

Thinking about Thinking

Roots of the Approach

The work leading to this course was motivated by wondering why, in software engineering, there are some people who are one or two orders of magnitude more useful than most people. If this was true of bricklayers, the building industry would be very keen to find out why. The problem of course, is that one can film a bricklayer, and later analyze what is happening at leisure. One cannot even see what great programmers do, and for some reason they cannot explain what the difference is themselves, although most of them wish they could.

We knew that the elements of industry best practice alone are not enough. Management commitment to investment and training are not enough. Innovative Quality programmes that explicitly include holistic concepts such as Robert Pirsig's *Zen and the Art of Motorcycle Maintenance*, which much of the industry would consider too radical to experiment with are not enough. Years of experience are not enough, nor are years of academic study.

There seemed to be only one way to continue the investigation if an industry dedicated to objective metrics had not found the X factor: we needed to look at the subjective experience of the people concerned.

Achieving understanding of what was happening took a long time, although the key ideas are known to most of us already. On the way we learned a great deal about the mind set of successful programmers, and were able to develop exercises that certainly helped many people.

Thus the material in this course has developed over several years, and is a mix of ideas empirically justified by experiment and later fitted into the logical picture, and material derived from the logical picture.

This course aims to address the element of 'experience' or 'judgment' referred to almost everywhere, but rarely described. Many of the topics are the kind of thing programmers discuss over a beer. Perhaps it is odd that no-one tends to ask how the issues that programmers see as most important relate to the 'formal' structures of modern engineering. Here, we do just that.

We have found that once we get into the swing of this, most programmers find they have an opportunity to put issues they have wondered about for years into a clear work context, together with their colleagues. We therefore ask you to relax, because you are supposed to be doing this, and have an enjoyable time!

Mapping and Software Engineering

Software engineering is in a terrible pickle. The so-called 'Software Crisis' was identified in 1968, but despite thirty years of effort, with hundreds of supposedly fundamental new concepts published, the general state of the industry is horrific. Projects run massively over-budget or collapse entirely in unrecoverable heaps. Estimating is a black art, and too many projects solve the customers' problems of yesterday, not today. The technical quality of most code is dreadful, leading to robustness problems in service and high maintenance costs. And yet within the industry there exist individuals and groups who enjoy staggering, repeatable successes. There are many ways of measuring the usefulness of programmers, but some are rated as over a hundred times more useful than most, by several methods of counting. If only the whole of the industry performed as well as the tiny minority of excellent workers, the economic benefits would be immense. If it were possible to write sophisticated, reliable software quickly and cheaply, the intelligence of society would increase, as everything from car sharing to realistic social security regulations became possible.

Within this model, the problem can be understood. What is presented as socially conditioned conventional thinking (called packing) is based on action. To be a good bricklayer, a packer must know what a bricklayer does. What does a programmer do? The most developed packer model of programming is the concept of the Software Factory. In this, statements of requirements from customers go in one door, and are processed by workers following procedures written

down in manuals. When the production line has done its work, programs come out of the other door. It works in car factories.

The trouble is, the analogy with a car factory is sloppy. Most of the car factory is filled with workers using machines to make cars, but around the back there is a little office where another worker determines how to use the resources of the factory to make as many cars as possible, all alike.

The workers in a software shop are not like the factory floor workers. The shop floor workers can be replaced with robots today, but the person who uses creativity to set up the factory is still needed. The programmers are doing the same job as the office at the back of the factory, and we cannot learn anything about what happens in there by playing at car factory shop floors.

Packers who advocate uncompromising process-based Software Factories are in fact claiming to be able to implement an Artificial Intelligence that simulates a production line designer, and to be able to do it by using humans pushing bits of paper around as their computer. Unfortunately, packing is just not up to the job of understanding software production, and gets terribly confused. This means it says some very silly things sometimes.

To understand what programmers really do, an alternative strategy of thinking (called mapping) is necessary, because programming is essentially a process of internalising the capabilities of the system, the nature of the problem, and the desire, and capturing the insight in a programming language. It is all about exploring the details of our desires, and understanding them in such a way that we can keep track of all the complexity. Mapper problem collapse can produce beautiful, tiny, elegant programs with no room for bugs in them. Mapping can do programming, but how it does it cannot be explained in packer, action-based language.

Packers therefore assert that hackers are 'irresponsible' and discount their work, saying that complexity is inherently not understandable and we must develop ever more complex procedures to abdicate our responsibility to.

Fortunately, many organisations' managements continue to foster reflection on grounds of personal intuition and empirical experience, without any justifications to place on action-based balance sheets. This is a difficult thing to do, but is the only reason anything gets done.

It is important to recognise that mapping is not another procedural methodology to be applied in a packer mindset. It is a different way of looking at things altogether. It is necessary to convince yourself that it really is possible to take personal responsibility for an undertaking instead of abdicating in favour of a procedure.

Programming is as near to pure mapping as you can get outside your skull. This is why it is fun. It is endless discovery, understanding and learning.

Object Orientation (OO) and mapping have an interesting relationship. OO is often seen in very different ways by mappers and packers. The mapper's map is a kind of object model that has a rich variety of objects and associations. Mappers see OO as an elegant way to design software once they have understood the problem. Packers seem to see OO as a way of wandering around the problem domain and creating software objects, then just wiring them up as they are found. Thus OO is taken to be a procedural mechanism for getting from problem to program without the intervening understanding. If it were possible to capture absolutely every aspect of the problem domain and one did not care about efficiency, this approach might even work. But in fact, good taste is always needed in object design and categorisation, because it is necessary to design software objects that have a good mapping with real world objects, but can be plugged together to construct a viable computer system. That takes understanding, and is a strictly mapper job. This explains the OO projects that grind to a halt with the product a tangle of real and utility objects using multiply redundant addressing schemes to communicate *via* Object Request Brokers, with no clear conceptual integrity in instantiation, flattening and journaling. Packer programmers often have so little control over their objects that they lose them, and end up with memory leaks that cause the application to fail. The packer solution to this is to buy a memory leak detection tool, rather than to regain control of their objects so that everything else works properly too.

Mapping and TQM

After WWII the Americans sent Dr. J. Edwards Deming to Japan to help sort out their manufacturing industry, which was an odd mix of the medieval and industrial ages, and war shattered. Deming introduced ideas including collecting statistics from the mass production activities, asking the workers that performed those processes to think of way of improving them and making sure that each worker understood what he or she was doing. These ideas were later developed into what we today call 'Total Quality Management' (TQM).

The results (we are told) were extraordinary. Within a generation, Japanese industry soared and moved from building bicycles in sheds to worldwide dominance of high-value industries like building ships, cars and electronics. 'Japanese Methods' were reimported to the West, and have been institutionalised in ISO 9001, an international 'Quality' standard that business has spent a fortune on, and which focuses on defining procedures for everything with lots of ticking and checking. The expected benefits have not yet been seen in general, and yet some organisations that have applied the work of Deming and his successors have seen staggering benefits.

Recognising the importance of mapping suggests another way of looking at what has happened here. Mapping can certainly be reawakened by trauma. One possible way to traumatise a person might be to:

1. Nuke them. Twice.
2. Rip apart their rigid, predictable feudal society.
3. Tell them the invader will be coming around tomorrow.
4. Leave them nothing for supper.

To eat tonight, this person is going to have to reawaken his ability to be imaginative. So by the time Dr. Deming got to Japan, the population he was to work with was already mapping. All of them. At once. Perhaps all Dr. Deming needed to do was take a leaf out of *Bill and Ted's Excellent Adventure*, stand on a tea chest and shout, 'Be most sensible to each other!'

When that worked so spectacularly, Dr. Deming and his colleagues would have naturally been impressed, and so started to work on methods that their work-force could use to get even more sensible, creating a culture which is an industrial powerhouse, but has the hidden requirement that it only works for mappers!

During the early reintroduction of 'Japanese Methods', mapper people from Japan returned to America, and with the characteristic enthusiasm and habits of mappers they showed the American workers how to ask interesting questions about their work, collect data, interpret the data wisely and improve processes. They showed them how to write down a description of their jobs, look at those descriptions and see if there might be any problems lurking in there.

It worked wonderfully, but again it was accidentally teaching people mapping that had done the real work.

When the TQM ideas became widespread, the accidental teaching of mapping just got lost. The ideas were sold to packer industry on their results, but packer industry just couldn't see the key bits of what they'd bought - the wisdom and reflection stuff.

Even creative managements of high tech industries can be thwarted by the communication barrier. To many of their workforce, the manifest artifacts of TQM look just like the stuff that Frederic Taylor, the father of scientific management threw about the place. Taylor gave us mass production before we had robots, by getting people to do the robots jobs. Perhaps that is an odd way of looking at it, but at Los Alamos, they simulated spreadsheet programs by sitting secretaries at grids of desks with adding machines! He was such a control freak that he used to strap himself into bed every night to counter his morbid fear of falling out. His slogan was, 'Leave your brain outside and bring your body indoors'. Our culture, from schools to legislation and concepts of status, is still riddled with Taylorism. In this situation, the worst case result of introducing TQM without an explicit understanding of mapping will be dumb Taylorism. The best will be that we are confused about why we do what we do.

In some organisations the results have been tragic. There is an obsession with micro-accounting, dumbing-down and writing poorly-designed job descriptions that are taken as absolute behavioural tramlines. Everything has to be done on the adversarial model of packing, not the intended co-operative model of mapping. ISO 9001 auditors appear in the workplace

and perform swoop raids on the paperwork, aiming to catch workers out in trivialities of paperwork regulations, like a scene out of Kafka. In some organisations, workers become more concerned with avoiding blame for microviolations of paperwork regulations than the work at hand, which becomes completely obscured by the intervening rituals. Think of Feynman's story of the six lines on the STS SRBs! Some people actually think that this is the idea!

Good TQM captures experience in the workplace and condenses this knowledge into lists of things that are worth considering. These checklists simply remind mappers of issues they should use their mapper common sense to consider, where appropriate. The packer corruption is to regard the job as ticking the boxes as quickly as excuses can be found to do so. How much consideration is 'sufficient' to a packer?

As the proceduralist orgy has progressed under the banner of 'Quality' in too many places it has driven real quality, which is about doing one's imaginative best to do the best possible job for the customer, completely out of the window.

Ironically, there are some organisations (all of whom seem to be able to make intelligent use of information technology) that have invented a kind of 'real proceduralism'. Telephone banking companies have dropped the pretense that they are offering an intelligent service from real people, and openly acknowledged the anonymous, proceduralised nature of their business. This has allowed them to think about their procedures clearly, and produce very good procedures that satisfy customers' needs twenty-four hours a day at low cost. This contrasts favourably in many people's eyes with an offensive counter-clerk performing a caricature of a pompous Dickensian undertaker and behaving as if the ridiculous 'regulations' he is applying are the customer's problem and not his.

Very successful financial organisations recognise that there are procedures that computers do well, and judgements that experienced people do well. They analyse their markets with mathematics run by the computers, and leave the final calls up to the people. They can use different criteria to describe the jobs of both aspects of the overall system, and evaluate the effectiveness of different algorithms and traders.

This gives an opportunity to try a mappers' technique. If we have 'Real TQM', 'Fake TQM' and 'Real Proceduralism', can we say:

Real TQM	Real Proceduralism
Fake TQM	Fake Proceduralism

and ask if there are any examples of 'Fake Proceduralism': organisations that swear blind that they are mindless automatons while actually indulging in a frenzy of mapping? What about the British Army's journey to Port Stanley in 1982? Remember, an army is an organisation that faces particularly difficult challenges. Even those that abhor all conflict can learn how to make their world more co-operative by understanding what makes an army more co-operative. The British Army are Fake Proceduralists? Now that's an interesting mapper way of looking at things, because then we can look beyond the paper and the language and see what the organisation does. The idea that they are all following rules all the time makes the British Army in action hard to understand. Once we realise that there are a lot of mappers in there, following the rules until the moment that they can see they won't work any more, things get clearer. We can also compare the customs of the British Army with the US Army. The Americans have always openly preferred an approach more like the 'Real Proceduralism' of the telephone bankers. They openly intend to do everything by procedure, and get their mappers to write the best procedures they can, in readiness. When this works, it works very well indeed, as in the Gulf, but it is brittle because it does not give the packers using the procedures much room to react to changing circumstances. This leads to inefficiency, as in the Grenada invasion.

The lesson is simple. Without the underlying mapping, TQM turns into a black comedy. With mapping, the Quality stuff can educate and provoke, and the enthusiasm and joy in work that the TQM advocates talk about is nothing but general mapper high spirits!

In this model, the Systems Thinking approach advocated by Peter Senge in *The Fifth Discipline*) can be seen as a collection of useful mapper concepts and techniques, optimised for management problems.

Mandate Yourself!

There are many more packers than mappers alive today. One purpose of this course is to explain effective mapping techniques, but others are to explain why for many of us, our insights do not seem to be endorsed by others. We have to recognise when our concerns as artisan programmers are not understood by packer colleagues, so that we can get them habituated to complex phenomena taking a while to think about. We also have to accept that being right is not necessarily being popular, but that a personal commitment to solid work often brings a more fulfilling and less stressful environment than any ostrich behaviour could.

We must also recognise that it is possible to communicate effectively with mappers, even those who are out of their domain. While accepting that there is a specific communication barrier with some, we must also recognise that with others, communication is often much easier than we might expect.

We must also keep in mind a clear understanding of the boundaries of our own responsibility. When talking to a customer about a subject which he does not seem to grasp the essential points of, remember that our personal, self-imposed goal of finding the best answer does not necessarily mean forcing the customer to accept that answer alone. Any contemplation that throws up one strategy usually throws up several others as well, each with strengths and weaknesses. You can always summarise these, and content yourself with the knowledge that you have done a good job of exploring the options and explaining the choices to the customer. If, with full understanding, the customer makes what you would see as a stupid choice, well how else can the customer organisation learn?

You don't have to save the world, just your bit and as much of the rest as you can reach!

The Undiscovered Country

In Tom de Marco and Tim Lister's *Peopleware*, the authors suggest that gelled teams make great software, and propose that initiatives are taken to assist the social cohesion of teams. Looking at gelled teams, we can see the social ease which they exhibit, and the effectiveness in their work. But add the concept of mapping into the equation, and the picture changes. Gelled teams look much more like groups of mappers, communicating effectively with one another because they can refer to parts of their shared mental map of the situation with a few, perhaps odd-sounding words. (There was once a guaranteed delivery comms buffering subsystem that its creators called the 'Spaghetti Factory'. It was to do with loops of stuff flying unsupported through the air.)

They can't just exchange information about their maps quickly - they can all grab hold of chunks of their maps and move them around. They can move chunks of each others maps around. They can react, as a team, very quickly. They all know what is going on, and they've all thrashed it to death, in their own minds. They don't make cock-ups, and they don't waste time on unsynchronised activity. They respect each other even though they may loathe each others' taste in music, politics and food. The performance gains are breathtaking, as anyone who has had the pleasure of working on such a team knows.

What one has to do is take the time to ensure that everyone has a shared understanding of what is going on, and life can be a more rewarding experience, because one has a sense of success at five o'clock.

Getting into this situation is not an accident, it is repeatable.

Knowledge Packets, Daydreams, Maps and Understanding

As software engineers, we might describe learning as forming associations between referents. The sky is blue. The rain in Spain falls mainly on the plain. We might call these learned facts 'knowledge packets': little bits of truth (or errors) that we possess.

One can go a long way on knowledge packets. Early learning (as directed by adults) for most children focuses almost

entirely on the acquisition of knowledge packets. Things that one should or should not do. Methods for performing tasks. Data to be retained and later recovered on demand.

The trick with knowledge packets is to identify key features of the situation, and determine what action to take. One can get A Levels and degrees, drive cars, even chat up members of the opposite sex by using knowledge packets. Very adept knowledge packet users can fill their heads with megabytes of procedural tax law and become accountants earning six figure sums. Some politicians omit the pattern recognition stage and use a single all-purpose knowledge packet for everything.

Of course, we don't just stack up knowledge packets like dinner plates in our heads. From our earliest years our natural response to gaining each new knowledge packet is to ask, 'Why?'

We attempt to connect up knowledge packets to create a structure within our knowledge, a mental map that gives us understanding of the causes and effects within a situation. This understanding allows us to derive a solution to any problem within the situation, instead of attempting to select a rote-learned response.

In later life, we must spend periods of reflection, or daydreaming, where we trace through the relationships between that which we know. This broadens our integrated map, and allows us to identify structures in the map that apply in different areas. We can then get a deeper map, where what mathematicians call 'isomorphism' provides what software engineers call 'inheritance', allowing us to reapply knowledge.

We rearrange our mental maps to produce simpler expressions, and allow more understanding to be held in the mind at once. When we find a simpler way of looking at things, we find it hard to remember what it was like when the topic seemed complicated, and we ourselves have grown. With understanding, where does the self end and the data begin? With knowledge packets, the division is clear.

We become adept at using techniques in reflection that allow us to explore our maps, and the knowledge packets we have not yet connected. There are likely to be neurological underpinnings to what we do when we reflect, but some kind of abstract pattern recognition activity must be under way. We learn to use our brains.

Without understanding there can be little intelligent action. Without mental maps there can be no understanding. Without reflection, there can be no mental maps, only knowledge packets.

There are computer data structures, called 'ontologies', that hold vast numbers of truths in networks associated by a form of predicate logic. The CYC database for example, can use maps of the meanings of natural language well enough to interpret photograph captions and find examples for pictures needed by journalists.

Mappers and Packers

Or at least, all this should be true. Unfortunately, we are descended from industrial and agrarian societies where one day was very much like another. Efficiency was dependent on getting everyone co-ordinated into simple group activities. On the other hand, there really wasn't much call for inventiveness. We developed social customs that teach people to stack knowledge packets and focus on action. Reflection ('daydreaming') is discouraged during early school. We observe children closely and note deviations from action-based behavioural norms with concern. One even hears parents who are concerned that their children may have physiological abnormalities if they do not wish to play a particular sport.

One cannot easily teach reflection to a child. Unlike the performance of physically manifest task, subjective experience must be discussed.

One cannot easily ascertain if reflection is proceeding well in a person. Only by careful discussion or watching the long-term results of a child's mentation can effective daydreaming be identified.

So there is nothing in our social history that motivates parents or teachers to teach reflection. There is nothing that makes teaching reflection in school a priority.

In fact, the reverse is true. When a child attempts to reflect, the consequent lack of manifest physical activity is chastised.

When questions prompted by reflection are asked by children, they are rarely addressed by busy adults. Where reflection succeeds and understanding is gained, this can become a handicap to the child. If there are another fifteen simple addition sums to do, the child will become bored, be chastised, and labeled as incapable of performing the simple task, although nothing could be further from the truth.

Notice that although adults chastise different effects on each occasion, what the child has been doing in each case is reflecting. Many people have actually been conditioned to think that reflective thinking is, in itself, socially unacceptable!

The traditional story is that thinking is taught at universities, but with a whole degree course of thirty years ago packed into the first year of a modern course in most technical subjects, this rarely happens.

In the workplace, educated people are still regarded as able to think, and indeed all programmers must be able to do it to some extent, just to accomplish anything. We are the amongst the most reflective people in society, but we are still a far from homogeneous group. Some of us are better at it or less nervous about it than others. Again it is not taught, and with the workplace a part of the embedding society, the cultural environment often remains based on knowledge packets and action, rather than mental maps and understanding.

This leads to two distinct groups in society. Mappers predominantly adopt the cognitive strategy of populating and integrating mental maps, then reading off the solution to any particular problem. They quickly find methods for achieving their objectives by consulting their maps. Packers become adept at retaining large numbers or knowledge packets. Their singular objective is performing the `correct' action. Strategies for resolving `hash collisions', where more than one action might fit a circumstance are *ad hoc*.

How to Regain Mapping

Our species' principal advantage over others lies in our generality. We can survive a greater range of temperatures than any other creature, but more importantly, we are inventive. Arthur C. Clarke and Stanley Kubrick celebrated this inventiveness in the famous `thigh-bone to spaceship' fade in the film *2001*.

We are all mappers, no matter how little we use the faculty. Those of you who spend time on solitary walks, in heavy metal bars or whatever does it for you, feeling somehow uncomfortable until suddenly a penny you didn't even know you were looking for drops, are already operational. You know who you are!

Otherwise, there is an easy way to start. So easy kids that are trying really hard to be natural mappers often discover it. Get yourself an imaginary friend, as smart as you are, but totally ignorant of the world. Whatever you feel you could relate to - you don't have to tell anyone that you find it easiest to talk to the 1960's cartoon character `Astronut' hovering about in his little UFO with a VHF television aerial on his head. Or maybe Sean Connery's canny medieval investigator in *The Name of the Rose* would be more fun. Explain everything to your imaginary friend. What it's for. Where it comes from. Where it's going.

At first your full attention is required for this exercise, but after a while the logic between knowledge packets becomes as automatic as driving, and your attention is only drawn to unusual situations: pieces of your map that need filling in or contradictions resolving. It works. With your maps building, discussion of techniques is possible, because we all know what we are talking about.

The Ways of Mappers and Packers

It is a surprise to discover that there are two distinct states of mind around us. It is similar to the experience of learning that someone you've known for months is illiterate. At first you are astonished: this cannot be possible! But then you realise how someone else can live a life very different to yours, that looks superficially almost the same.

In this section we look at traits of the two strategies. As we do so, many of the woes of the modern age, particularly in high tech disciplines, will come into a simple picture - the mark of a useful theory! Remember, most people, be they mappers or packers, have no reason to believe there is any other state of mind but their's.

What is packing? Well, you just stop yourself asking 'Why?'. You never really clean up your map of the world, so you don't find many of the underlying patterns that mappers use to 'cheat'. You learn slower, because you learn little pockets of knowledge that you can't check all the way through, so lots of little problems crop up. You rarely get to the point where you've got so much of the map sorted out you can just see how the rest of it develops. In thinking-intensive areas like maths and physics, mappers can understand enough to get good GCSE grades in two weeks, while most schools have to spend three years or more bashing the knowledge packets into rote-learned memory, where they sit unexamined because the kids are good and do not daydream. It really isn't a very efficient way to go about things in the Information Age.

With no map of the world that checks out against itself and explains just about everything you can see, it is very hard to be confident about what to do. The approach you have to take in any situation is to cast about frantically until you find a little packet of knowledge that kind of fits (everything has a little bit of daydreaming at its core, but the confused objective is to stop it as soon as humanly possible). Then you list the bits that kind of fit, and you assert that the situation is one of those, so the response is specified by your 'knowledge'.

Your friend has happened to grab another packet of 'knowledge' and so you begin an 'argument' where your friend lists bits of your knowledge that don't fit and says that you are wrong and he is right, and you do the same thing. You don't attempt to build a map that includes both your bits of knowledge and so illuminates the true answer because you don't have access to the necessary faculty of mapping, and anyway, without the experience, it is hard to believe that it is possible in the time allowed. Being devoid of the clarity that comes from a half-way decent map, you would rather do something ineffective by the deadline than something that might even work. Then when things go pear-shaped you say it is bad luck.

The consequences go further. Not having a big map means that you often don't understand what is happening, even in familiar settings like your home or workplace. You assume that this means that you do not possess the appropriate knowledge packet, and this may be seen as a moral failure on your part. After all, you have been told since childhood that the good acquire knowledge packets and stack them up in their heads like dinner plates, the lazy do not.

You are also overly concerned about certainty. Mappers have a rich, strong, self-connected structure they can explore in detail and check the situation and their actions against. Logic for them is being true to the map, and being honest when it stops working. It's not a problem, they just change it until it's 'logical' again. Without mapping, you have to use rickety chains of reasoning that are really only supported at one end. Because they are rickety you get very worried that each link is absolute, certain, totally correct (which you can never actually achieve). You have to discount evidence that is not 'certain' (although tragically it might be if your map was bigger), and often constrain your actions to those that you can convince yourself are totally certain in an inherently uncertain world.

The issue of certainty then becomes dominant. People are unwilling to think about something (erect a rickety chain) unless they are 'certain' that the 'procedure' will have a guaranteed payoff, because that, they believe, is how the wise proceed.

You become absorbed by the fear of being found to be 'in the wrong', because of the idea that the 'good' will have acquired the correct knowledge packet for dealing with any situation. The notion that the world is a closed, fully understood (but not by you) thing kind of creeps in by implication there. The idea of a new situation becomes so unlikely that you rarely spot one when it happens, although mappers notice new situations all the time. Your approach becomes focussed on actions that you cannot be 'blamed' for, even though their futility or even counter-productivity is obvious. You insist on your specific actions being specified in your job, even when your map is already easily good enough for you to accept personal responsibility for the objectives that need to be achieved, which would be more in keeping with your true dignity.

Some people have so little experience of direct understanding, produced by mapping over time, that they cannot believe that anything can be achieved unless someone else spells out in exact detail how to do absolutely everything. They believe that the only alternative to total regimentation is total anarchy, not a bunch of people getting things done.

Now, if you are used to talking to your imaginary friend about your map of the world, and keep finding holes and fixing them, you don't become very attached to the current state of it at any particular time. You do sometimes, if you find an abstraction that was a wonderful surprise when you got it and has been useful, but now needs to go. It's always important to remember that the fun only adds up: if finding something was fun, finding something deeper is even more fun. Generally though, you don't mind your imaginary friend knocking bits off the map if they don't work. So you don't mind real friends doing it either! When you see things in different ways you try to understand each others' maps and work through the differences. Two messy maps often point the way to a deeper way of seeing things.

Great thinkers are mappers. They rarely proceed by erecting edifices of great conceptual complexity. Rather they show us how to see the world in a simpler way.

Mappers experience learning as an internal process in response to external and self-generated stimuli. Packers experience learning as another task to be performed, usually in a classroom, using appropriate equipment. Particularly in the early years, efficient mapper learning requires internal techniques for exploring conceptual relationships and recognising truths, while efficient packer learning focuses on memorisation skills.

Aspects of mapper learning require higher investment than packer learning, and this has consequences. An emphasis on succinct, structured knowledge means that low structured off-topic considerations can displace disproportionately larger issues from a problem the mapper is contemplating. If a child is trying to understand a new idea in terms of as much as possible of what is already known, then likely the child's awareness will be spread over as much 'core knowledge' as possible already. The requirement to then consider the question 'Shall I take my library books back today?', bringing with it conceptually networked questions such as 'Where is my satchel?', 'Will it rain?', 'Will it rain tomorrow?' and so on is an imposition on the mind that a packer child would simply not experience in apparently similar circumstances. The packer child simply never has (for example) the form of the flows resulting from economic supply and demand curves (which might also actually be the same representations that are used to hold, say, parts of thermodynamic understanding) floating about to be displaced by a simple question about a library book.

Accepting a fact and being ready for the next is also a different process in mapping and packing. The mapper mind must explore the fact and compare it against core knowledge to see if it is a consequence that already has a place in the mapper's conceptual model of the world, or if it is in fact new fundamental knowledge that requires structural change.

Mappers are likely to be much more aware of the comparative reliability of information. Whereas packers tend to regard knowledge as planar, a series of statements that are the case, mappers tend to cross-index statements to verify and collapse them into more profound truths. Mappers are more likely to work with contingent thinking of the form: 'If X is true then Y must be true also, Z is certainly true, and W is nonsense although everyone keeps saying it is the case'. Mappers are likely to be concerned about the soundness of packer reasoning.

An aspect of packer thinking that drives mappers up the wall, is that packers often seem to neither seek out the flaws in their own logic, nor even hear them when they utter them. Worse, when flaws are pointed out to them, they are likely to react by justifying following logic that they cheerfully admit is flawed, on grounds of administrative convenience. The evidence of their own senses is not as important as behaviour learned through repetition, and they seem to have no sense of proportion when performing cost/benefit analyses. This is because packers do not create integrated conceptual pictures from as much as possible of what they know. The mapper may point out a fact, but it is one fact amongst so many. The packer does not have a conceptual picture of the situation that indicates the important issues, so the principal source of guidance is a set of procedural responses that specify action to be taken. The procedure that is selected to be followed will be something of a lottery. For the mapper, one fact that should fit the map but doesn't, means the whole map is suspect. The error could wander around like a lump in a carpet, and end up somewhere really important. Both parties agree that they should do the 'logical' thing, but two people can disagree about logic when one sees relationships that the other has only ever been dissuaded from seeing.

Mappers have lots of good ideas based in profound insights into relationships that packers rarely have the opportunity to recognise.

Part of mappers' extraordinary flexibility and learning speed comes from the benefits of seeking understanding rather than data, but some of it comes from the sheer amount of playing with a topic they do. It is quite usual for mappers to spend every spare moment for a week wandering around a topic in their heads, and then spend all weekend focused on it. Mapper focus is a terrible thing. A few hours of it can produce breathtaking results where a team of packers could strive for months. Every IT manager who has ever had an effective mapper around knows this.

Mappers have a linguistic tendency to want to talk in terms of the form of the concentrated knowledge they reduce experience into. Although mappers often use different internal representations of a sphere of discourse, they are adept at negotiating mutually agreed terminology at the onset of discussions between themselves, and this is one way that mappers are able to recognise one another. Mutual recognition occurs because of this series of transactions where one party traces a route through the map, stops, and invites the other to pick up where they left off. The objective of the exercise is to align mental maps, but it also reveals the presence of the other's map in the first place!

Mappers advocate changing descriptions and approaches often, because they see simplification benefits that are of high value to understanding, and whose map is it anyway? In social or administrative situations, this can cause confusion because the mapper does not realise that the packers do not have a map that they can move around in chunks. Mappers see packers as wilfully ignorant, packers see mappers as confused. In software engineering contexts, this failure of communication leads to arguments about `churn'. The mapper wants to move from a large mass of software to a smaller one that is more robust because of its necessary and sufficient structure. The packers are not practiced at seeing the proposed new structure, and see only a maniac who wants to change every single file in one go.

Mappers have a direct, hands-on awareness of the effectiveness of their reflections and so, in most areas, they have a sense of the universe in some unseen way `playing fair' with them, even rewarding them with wonderful surprises when they look deeply enough. This often gives rise to a `spiritual' or `mystical' element to their character, and often to unusually high spirits, even in situations where packers are despondent.

Mappers ensure that the known elements of a problem are held in their minds, before embarking on it. They draw on their own strength of character to find the motivation to do the hard work involved in keeping their background explorations going. To achieve a solution to a problem, a mapper engages all his or her strengths, and is rewarded with elation or a sensation of betrayal if things do not work out well. Mappers are `passionate' about `dry' subjects.

Mappers excel at conceptually challenging work such as complex problem-solving with many inter-related elements. They can perform tasks requiring insight, or imagination, that packers simply cannot do at all. Best quality software engineering, mathematics and physics, with genetics emerging as a likely area of unique contribution, are amongst mapper challenging science disciplines. Amongst the traditionally recognised arts, poetry and music are areas where the mapper faculty for manipulating structure is of particular benefit, although there may be value in redefining the `Arts' as what mappers do well. The very power of great art is only available to mapper thinking, because the artist uses a tone of sound or light, itself representative of nothing, but triggering the recognition of a deep structure. Pointing out the structure can then bring to mind instances of that structure, and the artist has added to the audiences maps!

All these differences are simply consequences of one person having a big map built by a great deal of disciplined daydreaming, and the other not. That these profound differences between two clearly distinct groups of people exist is the major surprise of the approach proposed. It means that it is very unlikely that either kind is likely to have any appreciation of the other's state of mind.

Packing as a Self-Sustaining Condition

We live in an action oriented society. It's been that way since we invented agriculture and developed a stable environment in which the tasks to be performed could be defined within. Not much thinking was needed. We have little experience of discussing and managing subjective, internal states - although they are as much shared experiences as external objects visible to all. We have a general heuristic that says we should confine our observations to the externally visible, which kicks in to prevent the exploration of subjective phenomena even before they have had the chance to give results and justify themselves.

When things go wrong, we seek to clarify action, and capture better descriptions of more effective actions. In situations where flexibility is an asset, this leads to reduced aspirations. If things are proceeding according to the actions written on paper, they are deemed to be going well, and the opportunity cost is not considered.

Worse, the behaviour of people trapped in lack of understanding can reinforce each other. If one person just doesn't understand what is happening, they look about them and see others apparently knowing what they are doing, feel vulnerable, because lack of knowledge packets is supposed to be a personal failure, and therefore they bluster. They stick their noses in the air and waffle about `due consideration' and `appropriate action' as if `undue consideration' or `inappropriate action' was also on the table, but don't suggest what the appropriate action might be.

The thing is, everybody is doing it! So the silent conspiracy to maintain the etiquette of bluster develops. If anyone violates the etiquette, that person will be assailed by inherently unclear objections and other pressures to `conform', apparently for the sake of it. These cannot be countered in action-oriented terms, only by reference to causal relationships that only one person is fully cognizant of. Mapping in a packing world can be a painful and depressing experience, particularly if one

does not understand the shattered reality one's packing associates inhabit.

In pathological situations, this can lead to an infinite regress wherein every problem is addressed by attempting to delegate it to someone else, a procedure, or a blame allocation mechanism. It's rather like holding your toothbrush with chopsticks - if you are holding the chopsticks just like on the diagram, the brush up your nose and the paste all over the mirror are not your responsibility!

Remember, we've described the causes of this misery not by waffling about 'the human condition' or our colleagues' 'moral fibre', but practically, out of socially-conditioned avoidance of 'daydreaming'!

The Mapper/Packer Communication Barrier

It's worth reiterating some key points here:

- Mapping and packing are very different strategies
- Packing is the strongly enforced social norm
- The world is set up for packers
- Business language is packer language
- The results of mapping are called 'common sense'
- Common sense isn't so common
- Mappers think packers are cynical or lazy
- Packers think mappers are irrational
- Packers spend much of their time playing politics
- The last thing that counts in politics is reason
- Mappers are often wrong about packer psychology
- Packers are usually right about packer psychology
- Mappers are often wrong about mapper psychology
- Packers are always wrong about mapper psychology.
- Mappers do not have a culture to guide them
- Most mappers teach themselves, like Mowgli
- Mappers can teach themselves!
- Mappers can learn from others
- Mappers often face significant social challenges
- Mappers currently rarely fulfill their potential
- Once a situation is understood, it can be addressed.

This file last updated 20 October 1997

Copyright (c) Alan G Carter and Colston Sanger 1997

alan@melloworld.com

colston@shotters.dircon.co.uk