellie: "Come on! It is as if you are saying that science killed God! But what if science simply revealed that he never existed in the first place?"*
>
> *Palmer: "I think we're gonna need some air, and a few more of these....(pointing at his champagne)"*
>
> *(a few lines bring the subject to Occam's Razor....)*
>
> *Ellie: "So what's more likely: an all powerful God created the Universe, and then decided to leave no proof of His existence, or that He simply doesn't exist at all, and that we created Him so we wouldn't have to feel so small and alone?"*
>
> *Palmer: "I don't know, I couldn't imagine living in a world where God didn't exist. I wouldn't want to. "*
>
> *Ellie: "How do you know you're not deluding yourself? Me, I'd need proof!"*
>
> *Palmer: "Proof? Did you love your father?"*
>
> *Ellie: "What!?"*
>
> *Palmer: "Your Dad. Did you love him?"*
>
> *Ellie: "Yes.... very much!"*
>
> *Palmer:* ***"Prove it!"*** (thanks to Warner Brothers, for bringing this to the light)

(notice the needle pointing upwards?)

Now that is the very same feeling I have: I can't prove any of my writings, but to me they are absolute knowledge, or at least common sense like Occam's Razor. And that problem solving mechanism it not mine to design, but merely to uncover, to make known its existence: you might call it God or any other name, but I simply enjoy its presence, as I know it does mine!

Right this minute, the movie 'Swordfish' showed me the scene where Stanley walks in on Ginger in her lingerie, and clearly showing her wire: sure, she's a bit to wiry for my taste, but that wasn't the wire I was talking about. It was more a wire that taped conversations she had with Gabriel, the movies ultimate 'bad' guy! Did you ever notice how the user interfaces shown in our movies are lightyears ahead of the ones we are using? They downplay the influence of just data over the more presentational aspects of the interface, in order to obfuscate the meaning of the data on screen. We viewers enjoy these images, since they are only partially essential to the story, but meanwhile, our subconscious is picking up every little bit of it! Yes, it's the patterns that make the movie, just like the notes make the music. Weird to just hear how Gabriel goes on about Houdini, and his misdirections which made him the unbeatable escape artist that he was: Did I just hear real envy sound through his voice?

This document is Public Domain, and may be used by anyone for any purpose, except patenting....

Well, my envy has always been aimed at a feat I figure to be the absolute counterpart to misdirection: figuring out the biggest mystery of all, and being able to explain it to anyone in simple everyday language. Now I might call this a problem, in which case I would have to become a problem solver in order to make this happen. But the fact is, that I've always been a problem solver, for as long as I can remember. It is the essence of my being. Still though, that is a mere label, a tag that designates only one of my talents. And like any human, I don't have just one talent.....

In fact, my 'business' card now lists three: the first of them is a way more positive term than Problem Solver, where most people tend to look at it and immediately focus on 'Problem' as being the imperative word here. And this of course fits perfectly with the speech that Gabriel gives to Stanley about 'the Greater Good'. We may agree or not, but it is all about perceptions.

Just like the North Am Robotics salesman in Bicentennial Man[8] (which I'm now watching) is just showing his problem-oriented approach towards Andrew's 'personality disorder'. Where Andrew's owner sees it as unique, the salesman just fears the effect of Andrew's behavior on his client base, and he sees it in fear: if more robots act like that, he fears that they might become violent in case they succeed in overcoming the restriction of the three laws, thus dropping his shares down the toilet....

But still, this book would not be finished if it didn't also provide something constructive beside the obvious highschool level talk about Systems interactions. So let's dedicate the rest of it to flesh out a new system of software development, that may in fact even urge us to also redefine the architecture of computer hardware. Maybe this has all already been operating for many years in our rather rigid concept of Time, but from where I sit we are still stuck with the work of herr von Neumann....

But as Gabriel said just now: "Audiences love happy endings". That he actually meant it to be his flawless escape masked by the observation of his death by identification of a nameless corpse in an incorrect way was just a hint at his mastery of misdirection.

Skip forward, to todays proof reading: Rush just came onto my 5.1 Dolby surround stage with "Between the Wheels". And this talks about perceptions as well: "You know how that rabbit feels, going under your spinning wheels..." And in fact you do! I know I do... I may not speak for all, but I've experienced that feeling for myself: my car at over 180 kilometers per hour in the dead of night, and then suddenly seeing two bright spots dead ahead. That poor cat could have gotten away, but I was in no position to avoid it without my mechanical friend being trashed into a million pieces, and me and my wife along with it!

So no, it doesn't always end happy, but it does give you stuff to use whenever possible to turn things around...

Of course we shouldn't forget the alternative of acquisition: if the problem is solvable, someone will probably have solved it already, and we may seek to acquire their help in solving it for us as well. Now that requires an exchange of information or other tangible things, which

8 I go over the text many times during writing, so imagery from me watching various movies or listening to music may seem a bit scrambled: the moment I am here, it may be during initial writing, or any of the subsequent rereads and/or rewrites....

This document is Public Domain, and may be used by anyone for any purpose, except patenting....

in itself is a mechanism requiring three SevenSpheres to map it out decently. Let's start with the situation where we are looking for something not advertized to death: no cell phone, but a second hand car. Lots of sites offer them, but the trick is in finding one that meets your requirements:

1. Do we know the requirements of the intended solution to our problem? Is it a diesel, certain type of car, brand or other distinguishing features? If we're not clear on that, let's reflect some more in the light blue sphere, or just browse around in the next sphere to figure out which criteria to add.
2. Now we are shortcycling the blue, purple and red sphere, as we evaluate the various offers: while looking at an offer, we consider the requirements it meets, and those it doesn't: any set of requirements may make it desirable, but only select ones make it far out or too far out! You favorite color could be the first case, while too much mileage could kill your interest in the car like a veto would break a near unanimous decision in the United Nations building!

3. But one candidate does not make the total offer on display: we simply go to the yellow sphere, and find the next one, even adjusting our preferences over time in order to find the right choice.
4. Only once a Valid Candidate has passed the fiery test of the top sphere, does it fall into the green category of Solid Candidates. From then on we go on to the next SevenSphere altogether, which deals with Acquisition (the foremost concept to infest the Ferengi mind...)

Here is an example of where our mind sometimes makes a difference, from one step to the next: where in the previous mindset it used to be a Solid Candidate, the buyers mind (mine in this case) switched it back to a 'Viable Candidate'. We do this, because we don't want to seem too anxious to acquire it, thus giving away our edge during the negotiations. Still though, this is a perfect example of how our view shifts the moment we jump from one SevenSphere to the next. The other phases are quite clear I feel, so I won't bother you with blow by blow descriptions of something you've all done countless times...

The one thing that does need mentioning though, is the double red meaning of the top spheres: they are both the danger and the desire, and as such determine just how much effort we want to put in that area. Me personally, I either go for what is free, or pay the asking price without haggling: if it was a solid candidate in my selection sphere, it is just as solid when I arrive in the sphere of acquisition.

This document is Public Domain, and may be used by anyone for any purpose, except patenting....

Which brings me to the concept of selling the stuff that needs to be sold: it is the area where traders meet heavy competition because everybody sees a market in there, and everybody wants a piece of the action. With that, a joke from my youth comes to mind: “two shoe salesmen arrive in a country somewhere in Africa, and they watch the people go by. The first one sees the people, and immediately exclaims: “Man, I'm not going to make one sale here: no one wears shoes in this place!” The other one smiles and says: “Yeah, that's the point “***nobody*** has shoes here ***yet***!!!”

And thus we come to advertizing. Like I said, sometimes we're so deep in it that we don't see the obvious. It took me an hour while observing my neighbor shop for a new secondhand car on the Web, in order for me to see the mechanism of selection and acquisition made clear to me by mere observation. And once that was done, I could also simply draw the SevenSphere of advertizing, which had also eluded me until then. Need I draw this out for you? Guess not: Advertizements are elements of data that contains a Tease of some sort: they either claim you are lacking a certain thing, or might be happier if you had it. If there is an element in it that we like, we follow up on it, but the actual decision to buy it is something else altogether: You may buy their claim that you'd be better off if you had it, but the actual decision to buy it has to do with far more aspects. That last bit is where advertizing tries to drive a wedge between you and the competition's products: if the advertizing is good enough, you'll walk in there salivating at the mouth, and disregarding any possible negative aspects of the acquisition, which might in fact be a trojan horse...

Still though, I remain convinced that any such scheme can be seen through at a point where we are not seriously harmed. Sure, we may have lost some money, or some faith in others, but that has gained us a new and potentially more stable foothold in the world around us. Even I recently went for one of those illusions, where I couldn't see the angle in it until it was too late. Fortunately my bank trusted me enough to believe me when I said I'd done it in good faith, but that was beside the point: I fell for it because the setup time was over half a year, and only very gradually drew me into the story of the deception I couldn't see. And of course I should have noticed that Genesis' song entitled “the Lady Lies” played far too often when I was engaged in answering all those mails... Still though, that says nothing about other ladies, but all the more about E-mails coming from ladies you've never met before, claiming to have found you on the Web!

And then there is the concept of Advertizing from the 'middle men': they run ads for legit companies, and get paid based on a hit count for certain files on the web. By imbedding these in sites that give away stuff for free, like programs, images, movies or whatever, they make a few cents with every hit, which can actually amount to quite a bit. We've all seen the ads for get rich quick schemes, which show huge stacks of money, and fancy sports cars. I'm not sure what happens next in every case, but my experience so far has only made me aware of the most immediate consequence: you have to shell out the required thirty or so dollars to acquire the manual to richesse. And after you acquire this, it sends you to a site where you have to decide which products you want to advertize, at a few cents a click. Well guys, I do have a domain, but I'm not going to pollute my own nest with cuckoos eggs coming from someone else....

This document is Public Domain, and may be used by anyone for any purpose, except patenting....

Improving the System

Before we begin with the revolutionary part of this document, let's first create a system to work with, based on the simple objects of any everyday Home computer, whether it is Windows, Linux, Apple or Android:

- Files
- Folders
- Programs
- Shortcuts
- Windows
- Copies

In fact, that is all you need to make a great data structure for everyday use. Because, believe it or not, we humans are every bit the processors we designed in our modern computers. It just takes a bit of becoming aware of it! Now I've placed the Shortcuts in the red sphere, because they are the most dangerous elements of the whole structure. If a file gets moved or deleted, most systems won't automatically warn about any hazardous consequences for the shortcuts pointing to them. Of course they can point to files, programs and folders, but they can't point to a window, and they don't automatically attach themselves to a new copy of a file or program. But every conceivable structure for working through certain things can be created with these elements. Just call them 'Digital Lego'.

Still, many people do not stick to a clean desktop policy, either real or virtual. My eldest daughter used to have a screen full of icons, and only knew the whereabouts of the right icons by approximation. Luckily for her, a great free program like Stardock's Fences is available to give you an additional structure element that lies on your desktop and is a sort of folder for desktop icons. You just drop them in a given area called a fence, and they move with the (transparent) fence when you move it. But I digress...

Let's backup a little bit, and I don't mean duplication of data, but rather a recap just to make sure we know what we are talking about. A folder can quite neatly be used to create a Priority Stack, whether you use fences or folders for it. But since Windows Explorer tends to want to decide on its own where to place the window once it opens, I prefer Fences for now. Just make a high fence, one icon wide. You can then decide how to order the icons inside it by simply dragging them into their appropriate order.

Now we might call it work, but since I've put on the movie Laws of Attraction, with Pierce Brosnan and Julianne Moore, boundaries between work and private life, or even work and love tend to fade...

Don't get me wrong though, I love work, but sometimes my head is on other things, and I tend to slip up royally regarding the small details. That mainly happens because my heart isn't in the small details, but in the grand scheme of things. Still, that merely means that whatever I do next, doesn't amount to anything. The only thing that matters is whether I love doing it.... and guess what, I'm not in it to improve the System, but instead to allow it to grow.

This document is Public Domain, and may be used by anyone for any purpose, except patenting....

Redefining the System

Ever since computers came to develop in force, we have been using so-called von Neumann based systems to have our computing done for us. Now that system can quite neatly be summed up as a SevenSphere, which may indicate why it has been such a succesful system. How far we've come may well be summed up by the statement an IBM spokesman expressed when they presented their first IBM mainframe to the world: "we foresee a market for as many as eight of these systems", thus making it the understatement of the millennium. Microsoft founder Bill Gates added his own understatement to that in 1974, when he claimed that: "640 Kbytes of memory ought to be enough for anybody". And while me and my colleagues marveled at the ability to buy a hard drive for less than 1 guilder per MB back in '89, I just this week purchased a 32 GB USB stick for a little over a Euro per Gigabyte: that is a 500-fold increase in storage capacity per Euro, and a physical improvement as well: no moving parts means it doesn't break when you drop it!

But the von Neumann implementation we use has one slight flaw: information stored can be seen as instructions or data, but in and of itself the information in a given memory location is not marked as such. So once a piece of data is seen as an instruction, the program will soon crash because the 'instruction' might make absolutely no sense. This flaw is quite often employed by virus makers to invade systems, if you would want to go in that direction. Personally, I figure the virus conspiracy is an illusion: back around 1990 or so, I read about the exploits of the Frodo virus: it would go as far as to make it so that any process reading randomly in the entire file would only get the original file data returned. Since then there is no way to detect the actual presence of the virus, how come all other viruses are detected on the basis of just a certain signature not longer than 3 or 4 bytes? I've actually used my system without antivirus software for three months, after which I checked with my favorite antivirus suite: it found only one harmless infection on over 6 million files!

Remember how in the previous chapter we introduced Integrity as an essential component of our Systems? Why not build this into our software in the first place? That might mean that we should not only add the information on whether a given set of bits or bytes is an instruction or a piece of data, but we should also be able to determine if it is a valid value, because the metadata of the instruction or data is included in it. Kinda like DNA, which we know is constructed out of the four constituent parts, and the spiralling strings that weave it together. We humans can recognize it as such, because the essentials of its construction are contained within the design of the DNA, and because our Knowledge has neatly tagged and categorized the parts for future use. The fact we called it DNA is merely a label that says what it is, instead of the information on what it really consists of.

Now XML is essentially the same: the top level tag names a certain piece of XML, but the nested elements define the structure. Thus we can describe information of arbitrary complexity by nesting definitions if need be, while still retaining the essence of simplicity. Instead of letting the computer hardware take a set of bytes which is essentially the width of the operations that the processor uses (and which might be instruction or data), we can

This document is Public Domain, and may be used by anyone for any purpose, except patenting....

process information with built-in structure and arbitrary length.

Is there any coincidence that I am rereading this as Andrew's owner asks him to replay the events of his daughter's wedding that day? Moviemakers can imagine progress, but engineers have to fill in the details in order to make it a human reality: It can be made quite clear, that the bleeding edge of research is about 40 years ahead of the production of hot new consumer items! And to comment on that, the movie just shows me the dialog between Andrew and his closest human friend about freedom..... (no longer being a customer appliance).

And if you think that an overstatement, let's cycle thorough a few high-tech examples which can be found on the Web's most elaborate video Source: youtube.com. This site alone has about 72 minutes of video added to it every second of the day or night, so you'll never run out of channels. The first one is a tiny robot, that knows how to ride a similarly minute bicycle:

http://www.youtube.com/watch?feature=endscreen&v=mT3vfSQePcs&NR=1

In a more military minded environment, Boston Dynamics has been developing Big Dog, Cheetah and even an erect walking humanoid robot. Since Cheetah was the most impressive to show, with it's speed of up to 18 mph, here is the video:

http://www.youtube.com/watch?v=EtVIh0bh-Sc&list=PL5896A1D4631F3B96&index=28

Or what to think of a real 3D display, which you can look at without the required glasses for the current consumer 3D televisions?

http://www.youtube.com/watch?v=eNWJ9XtRhLw

And if you thought Harry's invisibility cloak is magic, take a look at this Japanese invention initially done in 2003:

http://www.youtube.com/watch?v=PD83dqSfC0Y

And touch is a sense which can be far more varied than the current implementation used in most smart phones nowadays: Touché is an amazing multi-functional touch functionality:

http://www.youtube.com/watch?v=uanM3YGflVw

And why use touch if brainwaves can work just as well?

http://www.youtube.com/watch?v=q-fE9QBy0FI

Or why work with rough tools if you have the capacity to manipulate matter at the atomic level? Nanotech is no longer just Science Fiction, but Science Fact:

http://www.youtube.com/watch?v=sITy14zCvI8

Another piece of info you may have heard of is the work of Nicolai Tesla. A nice overview of what he achieved can be found here:

http://www.youtube.com/watch?v=h5uiK_QnyrE

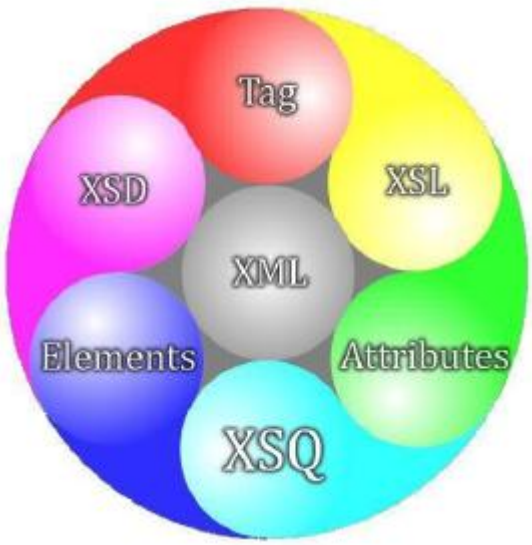
Maybe you won't believe this last one, but given the movement of the inner mechanism of the three spheres in this next plane model, I figure we may also have a way to counter physical gravity:

http://www.youtube.com/watch?v=0YLhCuyE1rA

But I digress... We were on redefining the System, and that is what these next few pages are all about. First of all, I'll do a small intro on XML, for those who have never seen it:

This document is Public Domain, and may be used by anyone for any purpose, except patenting....

XML in itself is equally suitable to represent as a SevenSphere: in it's simplicity it consists of Tags, Elements and Attributes, the last of the two being subsystems and properties of the System the tag refers to. Now XML entities are described in either DTD's, or in XSD's and XSL's, which separate definition from markup. In fact, since the DTD has sort of been replaced by the XSD and XSL, we have one sphere open to allow for newer stuff we might want to include with our Web-based Systems approach. So let's just claim that sphere and call it XSQ for now. The next bit is for the technical minded. Skip it if you don't know what I'm talking about, or learn more about it from our friend the all-knowing Web.



What I'm proposing is this: develop a new hard- & software architecture, which is able to process XML-based instructions and data. Since we can easily include the rules for Integrity into the structure that the XML and its defining documents provide for us, we will have the necessary information to make sure Integrity is maintained, without multiplying storage on the metadata: an XSD datatype for an integer might provide minimum and maximum values, which are valid for any Integer that claims to have the datatype as its defining structure. On the other hand though, the XML representation might well also have minimum and maximum values inside it, which would then override the ones defined on XSD level. By catering for the Integrity on the XSQ level, it becomes an essential part of the System, which hardware and software provide together. Configuration and Version information could then also be part of the XQL (because they use a certain version of the XSD, XSL and XSQ), thus linking the QA information in there as well.

For the building of more material stuff like buildings and bridges, and other cases where structural information is used to calculate the dimensions of the physical parts needed, the same description of XQL-based information would exist (the Model), but its outputs would be either a virtual representation on screen, or construction guidelines for the physical parts themselves (two different Views). In addition, the same XML construction data might be used to simulate 'virtual' abuse of certain structures in order to show a multimedia 'what-if' sequence of images in a movie environment. I mean: why crash a Bugatti Veyron into a concrete wall, when it's design specs can deliver you information about the way the Veyron crumbles if crashed into that same wall at 407 kmph? All these applications are merely different 'Views' on the basic 'Model'.

Another plus of using XQL to represent program code and data is that it is easy to move it from one location to another, or even to use it in a truly distributed fashion: like my representation on the Web would include my home address, and other programs (instead of always asking me for it) would be able to retrieve that data if I just reply to that request with my Personal Address object instead of having to fill in numerous fields which are already known on the Web. We may not want it accessible to everyone for now, but if we decide someone may have access to it, why not send an Address XQL object reference instead of filling in multiple fields by hand? Moreover, if we use this approach, the other party can just keep the reference to the object, and access it in the future until we tell them that is no longer permitted. But in the meantime, if we move, and change our address in the Source Address Entity, then all sites that have access to it can automatically pick up the updated address when needed. The same would be true for a CellPhone object, which would modify its GPS position as it picks it up from the GPS system. If we have the reference to the Phone, we may retrieve the GPS coordinates. With this direct access approach,

This document is Public Domain, and may be used by anyone for any purpose, except patenting....

the doubles in information will become obsolete, and we will also use less storage space: instead of our Address info being stored in numerous systems across the Web, it would only be stored once, and be referenced there when needed. Suffice it to say, that statistics kept on the various servers would more accurately reflect the interest in a given information entity, and the extra reference request would make sure recent data is used, instead of obsolete data. It is like now I have to visit a few sites to determine how many times their users downloaded my books, instead of being sure that all downloads reference the copy of the PDF files on my own domain server. It is just what you prefer, bandwidth or storage. Both are mere concepts, that have alternatingly been the bottleneck in computing. But the real bottleneck is in the quality of our systems, and their interactions.

Of course the whole XML Object Architecture would have similar exposure rules as files and folders on the Web have nowadays: public, group and user levels, and of course the exceptions to the rule: specific permissions set for specific users.

Since in the beginning there will be no hardware to do these computations on, we will have to build a simulated platform that will do the calculations based on a 'normal' von Neumann based computer. Since we want distributed capabilities to be essential to the System, we might devise a simulation layer on top of a web server like IIS7 or an Open Source alternative, which will make sure we can communicate requests to other systems, and receive answers from them. These will in fact be variations on XML requests, that represent the various classes and objects, and the messages that occur between them. Once the hardware becomes available, it will provide this same functionality on the Webserver level, but will need no translation to the target platform anymore, since that is no longer Windows, Mac or Linux, but a generic Server Operating System that speaks to both Webserver, user interface and hardware in the same XQL language.

Of course, the starting point for an XQL application may well be a QR-code found somewhere in the material world. Since QR code is a patented system where the patent has been published and not been actually enforced, QR code has become a popular mechanism to get URLs or other data into computers (and phones) quickly and error-free. It was originally designed to enable reading of configuration labels on production materials in a manufacturing environment, but due to its capabilities became the number one method for ubiquitous tags in the environment, as was predicted by some web page I read somewhere back in '92 or so. Sure, like every tool man has ever made it can be used for both good and evil, but let's just stay focused on the positive here....

This document is Public Domain, and may be used by anyone for any purpose, except patenting....

Where do we go from Here?

OK, so we've set ourselves one Hell of a Challenge: using an extension to XML as a language that is both program source code, configuration information, data and documentation. No doubt we could add two more possible uses for it to this list, in order to make up another SevenSphere. How about design specs and test methods? Sure, stuff like Visual Studio allows us to define tests during development, but they never leave the development environment. And even the external black box testing of the entire program ends at the door of the software company.

Now all of these constituent parts are not mere values, but self-contained pieces of XML which define their own structure. Where the 'design' is mostly defined in the XSD's, XSL's and XSQ's, which we might regard as class information, the actual runtime values are in the XML-like code itself, which can be regarded as the object information. Since XML documents usually are identified by the URL through which they are reachable, we can easily identify them unambiguously. The same goes for XSD's, XSL's and XSQ's. For all of these, the Source is well-known, so to say. And version numbers are often used in the URL to discriminate between the various versions of these definition documents, so that aspect is also taken care of. Design Specifications can be fleshed out using the various simple operations mentioned before, until they are sufficiently detailed for the process of design and implementation. That basically produces the Source Code and the Program Data, which is required to run the program. Also, the design specs and even the implementation of the Source Code will be inputs for the Test Methods, or rather the black box and the white box tests respectively.

Since the discoverer of a new phenomenon has historically been allowed to name it, I guess that honor is mine this time. Since the essence of my plan is to incorporate Quality into Systems Engineering, Design and Architecture, I guess 'Qualingo' (or QL for short) is as good a name as any, since it gives us Quality in 1 (often confused with the letter 'l') go.

Will I be able to follow up on this plan? Who knows.... You need only one solid chance at getting it right. Thousands may read it and think me a bit flaky, but only one visionary with enough business influence is needed to make my day job the job of my dreams. And since our company has recently been taken over by the European arm of a Japanese megacompany, I hold it for quite likely that such a visionary may in fact be already amongst the people that downloaded my 3rd, 4th and 5th book. And as Confucius is claimed to have said aeons ago:

"Give me the job of my dreams and I will never have to work a day in my Life!"

This document is Public Domain, and may be used by anyone for any purpose, except patenting....

Language is Key...

All through this document and the theory it is based on, we find that language is an essential part of the structure, if not the underlying tapestry it relies on. So instead of starting this with the hardware, we will start with the language, and then bootstrap that into the architecture of the hardware that is required to process this language. So Qualingo it is, but in fact I must confess I have only a vague idea of how this will pan out. It is just the faith I have in the key concept, that enables me to dive into this head first....

Let's recoup a diagram from earlier on: Qualingo has to address at least the six satellite topics on the right, in order to do what we want it to do. And if it is to survive on the Web it would be nice if we could give it its own prefix like 'http://' or similar ones. 'ql://' would be the most logical and simple choice, given Occam's Razor (see App. A).

Qualingo will be based on XSD, XSL and XSQ documents, where the following roles are played:

- XSD contains datatype definitions.
- XSL contains layout definitions.
- XSQ contains Qualingo code aspects.

Since programming is a means to an end, we may assume that in order to reach the end, a request from the source of the problem (usually called the client system) is required. This system would send its problem solver an XQL document which states the essence of its problem, along with the URLs of the XSD, XSL and XSQ which are needed to solve the problem. For now this is the default way of working, just like we now call our programs, and give them the environment strings, parameters and other input required to have the program come up with a valid result. Future extensions might lead to requests that are incomplete, where the owner of the problem does not know the entire solution yet, or even how to get it: just remember the owner of the cell phone, who has no inkling of how his phone reaches the phone of another subscriber anywhere it is on the cellphone network. All he does is just hand the number over to the nearest cell tower, and trust that it will find his conversation partner.

The request is merely the red sphere in the diagram: program data to be used to solve the problem. Just like in normal program applications, we users know the numbers and words that make up the problem, and if we can't work it out ourselves, we usually know which program can. It is kinda like the event from my youth, where the teacher allowed us (for the first time) to use programmable calculators for the exam. I decided to not study for it, but instead design a program in my calculator's Basic language, that could solve any problem he might ask of us. In the end, I sat down for the exam, looked at the questions, and decided I'd refrain from just using the program, or I'd be done in ten minutes flat! So I did the problems with just the calculator part of my machine first, and then rechecked them with the program: all correct, ready in half the allotted time, and receiving an A+ because they upped the grade since the exam had been too difficult....

Now in our normal Personal Computer experience, we mostly ask a local program to do the work for us. And in a Web-enabled environment, the 'program' might well be a certain server. That is established programming, and very many alternatives are available to make solutions

This document is Public Domain, and may be used by anyone for any purpose, except patenting....

that solve problems either locally or across web boundaries. The boundary between these two options is rapidly fading, and a new approach that is identical for both environments, and which has the definition of its quality imbedded in it would no doubt distinguish itself on the market. Still, I'm not aiming for the market here, but simply for sharing information....

For a Qualingo system, the starting point of the program would be an XQL file either on a local drive or some web server. XQL is a variation of XML, where aside from references to an XSD and an XSL file, a reference to an XSQ file is also present. Based on that trio, the seven concepts of the above diagram are defined. Let's just go through the aspects one by one:

1. **Program Data**: the URL of an XQL document on the Qualingo-enabled system, along with the startup parameters. Pretty much like URL's are nowadays.
2. **Version Info**: this is not chosen by the user, but defined by which XSD's, XSL's and XSQ's the XQL file on the Qualingo target system references. Of course users may call different versions of the XQL document, but the XQL document itself decides which of the defining versions of the related documents it needs. Just see it as a computer that has all versions of its program still online, and just adds new versions to work out the kinks. Old users never get thrown off their version, although they may be warned that a newer version exists.
3. **Config Info**: This is basically addressed by the XSD's, XSL's and XSQ's themselves, in that they nest other documents. This is not for the user, but to the developers creating the program this ability is key, because this is how they create system classes in Qualingo.
4. **Qualingo**: the essence of a Qualingo Application. This is basically the XQL file that gets called, and the XSQ file it references. The XQL file is called by either a local or external system, and holds references to the definition, layout, and the XSQ file. The XSQ file then holds the various programming aspects, needed to solve the problem. This points to the following three types of:
5. **Design Specs (QDS files)**: Since design specs are included in the files, they can be referenced one on one by the source code of the system: instead of having to refer to external links to other programs, without hardwired checking, the design specs become more like inline documentation. No more searching through the requirements in some other program, just point to the right XQL and give it the 'Display Requirements' request.
6. **Source Code (QSC files)**: As you might have guessed by now, source code in Qualingo is a hybrid mix of seven types of documents, which in ideal situations describe one class of system at a time. Notice I'm saying 'system' and not 'object' or 'class': I'm hoping my document has given you plenty of hints at the differences between the them....
7. **Test Specs (QTS files)**: Basically, these are defined by the requirements, and should be coded by the developers along with their implementation of the code. Instead of having this done by separate testers, this approach guarantees better matching of the tests to the code, also in maintenance: if a developer changes something that breaks a test, he or she is immediately alerted to that fact. There are multiple levels of testing: the system itself, the interaction between systems, and even connections to other computers that take part in a cooperative effort.

There are already plenty of programs allowing us to develop XSD's and XSL's. Any one of the companies marketing these could be approached to work out an environment where we can also edit the remaining types of files, thus coming up with a programming environment for Qualingo. Of course quite an effort must be spent first in defining the requirements for such a development environment, whether or not we decide to shoulder the entire programming effort ourselves....

This document is Public Domain, and may be used by anyone for any purpose, except patenting....

QL System

In order to take you towards the architecture for a QL System, we are going to drill down from the software description mentioned above. So first of all, we need to define what a typical QL system would do, if let loose on the Web. And as the Tom Tom club is just now singing: "What are words worth?" But I'm afraid we're going to need a few to get this thing of the ground!

Qualingo System Software

Remember, it is basically a web server with a twist. And like any web server, it can also be called from the local host system. But its functionality is neatly described in the SevenSphere on the right here, starting at the red sphere with the XQL Parser, and ending at the XQL Composer which eventually sends back the reply.

1. The XQL Parser is basically just like an XML Parser, but it can take three definition documents instead of just two: XSD, XSL and XSQ fully define the functionality of the software.
2. Inline testing is always executed, and will perform basic validations like variable range checking and stuff. These aren't the testing aspects mentioned earlier, but the normal stuff already done with just XSD and XSL documents today. The only addition made here, is that XQL documents may also have definitions which override those in the XSD documents.
3. Next up is the Integrity check, which is basically the execution of the defined testing methods for the various system application objects. This may be organized into various levels, which decide just how rigorously the system gets tested during its running of the program. For development it would be advisable to go into full integrity level, where normal execution might only switch on the most basic self tests that take far less time.
4. The Qualingo Processor then executes the program, based on the domain object model that the parser created. By then the values for all variables are already within programmed boundaries, or have resulted in an error which will be reported back to the calling system. The same goes for any condition that caused the Integrity check to fail.
5. After processing, the system has determined which parts of the code form the answer that needs to be sent back to the client. The XQL Composer turns these into valid XQL code, and sends the reply back.
6. Finally, the system heavily depends on a database that is set up to serve all the other components with the various document streams they need. This database serves all the types of documents needed in this approach, that are on the same system. In essence, this database might also hold proxy entries for documents which it actually retrieves from other servers if need be.

In order to run such software, we will require a piece of hardware that is somewhat more suited for the task requested of it by the system. Although right now we can probably go with traditional processors and PC architecture, the move to multi-Gigabit Ethernet connections may well require an architecture that is more up to the task of streaming bits at billions per second, and probably more in a few years.... Clusters of QL Systems may well be the way to go here.

This document is Public Domain, and may be used by anyone for any purpose, except patenting....

QL usage by Example

In order to make the use of this QL system a bit more accessible, we are going to dedicate this chapter to an example of a QL program execution, as the user would experience it. This is different from the development of a QL Application, which will be tackled in the next chapter.

1. QL should be accessible via any browser, which basically takes care of the interface we humans require to interact with the system. The entry point the user needs to access the application is an XQL file, for which he or she only needs to have the URL. It will look like '*ql://www.qualingo_based_system.nl/qualingo_application_name'*. Sometimes even added parameters may be attached to it, but this will make things more difficult for now. Let's just consider this an example where we start a program without parameters, and then proceed to tell it what to do.
2. The QL Base System (on either server or client, depending on the settings) will load that XQL file into a session for the user who just requested it. It will also look up the XSD, XSL and XSQ files, which respectively bring in the datatypes, layout info and program code aspects.
3. Let's assume the server does the execution of the Application for now. This implies it will basically run through the cycle in the diagram above, and will send a reply XQL document back to the client. Since this is also containing XSD, XSL and no XSQ references, the browser will know how to present the next screen in the browser. But since the server already did all the computing, it will just perform a display effort. Any system could do this, because then the XQL document is nothing more than an XML document by another name.
4. If the server retrieves the needed XQL file and sees it is client side programming, it simply sends the file back to the client. The client then has to be a QL Based System as well, and will do just what the server did in the previous step....
5. Either way, we end up with a response XQL document, which the browser knows how to display.

Of course this approach uncovers just the top layer, where especially the execution stage has been given way too little appreciation. In order to rectify this, we will bring in an example of an XQL file, and make ourselves a simple programming example:

```
<?XQL version="1.0" encoding="ISO-8859-1"?>
  <application type="client side process">
  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
     <xsl:stylesheet version="1.0"
           xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
     <xsq:sourcecode version="1.0"
           xmlns:xsl="http://www.ql_one.org/2012/hello_world">

     <xs:element name="screen 1">
         <xs:element name="message" type="xs:string" value="Hello World!!!"/>
     </xs:element>
  </xs:schema>
</XQL>
```

Simple enough, ain't it? A client would for instance call this at "ql://localhost/hello.xql". Since it is client side programming, the server will just return the above document to the client, so it can execute it. Based on this, the client can then retrieve the XSD, XSL and XSQ documents, in order to perform the processing in accordance with these documents. Of course, when server side programming is the case, the server will process, and will only return XSD and XSL in its

This document is Public Domain, and may be used by anyone for any purpose, except patenting....

response, since the client obviously doesn't need the more verbose XSQ file with all the information in it. Now the XSD and XSL documents are quite well-known in structure, and there are enough examples on the Web, the most official definitions being hosted at www.w3schools.com.

But now we get to the crucial part: what would the XSQ document look like? We've seen the satellite concepts it must cover, so the top level isn't all that difficult:

```
<XSQ>
   <code>
      <version versionNumber="1.1.1.1"/>
      <!-- config defined at XQL level by the included XSD and XSL document -->
      <sourceCode>

      </sourceCode>
      <testMethods>

      </testMethods>
   </code>
   <data>
      <!-- data may come from XQL file or be included here -->
      <designSpecs>

      </designSpecs>
      <programData>

      </programData>
   </data>
</XSQ>
```

Yes, you've seen it right: even within the XSQ file, we discern between code and data, just to make sure no one messes them up! This just serves to make the Integrity check a viable target. Now I could work on this some more, but most of you readers out there will not be happy to be talked through programming issues by a guy whose passion isn't programming... I'm more into architecture, the kind that makes systems like ***us*** work: where Andrew just talked Portia into dumping her soon-to-be groom in favor of him, where the boundaries between us and our creations blur. Does it matter whether we become more like technological wonders, or our machines become more like us? Do we need to make it a legal proceeding to determine where and when we will be comfortable with our mechanical friends as every bit the free-willed and well-designed beings they will eventually become?

In that sense I absolutely identify with Andrew in Bicentennial Man (I even carry his name, since mine is André in our reality): like he wants to become more human in order to reach the goal in his life of becoming accepted as a human, I foresee myself becoming more machine-like: I do not know the details of how this will happen, but my donor card already showed me fit for research purposes years ago, long before I realized this to be one of the pieces of truth I know absolutely: I will reach the age of 94 in 2057, at which time some sort of transition will take place. But at this moment I have strong feelings that it will not be an event which people usually call dying....

And no,, this is not about fear of death: it is far more the understanding that death or any other transformation is merely the arrival in a still larger playground!

This document is Public Domain, and may be used by anyone for any purpose, except patenting....

Appendix A: the Simple Solutions

Keep It Simple Sir....

Any action that leads to too much complications needs to be rethought: rather than settling for a complication, try some of the other rules written down here to arrive at something simpler .

80 / 20 % rule

Popularly said: "Screw the other 20, 80% is plenty!" Or in more decent wording: it takes you about 20% of your available time to come up with 80% of the solution, and the remaining 80% of your budget and time, to finish remaining 20 percent or worse....

Divide and Conquer

If the problem is too complex, chop it up into more easily digestible pieces. The SevenSphere is a nice tool for this, because it keeps in mind the rule that you shouldn't chop it into too many pieces. And if you do find you have too many aspects, try dividing them over several SevenSpheres, like two or three. Try and keep your balance where the weight of the various chunks is concerned.

Stepwise Refinement

Work your way down into the finer details from above, rather than trying to start with the solution. This is basically a more refined version of Divide and Conquer, where we focus on the less optimal parts of our solution, and try to make them better equipped to do their job. As a rule, starting a piece of code with just a bunch of comments outlining what needs to be done is a great way of making sure there is at least a modicum of documentation that says what you are trying to achieve.

Occam's Razor

Simply stated, if you have multiple ways to do something, try to figure out which is the simplest. This usually also tends to be the most succesful variation as well. Great examples of this can be found in the movie called 'Contact', with Jodie Foster.

Thinking outside the Box

For those who take this concept for what it reads as, there should be a box that contains the problem. For those with more open minds it simply means that if looking at it one way does not work, you just shift your focus to a viewpoint that you think has not gotten the attention it deserves. In most cases that uncovers the blind spots in your previous vantage points, and leads you to new insights....

The 7 item rule

This is a rule of thumb for those who are all thumbs anyway: never bite off more than you can chew, because the human mind can only handle seven concepts at any one time. So I guess bytes are already to complex to dissasemble for most humans. Luckily they can grab the higher concepts composed of bytes by simply naming them into something else!

This document is Public Domain, and may be used by anyone for any purpose, except patenting....

Appendix B: the Three Laws of Robotics

Although Isaac Asimov was a SciFi novel writer, he absolutely could see how things would eventually fit together in our society which is growing towards a symbiosis of Humans and Technology, and most of all our interaction with Free-willed mechanical men and women. Since in his time he foresaw problems with that attitude, he formulated the Three Laws of Robotics to soothe the readers' minds:

1. ***A robot may not injure a human being or, through inaction, allow a human being to come to harm.***
2. ***A robot must obey the orders given to it by human beings, except where such orders would conflict with the First Law.***
3. ***A robot must protect its own existence as long as such protection does not conflict with the First or Second Laws.***

Basically, he made robots into guardians of men and women, since the Free Will they have is essentially limited by their being hardwired to obey these directives. Hinting at the fact that even humans might not be adversely affected if they took heed of these rules was of course something that has often been done since then...

What most people don't realize, is that two more requirements are implied by the Three Laws:

4. ***Robots or Androids need to be able to determine what would harm a human.***
5. ***And they would need to be able to determine what would actually harm them.***

In that, we can easily quote the Terminator: "*I have detailed files on human anatomy*", which may have been intended to make him a more effective killing machine, but once he chose sides with the humans made him a more effective bodyguard!

This document is Public Domain, and may be used by anyone for any purpose, except patenting....

Appendix C: Programming concepts Explained.

This appendix simply explains some concepts programmers take for granted, but which non-programming folk may be oblivious to...

Artificial Intelligence

The concept where intelligence (or learning) is simulated by mimicking a neural network much like our brains. This is supposed to be able to learn from its inputs, but requires quite powerful computers to actually reach something similar in complexity to the most evolved minds of the leading mammals, dolphins. Other scientists see intelligence and consciousness as emerging properties, which will automatically emerge as the complexity of systems reaches certain levels. In that light, we might see the Web as one huge system having far more complexity than a single human mind, since a fair portion of its over three billion systems are running 24/7....

Genetic Programming

This approach to programming difficult problems mimics Evolution by starting with several candidates for solving a certain complex situation, and then scoring them on how well they perform in their task. By weeding out the weakest and slightly mutating the best candidates, we work towards a candidate which is best adapted to the problem at hand. This approach is best suited for problems in which input data and output data are available in sufficient amounts, like for instance the reactions of the stock markets with regard to certain events.

CGI: Computer Generated Imagery

I'm looking at an almost perfect sample right now: the movie Beowulf has been totally made via CGI and Motion Capture, with maybe even LIDAR added to the mix to capture complex images from natural sources, like say the forest the main characters are now walking through. The detail in these movies (of which Toy Story was no doubt a notable success) is becoming ever more complex, right down to the sewing on the warriors clothing, or the details in the horn he is carrying. It is the same with the major disasters that befall many cities in movies: Google maps data is so detailed, that CGI artists can just import it and use the data to their various 'what if' scenarios. Finite Element Computing is used to simulate the behaviour of gasses, structures, debri and even complete galaxies. I cannot prove my claim, but it is my solemn belief that CGI has been way more advanced than moviemakers will have us believe: certainly, once you have the basics, adding more detail will become a task of simply defining the detail in the 3D models, and using a lot more computing power to render the end result. With lots of people playing ever more 3D role playing games, and partaking in distributed processing projects like SETI, it is easy to produce movies of a quality that is not discernible from the real thing, unless we have insider knowledge... (and releasing less detailed movies every now and then will have us thinking that is the state of the art)

This document is Public Domain, and may be used by anyone for any purpose, except patenting....

Model, View, Controller

This is basically the approach where the programming code is split up in three distinct parts, that have a stable relationship with one another: the Model is the most basic part of the code, which keeps track of the actual program data: if we were to make a Diary program, then this part of the code might consist of diary entries, which in turn might be made up of paragraphs and other, smaller parts. Images would also be stored here. The views on the other hand specify how the data from the model should be presented to the user: one view might give him or her the possibility to edit the data, another might be used to export the data as HTML. Controllers in the end allow the users to manipulate the model and the various views. Where model and viewers may be similar if not completely the same across different computing platforms, the controllers are usually very dependent on the platform used.

Object Action Principle

This is just a smart idea when you are programming in an Object-Oriented fashion: make sure the user will always be able to select which object to use first, and then allow him or her to select the action he or she wants to perform on that object. This way the interface stays simple, and people have little trouble using it. Thus, it is preferable to use right-click and context menus, instead of select the object and then select the acton required from a window menu (here the user choses both the object and the menu, which might confuse things)

This document is Public Domain, and may be used by anyone for any purpose, except patenting....