Improving processes, vs making new products

The challenge making digital technology for human needs industries

Karl Jeffery and Dimitris Lyras

January 2021

SOFTWARE FOR
DOMAIN EXPERTS

Published by Software for Domain Experts,
39-41 North Road, London, N7 9DP, UK
http://www.softwarefordomainexperts.com

Table of Contents

Improving processes, vs making new products 1

Introduction 4

For people to do better at their jobs in 2021 – requires better data integration 4

The situation awareness challenge 5

The process vs product mismatch 8

What this means for technology 9

What we mean by domain experts 12

Why we tell this story – what we see in maritime 15

The prize from getting it right 17

What this book can do - the Software for Domain Experts tool box 18

Breaking down the improving process vs new product mismatch 19

Breaking down domain expert work 19

Unpacking “situation awareness” technology 21

Unpacking “orienting” technology 23

Using football to understand mental modelling 25

Adaptability and rigidity 26

Unpacking the technology industry’s pursuit of novelty 26

Innovation in today’s culture 28

Domain experts don’t necessarily need “big” software 29

Why entity relationships make rigid software 30

“Requirements analysis” vs OODA analysis 31

The data integration challenge 33

The price of ‘digital transformation’ 36

Understanding the strengths of people vs computers 38

Understanding the opposing forces 40
Letting domain experts ‘design’ their own software 41
Introduction 41
This is not really "user experience" 43
Domain expertise and cybersecurity 44
Bringing AI into the domain expertise world 46
The supporters of the computer-does-everything world view 47
Platforms and the integration challenge 49
Introduction 49
Platform approach aids clarity 51
How to approach the integration challenge 52
Reducing platforms to their “schemas” 55
A world with no domain experts 57
Ending 60

Introduction

For people to do better at their jobs in 2021 – requires better data integration

We would like to see climate change avoided, cybersecurity problems avoided, cash more fairly distributed in society, and everybody having what they need to live in safety and security. To get there, we need the people in the critical roles of delivering this to be able to be better at their jobs.

Such as, the people who make decisions and plans around fossil fuel use and mitigation methods; who build IT systems, so they are secure and monitor what is going on; who provide employment in society; who run systems to deliver our food and energy; who operate policing and other government departments.

What does it mean to be better at your job in one of these areas? It means continuously looking for ways to improve the processes of our work.

And to do that, we need to have a better idea of what is really going on.

The problem is that the technology industry serving our organisations is not orientated around helping us improve processes and get a better idea of what is going on.

That’s because the most difficult problem with organisational software is integrating one system with another one, particularly if the systems are made by different companies, which is necessary if we want to see what is happening using data from two computer systems at once.

And the second most difficult problem with organisational software is making tools which allow us to improve our processes. Organisational software is generally designed to support the opposite – embedding our work processes within rigid software, built around relational database models which are almost impossible to change over decades.

This short book explores the problem, suggests some solutions, and also suggests some ways you could improve the digital technology in your own organisation, or the ones that you work with.

The situation awareness challenge

For people to do better at their jobs, the most important step is that they know what is going on, or going wrong.

They want to quickly spot emerging problems so they can be fixed. They want to predict a problem which may emerge, such as having a pandemic with no supplies of protective equipment, so mitigation methods can be put in place in time.

They want to make more efficient use of the resources they have, including people, equipment, time and money. They want to predict what might happen if things are left on their current course. They want to find better ways to do the work they need to do.

To do this means gathering and synthesising information from a multitude of sources, some digital, some not. None of us have any shortage of data or people telling us things, but the challenge is bringing together what is relevant in a way that we can see what is happening.

Digital technology as of 2021 barely does this at all. Digital technology does specific things it has been built to do, like deliver e-mail, process transactions, manage customer data, store and search files, show website pages. It can do sophisticated analytics and machine learning but only exactly what it has been programmed to do.

The reason digital technology has not been built to help us synthesise information from different places, to get a better understanding of what is going on and what we can improve, is that it is still really hard to build digital technology to do that.

The current business models we have for developing enterprise digital technology is largely big companies making big products with walls around them.

In our own minds, we can combine information from different places to get a picture of what is going on extremely skilfully. These are the same skills we used to win wars in ancient times, to play team sports, to build things, to bring up families, keep good relationships within our 100 person tribe, and everything else we have done.

We are not making a people vs machine argument here, but seek to show how digital technology can be better developed and implemented so that we, as decision makers, can easier synthesise information together from different places, build mental models of what is going on, or perhaps digital models, and make tools to better understand, make better decisions and better manage.

We want to show that the emphasis in society is moving away from this focus on continual improvement of processes, to a pursuit of "novelty" and "innovation". Innovation doesn't have to mean pursuit of novelty, but generally, it does. And do we have a society which has lost its ability to improve its processes gradually and continually?

In industries other than digital technology, we have ways of working which have evolved over hundreds of years where all the different components integrate together. So people don't necessarily have all the situation awareness, but they have the situation awareness they need. Such as our own industry, shipping, where we have evolved integration between shipowners, banks, cargo owners, crew agencies, fuel suppliers, port authorities, insurance, equipment suppliers and more.

Nobody planned out centrally the way shipping would work, it evolved over hundreds of years, from people solving problems along the way. The need to move goods over sea, the need to make it into a business and well-functioning market, the need to regulate that business, the need to supply that business.

The process vs product mismatch

The root of the problem, perhaps, is a cultural mismatch. The culture of industries and organisations which serve human needs, such as agriculture, shipping, policing, banking and government, is about continually improving and adapting processes, to do things better, to adapt to changes in the situation, or better maintain stability.

The culture of the digital technology industry is around developing and selling ever better new products. It is an industry driven by a quest for novelty.

This is also reflected in society at large. Fashionable discussion in 2021 is all about novelty – new products and trends, celebrities and movies, and novelty provided to us by the technology industry.

As a result, “human needs organisations” are not able to take advantage of the full potential which digital technology can offer them. But it also means there is a business opportunity, perhaps for you, covering the gap.

We are not proposing to try to change the culture of the technology industry. But to add an intermediate layer to connect these two worlds, which you can be part of.

We see the best way to cover this gap is to develop an approach to digital technology which is orientated around, and as far as possible driven by, domain experts.

Domain experts are the people who monitor and understand situations, make and implement decisions, in the human needs organisations – and know the most about what kind of software would help them the most.

What this means for technology

Human needs companies, and their domain experts, want digital tools which can show them what is happening in their domain, what is changing, how their decisions and plans are working out, what they need to be alerted about.

They want tools to better allocate and schedule their resources, people, tasks and customers. They want to be able to set up these tools the way that suits them, to best support their work. Once they have software they are happy with, the last thing they want is to have to learn new software.

The best way to deliver this, we envisage, is a mixture of large scale digital "platforms", and smaller scale applications running on these platforms giving domain experts specifically what they need, which are designed by people with domain expertise, and perhaps the domain expert 'users' themselves. These smaller tools can gather and integrate data from different platforms in a way which is useful to a domain expert, present it to them, and support their work, such as planning.

Big technology companies can continue their relentless pursuit of better platforms. This can include platforms for low code, analytics, AI, cybersecurity, transactions, visualisation, as well as customer databases, employee databases, product databases, cloud hosting, and other great stuff people can think of. There could also be platforms which integrate data from other platforms in a way which is useful to a domain expert.

To be as effective as possible, these platforms need to maximise their integratability and transparency, so people can see how they work.

Currently there is no shortage of such platforms, and continuous effort to improve them and develop new ones. But we are missing this layer of smaller scale applications running on top of the platforms.

There is still a lot of effort to develop large scale software applications to be used directly by domain experts. But the challenge here is that the need of domain experts everywhere is diverse, and the variation in needs shows increasingly as we get more deeply engaged with the software. Large scale software is expensive to build, so only works when the same product is provided to a large market.

For example, there are large scale software applications for managing purchasing and inventory, customer relationships and cybersecurity. But every industry manages its purchasing and inventory in a different way, has different types of customers and customer data, has different cybersecurity threats and different ways to manage people.

Many domain experts make schedules for resources of one type or another, but the specific needs are very varied, such as how much “buffer” should be left if one part of the schedule does not go to plan so it does not impact the rest, or what tools are needed to change the schedule. Think of a school which must find supply teachers when teachers are sick, or a police department which must make staff available immediately to answer emergency calls.

Big tech companies try to accommodate this with increasing amounts of functionality. This makes everything more complex. Less and less people understand it, including staff who work at the software companies and their salespeople whose role it is to explain it to their customers.

We might have a customer relationship management system with sophisticated tools to segment or analyse customers. But it is still not set up to tell a domain expert what they really want to know, such as how certain customers are changing what they look for, or what they think of your company.

The enterprise software market is not usually seen in these terms. People assume that big software companies chase the big industry 'verticals' and leave only the small verticals for smaller companies.

Perhaps the business model of big technology companies making big, centralised products to directly serve industries and their experts has reached its limit.

We are not saying that big companies are unable to serve the needs of varied domain experts. But to do so may require a different working model to the one they already have.

There are some big companies which do actually have an in-depth understanding of the varied needs of domain experts in multiple industries, for example business news organisations. But these companies operate in a very different way to big tech.

What we mean by domain experts

The focus of our story is the domain experts in human needs companies, the people who manage the services which give us what we need to live.

To define human needs companies, we can start with Maslow’s definition of basic human needs as food, water, warmth, rest, security and safety. Industries and organisations which directly help provide these include agriculture, engineering, transportation, energy, construction, retail, health, education, policing, manufacturing, government.

All of these industries have a number of sectors, and there are many supporting industries. Some examples of related sectors are roads, mining, emergency services, cybersecurity management, environmental management, financial services, management of our economies and social safety nets, sales, making trade agreements.

All of them have domain experts with a deep and continuously evolving understanding of how to best provide their service, trying to make small improvements or keep the organisation in steady operation as external factors change. And they know they are more likely to achieve their goals if they can do more with what they already have, not look for a new machine or tool which will change their lives.

Leaders in policing are normally wise enough not to put hope in some fantastic new technology like face recognition or predictive AI making their lives easier.

To understand how a domain expert works in a general way, we suggest the military "OODA" observe / orient / decide and act framework. Observing the situation, "orienting" in the situation by combining the observation with past experience and expertise, making decisions about what to do differently if anything, and implementing this decision.

There may be some specific goal to achieve, such as to improve efficiency of an organisation, reduce its CO2 emissions or improve its security. In this case, the OODA process is around making decisions to better achieve this goal.

Or, the domain expert may be responsible for a specific task such as scheduling resources, tasks and people, or diagnosing and fixing problems, and management responsibilities. In this case the "OODA" happens simultaneously with the task being done.

Domain experts draw on their experiences, what they are used to, what they have seen work before, what they trust and feel comfortable with. They build and maintain mental models of how things work, or the abstractions they draw from what they see. When it comes to learning from others, they are much more interested in learning from someone with a similar role, goal or working culture to theirs.

The popular term "agile" can be applied in different ways. Technology companies may see "agility" as the way to make new products quickly with experimental approaches and see people who oppose the implementation of new technology as an opponent to agility.

But domain experts might see "agility" as the ability to adapt their whole organisation to external changes, which happen all the time. For them to have to devote resources to understanding new software can be in opposition to that.

Technology people often criticise individuals in their target customer industries as being “resistant to change”. But this resistance may be due to the fact that a domain expert is being proposed software which is designed around a completely different model of the world to the one they currently have, and they do not see that it makes sense for them to change their model of the world to match the software.

Domain experts may also “resist” new software because they do not understand its internal logic model. Like the logic of the Boeing 737 MAX, which tipped the nose after receiving certain information from the sensor. To fly the plane safely you would need to know this logic and also understand that this sensor might be faulty.

Providing clarity of internal logic is not a requirement for non-domain expert digital products like computer games or websites, and something software developers may be unaccustomed to providing.

Sometimes software companies can inadvertently make life for domain experts harder. Such as when they use fear as a sales technique, “all your competitors are using this”. They may make domain experts feel forced to use their software, because someone else in the company has agreed to buy it. This leads to a reduction in the ability of a domain expert to understand their world, because they are now constrained by what the software is capable of doing, where the outside world has no such constraints.

Up to now, the digital world and human-needs-domain-experts world have been fairly separate. But the digital world is increasing in its reach, and increasingly clashing against people.

The people who work supplying human needs are invisible to much of society much of the time. Except, of course, when human needs are not being met.

Why we tell this story – what we see in maritime

The authors tell this story drawing on our experience working in digital technologies in the maritime shipping and oil/gas sectors, where we often see a mismatch between what ‘big tech’ thinks that industries want, and what industries actually want, and how domain experts can end up badly served by technology as a result.

The maritime sector may be extreme among human needs industries, being very diverse, and often subject to unplanned events. Its working environment continually changes. This means that the industry is particularly unsuited to many of the highly structured products which ‘big tech’ makes, such as for managing transactions, maintenance plans, and workflows.

But we suspect the mismatch between the big tech offering and industry needs is just highlighted more in shipping, but still applies everywhere. Every industry has diversity, unplanned events, and an external environment that changes. Even manufacturing, which can reduce diversity from its manufacturing processes, is subject to changes in market preferences, competition and supplies.

Here is a taster of some of the diverse challenges the maritime industry faces. From the industry's customers (the cargo owners), there are changing quantities of cargo which need transporting, changing demands from customers of what they want from ships and their management. There are changing market dynamics in the matching of ships and cargoes, which affect the prices customers pay.

The crewing department has to hire crew, manage evolving training requirements, handle constantly changing travel restrictions, and keep crew happy onboard, as crew interests change from one generation to the next.

The maintenance department has to deal with a wide range of machinery types. There is much unplanned maintenance work due to equipment failures. They may need to find external maintenance experts for a specific task. The equipment has alarms which need to be configured so they convey only useful information. There can be a range of unexpected issues during port calls. Ships need deliveries of food for crew and materials in ports around the world. Shipping companies are under pressure to improve environmental performance and exploring new ways to gather data which is useful in doing that.

There may be a need to retrieve past data, including understanding why certain decisions were made, how a problem was solved, why something was installed in a certain way, how to fix a certain piece of machinery. This may also involve looking through past e-mails as well as documents.

The prize from getting it right

The "prize" from making better digital technology for domain experts can be better programs which are more appropriate to people, combining people’s strengths with computer strengths. Tools more specific for people’s interests, needs and experiences.

This means that people feel more inclined to use computer systems provided for them, and use their own ingenuity to work out how to make it better, and help them in their jobs.

We can have data stored in ways which align with how people actually work with data, and the work processes it is associated with – so we can search corporate archives in better ways than looking for unique names.

We can have programs which are easier to improve over time, programs which are easier for people to understand. We can have less cybersecurity problems.

We can improve how we roll out software and other digital tools in our organisational environments. We can better support people to collaborate with the help of digital technology.

We can make technology which is easier to learn how to use. That’s equivalent to having a co-worker who is really easy to co-ordinate with.

What this book can do - the Software for Domain Experts tool box

Our goal in writing this book is to create a “toolbox” of concepts and methods which can be used to understand the problem, showing why the business opportunity exists and how to exploit it, developing better digital tools for domain experts.

Digital project managers often say they feel inundated with the complexity of the digital world, including the challenge of understanding what terms like agile and scrum mean and how they are relevant. We want to find ways to overcome this.

We have been developing this toolbox over 2016 to 2020, including publishing seven short books including this one, available on Amazon for Kindle download (search for “Software for Domain Experts”), and running workshop events in Athens, Greece.

We have also been applying these ideas to our work in the shipping and oil+gas domains, including producing software, running workshops, and publishing magazines about developments in digital technology.

Breaking down the improving process vs new product mismatch

Breaking down domain expert work

Companies and organisations have many different things to manage – their activities, their goals, their people, their equipment, their buildings and other assets. The domain expert, as we define it, is the person who is responsible for this.

“Managing” implies that there is a required output, which may be making steps towards a goal, or just avoiding or resolving problems. There will be various tasks involved, various risks and various resources.

The OODA (observe / orient / decide / act) framework, developed for military use, is a useful tool for breaking down what any domain expert does into components, which can then be a basis for working out what digital technology could do to best support them.

“Observe” – means to have a situation awareness as rich and deep as possible about what is happening. We can draw on a range of technologies which inform and alert us, and we may have people who directly inform our situation awareness. There are different time scales we might want to observe – urgent items which require us to do something right now, the general current picture, and slow trends.

“Orient” – means trying to understand what is really going on, what you think might be driving what you see. It means combining this situation awareness with models and past experience. These can be a mixture of mental models informed by the data, or computer modelling, including analytics and AI. The models themselves, both mental and digital, are informed by past experience. You may also want to see past information and data from the company’s records, which will require good data search / retrieval systems. You may want to talk to others about what you think is going on, or hear from others about what they did in a similar situation.

The “deciding” itself could be to change how something is running in the system being managed. It could be a decision to buy or sell something. It will nearly always be made by a person not a machine. If we do have a machine “deciding” it will be following specific logic made by a person, such as for autonomous cars or automated trading. There is a future promised where computers can make lots more decisions, but it is not here yet.

The “acting” – the implementation – can be done with a range of digital tools, such as updating a schedule or issuing an instruction.

If digital technology were fulfilling its full potential it would maximise people’s capability to do all of these things, all at the same time, including supporting collaboration with others.

Software tools will generally support just one of observe/orient/decide/act, but not all of them, and not in an integrated way. For example, a technology to support a certain maintenance task helps you “act”, technology to model a specific decision helps you “orient”.

Tools for collaborating ("orienting") and planning ("deciding") will benefit from having a connection to the tools for situation awareness, so they can tell us who is the best person we can collaborate with right now, and which aspect of our plan should be changed.

Unpacking "situation awareness" technology

When it comes to technology for situation awareness, there are many technologies available which gather and provide data. The technology challenge is compiling it in a way which is useful for people. That requires data integration. Normally domain expertise will be required to do the data integration, because the data does not make sense unless you understand the domain.

We have all seen dashboards and e-mail alerting systems, and probably reflected on how well their model matches the situation awareness we need, i.e. do they really tell us what we need to know.

To get a sense of what technology could potentially do with situation awareness and the obstacles involved, consider the decarbonisation challenge for industries.

A first step for decarbonising would be providing people with situation awareness about where they are with their various current CO2 emissions, what the organisation is getting in 'return' for them, and how hard it is to reduce them. This can then lead to "decide / act" stages of OODA, in this case tweaking the system until CO2 is only being emitted when the company gets something really useful or essential, and the emissions cannot be reduced further.

Perhaps there are ways to make big cuts in carbon emissions without changing our fundamental lifestyle and businesses at all, if we just knew where they were.

In the carbon emissions world, there is a lot of data and software tools available, but not so much situation awareness.

Or consider the industrial cybersecurity challenge. Cybersecurity is a subject which sits in the realm of digital technology, and this makes many people assume that the challenge of fixing it is best done with digital technology. But the biggest successes in combatting cybersecurity generally come from human experts, not machines.

Being hacked over a long period of time could be described as a deficit in situation awareness, in that people were not able to see that there was an intruder in their computer systems. Like having a thief living in your garden.

To have a digital tool that can tell us if something unusual is happening on our networks sounds like a fairly inexpensive and simple piece of technology. But as of 2021, to most companies, "cybersecurity technology" means anti-virus, firewalls or identity management, things which try to stop someone getting in. We have tools for spotting if something has changed on our network, which are branded as "AI" tools and come at great expense, or for big companies only.

Unpacking “orienting” technology

Orienting, according to the OODA concept, means combining what we see with our past models and experience, leading to the ability to make a decision about what to do.

Consider that doctors were criticised for being “resistant to change” because they did not embrace remote patient communications technology until forced to by COVID-19. Looking at it more politely, doctors were relying on their own experience, which told them they could rely on their abilities to give a good diagnosis when they had a patient physically in the room.

We do have entire industries geared around selling products which are based on people’s past experience, such as the fashion and hospitality industries. Our past experience tells us it was good to eat in a room which had a certain atmosphere, or wear clothing which had a certain feel to it.

Advertising builds links between the product it is selling and people’s positive past experiences, such as Christmas. Or times that they felt in control, such as driving on beautiful open roads. Retailers know that people’s most negative experiences are associated with not feeling in control, and so endeavour to make their customers feel in control at all times. People feel that they choose to walk into the shop and choose where to go when in the shop, and there is no mess, which acts against people’s feeling of control.

Technically oriented people may think people are being superficial and weak by following advertising or fashion. But what is actually happening is people are drawing on their biggest strengths, using their past experience to help them “orient” their decisions today.

In industrial situations, people believe they act logically, but we can see domain experts making decisions using different models when they are in different cultures. Such as the differences in how shipping industry people behave between Norway, Greece and Singapore.

Our mental models also drive our willingness to take risks. If we haven't done something before, we may overstate how big the risks are. But at the same time we are comfortable with comparatively enormous, but very familiar risks, such as from driving.

A digital tool to support a domain expert's "orienting" would be helpful if it could connect an expert with another expert who has been in a similar situation, either through direct communications, or by retrieving a video of where this person talks about their experience. We prefer to make decisions based on our own experience, but second best is someone's expertise who has been in a similar situation to the one we are in now.

When industries and organisations say they are only as good as the expertise of the people in them, they might also be saying, they are reliant on the mental models of their experts.

Perhaps the biggest reason that industries structure themselves the way they do is to enable people to be focussed on the sector of the organisation's operation they can develop an integrated mental model for. The mental models for marketing and financial management are very different, so we do not usually expect marketing and financial management people to work closely together.

Using football to understand mental modelling

To better understand mental modelling, and people’s enormous capacity for it, consider the complex models that we all make when playing football or even understanding what is going on.

The basic model of football, which every observer and player understands, is to get in a location where you have a low risk shot at the goal. The number of opportunities you get to score increases the further you are from the goal, but the lower the likelihood of scoring.

Then there are multiple other models which players and observers understand, such as how playing a more offensive game means increases in risk, and how behaviour will change as players get fatigued. Players need an in depth model about their own abilities, so they know what to try and what not to. They will have sophisticated models for how to behave in dealings with other players and the referee.

Sport modelling differs from organisational modelling in that it all happens intuitively, so there is no need to describe it in the way we did above. The time scales are shorter than for most domain expert work, and there is immediate knowledge about whether something was successful. But other than this, the basics of modelling for someone to watch or play football are the same as any domain expert would use.

Adaptability and rigidity

Our minds are able to change and adapt. Some of us can adapt faster than others, but all of us are capable of dropping our existing model and developing a completely new one.

We can easily see how something can be adapted to something else. We can see that an axe can be used as a hammer for a small job, or see that planning a large party is similar to planning a dinner party. In the COVID period, we all worked out how far we could continue doing what we want to do, while working with the new requirements to stop the disease spreading.

The physical world does have rigidity – because things are built which cannot be easily changed.

The software world does have a lot of rigidity. It takes money and time to build software. But it is technically possible for software to be adaptable, as well. And it would help a lot in making tools for domain experts if we can reduce the rigidity. Because, as we started off by saying, in order to do better work, we need to be able to improve our processes and integrate data together in new ways.

Unpacking the technology industry's pursuit of novelty

The technology industry makes money, and gains investment, when it promises hot new products. The continuous improvements in hardware and communication speeds make this possible.

We expect to be ‘wowed’ by some novel development at technology events.
Novelty can be a good sales technique. We are attracted to the idea of a quick fix or improvement.

Having a focus on chasing novelty is a human characteristic we see in other areas of life, and it is not usually a positive one. Think of someone who is always going from one life relationship to the next, dropping a partner as soon as the relationship loses its novelty, rather than letting the relationship mature slowly and going through the ups and downs with it. Or think of someone who is always moving jobs, or houses, or having different friends.

Technologists will counter that making new products offering a big improvement is the best way to make money, whatever industry you are in.

Looking at data about the “fastest growing companies” backs this up. In 2020, we see lots of companies in renewable energy, e-commerce, payments, social media, pharmaceuticals, and multiple Chinese sectors, as the fastest growing. They are all offering something which was not there before, not incremental improvement.

The purpose of this book is exploring ways to help human needs organisations do more with digital technology, not showing technology companies where to make the quickest money. And with the “new products” game being very winner-takes-all, many technology companies may be interested instead in better serving human needs organisations.

We can also consider the size of the markets, and amount of money spent, in human needs companies and organisations. That includes agriculture, construction, energy, manufacturing, mining, “service industries” where people give advice / personal attention / personalised work, retail, hotels and restaurants, real estate, transport, household goods. And government sectors like administration, health, education, defence and policing. Surely a majority of the population in any country.

Innovation in today’s culture

Today’s culture emphasises innovation. It seems that this has been an entrenched part of our society forever. But it isn’t. Innovation is not a word we associate with 1980s business, for example.

Innovation is not necessarily about pursuing novelty – it can also be about improving processes. But when people use the word ‘innovation’ they are generally talking about novelty.

Politicians and business leaders emphasise ‘innovation’ perhaps because they are concerned about western economies maintaining their competitiveness and see this as the best way to achieve it.

Software companies emphasise innovation as a way to sell their product – an innovative company is one who uses cloud hosting, for example.

But perhaps our businesses would be more successful, and society healthier, with more focus on gradual improvement and solving difficult problems with our minds.

Domain experts don’t necessarily need “big” software

A trend in the organisational software market over past years has been more and more “big” corporate wide software systems, such as “CRM”, “ERP” and so on.

Selling corporate-wide software is very attractive to a software company, but not necessarily what a domain expert organisation needs. “Big” comes at a price of complexity, including complex challenges integrating the software and managing cybersecurity.

Smaller software is likely to be easier to manage, supporting the needs of experts doing a specific task.

Just like there can be an optimum size to companies, and there was an optimum size to human tribes, there can be an optimum size to software.
Software should be as big and integrated as the domain demands.

There are some ‘use cases’ where it makes sense to have large integrated software, such as a grocery chain which wants to consolidate its purchasing, or a national health system which wants a single database for everyone in the country. But most domain expert situations are not like this – such as someone managing water sewerage in a single town. They do not need to use the same maintenance management system as anyone else.

Bestselling business books (see “Humanocracy” by Gary Hamel) have been written about the idea that breaking big companies into smaller units can be more pleasant for workers, leading to higher levels of economic productivity.

Digital technology itself has enabled a lot of this organisational centralisation. Before digitalisation, companies had no choice but to let their various operating units operate with some autonomy, whether it was another branch or factory, or a ship in the fleet, with people handling their own budgets with cash. Big software products enable this to be done centrally.

Some tech enthusiasts believe technology is all about empowerment and letting people do what they want. That's not true. Technology can also be a tool to control people, and some people like it that way.

Why entity relationships make rigid software

A root cause of the rigidity in organisational software is that many products are built around a database, and the "entity relationships" associated with it.

These connect with the domain expert's process, as they are understood by the developers, at the time of first building the software.

An "entity relationship" may be a model, but it is a very basic one. It is built on data relationships, which is different to how different pieces of data relate in an expert's process. For example, a data relationship is that if I buy two pints of milk and put in my fridge, my funds go down and milk in the fridge goes up. An expert's process is remembering one of your children has started preferring one brand of milk to another, although they both taste the same to you.

So software companies, with software made in the conventional way, find it very inconvenient to help a domain expert improve their processes, in case it means they prefer to use a different model to the one that the software is built around. The software is designed in such a way that improving the processes is very hard.

The software is serving a real world process, and real world processes always evolve.

It is possible to build software around models rather than entity relationships. There are services from ‘big tech’ which make this easier, such as low code platforms, NoSQL databases, and a range of models being offered on services like Microsoft Azure.

“Requirements analysis” vs OODA analysis

The reliance on entity relationships as a basis for software leads to the standard process for planning out a software construction, which is called "requirements analysis". This is a process for defining in detail what the software needs to do and how the data relationships in it will work. An aim is to limit the scope of the software upfront, to constrain the development costs and complexity of what is being built.

But it is only possible to start this process when you already know what is relevant and important to the domain expert.

Breaking down the domain expert's work using a process like OODA, on the other hand, will identify what is most relevant, before even thinking about software. What is the domain expert observing, what do they need to orientate what they see, what decision do they make and how do they implement it?

The military may teach someone how to observe, orient, decide and act in a range of situations. The equivalent of a "requirements engineering" process, on the other hand, would be working out what specific specifications of weapon or clothing they need and how that would be supplied.

Or to take a housing analogy, when you plan to build a home you are going to live in, the "observe" means looking at places you might want to live and opportunities to buy land, "orient" means assessing your capability of raising money and paying the loan, and thinking about your future needs, “acting” means signing for the transaction. The "requirements analysis" is asking how many windows and how many bricks you need.

Once you are managing your life in the house, an OODA analysis shows you are “observing” things like whether your children need encouragement going to bed at a reasonable time, if they are eating properly, if the bathroom floor is clean, if the house is secure when you go out. This leads to orientating, deciding and acting to fix these things. A 'requirements analysis' on the other hand would be asking what type of burglar alarm you want and what kind of sound it should make.

The danger of getting into “requirements engineering” too early is that the discussion of the details can easily dominate the discussion – like people planning a wedding getting obsessed with the details of the ceremony.

And once the details are fixed, you have lost the opportunity to change the big picture – if you are already discussing paint, you won’t want to say that you have changed your mind about which house to buy.

It is the same with software – a discussion about data elements, like product numbers and prices, is a step removed from the real world challenge such as keeping our warehouse stocked with the right products. You have already lost the opportunity to discuss whether it is most relevant for the domain expert to know about product numbers and prices, and if so which prices and product numbers.

Another analogy to explain this problem would be starting a conversation with a plan to define what words you are going to use.

Military trainers may advise people not to get too tied to a methodology, because that might give a way for opponents to take advantage, by coming up with an attack strategy where this method does not work.

An OODA analysis is not tied to any particular method. Any domain expert process working towards a goal has elements of observe, orient, decide and act.

The data integration challenge

The mismatch between the “product” orientation of technology companies and “process” orientation of human needs industries shows up most painfully when it comes to integrating digital systems together.

As we discussed at the beginning of this book, the pathway for domain experts to do better work often comes from integrating data together in new ways so it makes sense. That means that the domain expert needs to also define how the data is integrated together.

1:1 data integrations are very time inefficient and hard to adapt – so what we need are ways to bring the data together into a model, just like it happens in our heads. To make and update this model, you need to understand how the domain works. You might also want the model which follows the model a domain expert has already built in their heads.

To understand why data integration is hard, consider the various 'data sources' we might need to do a better job of managing our domestic life.

Factors might include our family's food, clothing, children (mental and physical health, education, entertainment), the house itself, the garden, our finances. We have a mixture of routines – things we do every day, week, month and year. Then things we do when something needs doing, based on some alerting trigger. Sometimes we spot something going awry slowly. We have longer term projects.

You could imagine the logic such a computer would follow, because you already have a logical model in your head – what you do on a Saturday morning, how you load the dishwasher. When you do laundry, restock kitchen supplies, mow the lawn. What you do when a certain event occurs, like the neighbour complaining your garden tree blocks their satellite dish.

This logical model works for you, but it is also personal to you. Differences in people's logical models is the source of many domestic arguments.

If you were to build a digital tool to integrate the data and advise you what to do, it would be too cumbersome to build it from a set of logical rules, like mow lawn when grass height = x. You would build it as a model the same way as you do it in real life. Some things you do as a routine, some things you do when something attracts your attention, such as running low on milk, some things you do when prompted by others.

Consider the information sources for running any small business. It would probably have sales and marketing, purchasing, real estate, staff, finances, inventory / supplies, needs for different skills, some of which it has in-house, some which have to be brought in. There will be a lot of communications, internal and external. To run the business better, you somehow need to synthesise all of this information into your mind, or on a digital system, so you can see what is going on.

Since understanding someone else's model is the same as understanding why they manage their domestic life the way they do, it is preferable if the person who will use the model is as involved as possible in creating it.

Also – if the digital technology is following a model, rather than a set of rules, it is much easier to update. Just as we can easily update the models in our minds when our children tell us they know longer like chips and prefer pizza, so we have to change our shopping list. Or understand what we need to change when something major happens like schools closed due to a pandemic.

If we can’t see the logic of the model, then it is hard to improve its algorithm. We end up with a situation of computers making useless recommendations, a familiar situation to many of us. Like the supermarket self-checkout machine which beeps every time we use it for the same reason.

And what does big tech offer for our domestic world? Heating we can control from our mobile phones, buttons we can press to get Amazon to bring us more toilet roll, a voice activated way to play us more music. Controls to stop our children spending too much time on the computer which are difficult to implement. Software which analyses our bank accounts but has no idea how essential our various spending was. It does not get very close to helping us manage domestic life.

The price of ‘digital transformation’

“Digital transformation” is a fashionable term, implying that the organisation goes further than adopting technologies, it actually changes.

When you factor in the importance of experts’ mental modelling to organisations, you can see that digital transformation may come at a cost. If you transform too much, these models are no longer valid, because the organisation has become something different.

A common argument for digital transformation is to say that your industry will be transformed the way Uber, Airbnb and Amazon transformed their own industries, with a completely new business model.

The counter argument is that these companies used technologies to exploit inefficiencies in the previous system which are obvious now – empty rooms in residences in areas desirable to tourists, competent car drivers willing to transport paying passengers for less than taxi drivers, and the real estate advantages of e-commerce.

Most industries don't change that much through use of information technology. There are improvements which can be achieved through steady improvement by domain experts, getting more crop from a field, deploying a ship maintenance team more efficiently, finding ways to improve policing.

And Uber, Airbnb and Amazon did not transform themselves, they are new companies. For an organisation to transform itself is extremely difficult, because people have good reason to hold onto their existing models. Perhaps it is actually better to start up a new company rather than Hilton Hotels trying to become Airbnb.

In 2020 the most profitable tech companies are enabling someone else to do something – look at Microsoft, Google, Facebook and Netflix.

Microsoft has moved its focus away from selling software licenses, to its platform-type, cloud-based tools to support businesses, which specialist companies can build on top of.

For Netflix, the 'someone else' is people who can make terrific television. The company's strategy is to find the best makers and then get out of their way.

If your human needs organisation has been in business or operations for some time, we can guess it has competence to do something well, its experts have good models, and it cannot be simply replaced unless there is some underlying reason for it. Such as, that you are an expensive taxi firm in a place with lots of competent drivers with time on their hands.

Understanding the strengths of people vs computers

The question of man vs machine often come up in discussions about domain expert software. As though software developers think that building software around the way people think is pointless, since people's intellectual dominance is short-lived. Here's our perspective.

Computers and people think differently. That is un-arguable fact. Unlike computers, people are able to learn about a domain, make models, and continually improve them, taking into account all kinds of new information, including information they have never seen before. People process their experience and make abstractions from it, understanding which factors are driving other factors.

Computers, by comparison, can handle much larger amounts of data, follow precise instruction, and process numbers much faster than a person. This means they are good at searching large amounts of data, and precisely following defined steps (such as in issuing an airline ticket). When a computer is processing millions of transactions a second, it can appear to do all kinds of things. And some people argue that it enables a computer to approach thinking like a person.

But thinking like a person includes self-orientation in the world, which computers are not yet able to do. If they could, a computer program would know its own limitations, like a human knows the limitations of a computer program.

Think of the difference between buying a train ticket – something which a computer handles easily – and planning a holiday, where we want to visit multiple places, keep our costs and risks down, and perhaps maintain some flexibility within constraints, such as freedom to make different trips but to start and end on a fixed date. A computer would need a lot of custom programming to do this and would still only be able to do this in the way it had been programmed.

A computer cannot understand a scene directly as a person can – it can only run image recognitions and algorithms and follow instructions.

There are many well publicised man vs machine competitions, including in chess and shooting F16 planes, but this is answering a different question – whether computer capability at this task, after a lot of custom programming, is better than a person's capability at this task.

A better question would be whether doing a task relies on the sort of continually improving mental models, based on a wide range of data, which only a person can make. Most challenges in the maritime industry certainly do.

In the past, people and computers have largely lived in separate worlds, and so the difference in how they think hasn't mattered, like with people and dogs. But as people and computers come ever closer together the differences in thinking become more of an issue.

Understanding the opposing forces

All of this goes to show that there are forces in organisations and the digital technology industry which are in opposition to the idea of giving better tools for experts.

Instead, they are looking for new technology markets, which they see as areas where technology can do something fancy by itself, (not support someone else to do something). This is like trying to make a shirt which looks great in the shop and sells well, even if no-one finds they like wearing it once they have bought it.

These people do not see much logic in the idea that people will maintain intellectual superiority over computers for much longer (and don't agree that they have it now). So they see a future which has more and more AI, whether or not anyone can understand or control it. And they see a world they don't particularly wish to live in.

These people have a lot of credibility in today's society.

The best answer is that the real world is pretty complex. We are getting more, not less, problems with the challenge of getting all of us the food, shelter, health, safety and security we need, and there is a lot that needs fixing.

Automation can only work within the realm it is programmed to work in. This is the reason autonomous cars did not meet expectations. It proves impossible to program a car how to handle all of the situations it may meet. We need human judgement after all.

Letting domain experts ‘design’ their own software

Introduction

When domain experts observe, orient, decide and act, they like to do so on their own terms. They like to decide what to look at, what to use to ‘orient’ it, they like to make the decision and ensure it is implemented.

The reasons are obvious – they are responsible for some outcome, which might be stable operations, or it might be a specific goal. Their task is to work out how to achieve that outcome. They develop their own models for how to do that, and want to use them. And explaining your own model to someone else is very hard, let alone persuading them to adopt it when they believe their own one is better. But everybody has a different model, because everybody has a different past experience.

Understanding this could be as simple as seeing it is the same concept as a grocery store which lets people choose when they visit and what they are going to buy. Some people prefer to drive in their own car rather than ride on a bus, and some children prefer to cook their own dinners and manage their own finances.

Consider how traders develop their own models for understanding what is going on and when to make a trade in their domain. The strength of these models is what makes the strength of the trader. They may be so complex they cannot explain them to other people or may not want to. But they can build their own spreadsheets or other tools to show the data according to their models.

If a domain expert were just following a rigid structured process in their work, they would not be a domain expert, they would be someone waiting to be replaced by a robot. But not many of the roles in managing human needs services are rigidly structured.

So for digital technology to support domain expert work, domain experts need to understand the logic behind the software, so they can see how it relates to their mental model of how the domain works. It would help a great deal if the person making the software is also an expert in the same domain. It may also help if the domain expert "user" is able to influence the design or logic of the technology they use.

Defining and configuring software requires digital proficiency, which not all domain experts have. But many domain experts are digitally proficient. Also software platforms are developed which reduce the amount of technical proficiency needed to define software, such as "low code", providing some prefabricated software building blocks.

Domain experts everywhere can usually define what information they are looking for, since this is a core part of any human needs domain expertise. And domain experts can work together with people who are more digitally focussed.

Supporting this user-configurability or user-design could be understood using a platform analogy, splitting the software product into different levels.

In your house, it is easy to change ‘top layer’ elements, like the location of your furniture and how you organise your kitchen. There’s another layer down where you can make changes with some effort, such as repainting the walls. And another layer down you can make changes but with other experts and costs involved, such as redesigning the interior room layout.

Similarly in the digital world, there are some elements which are easier to be “user-configurable” than others. But what we would like to see is potential for user configurability to go much deeper, including to the level of what information is shown and the logic in how the data is handled or analysed.

This calls for powerful but flexible and easily integratable platforms beneath the “user definable” layer. Such as low code, data integration platforms, AI platforms, cloud hosted models and databases.

This is not really "user experience"

“User experience (UX)”, according to Wikipedia, is about “a person's emotions and attitudes about using a particular product, system or service”.

The way software reflects a domain expert’s mental models will drive their attitudes and emotions about using a product.

But we don’t see many discussions at UX events anything like the ones in this book. Most UX discussions are a short step away from user interface design, which takes into account where people expect or like to see buttons.

The gap between domain expert perspective and the UX perspective is something you may experience if you fill in a website survey made by a UX person. The survey is asking you how you felt using the website and whether you felt able to trust it. Actually, you are raging mad because the parcel you are expecting today will arrive after the office closes and the courier company won’t let you contact them, since you are not their direct customer. This was due to how the courier company designs its processes for managing customers, nothing to do with the website UX consultant.

Domain expertise and cybersecurity

We can call the people who manage digital systems, including cybersecurity, domain experts. And just like with other domain experts, they have elements to observe, they orient, they make decisions and act.

The ideas in this book could also help these people to improve cybersecurity.

A first step is to recognise that just because cybersecurity is a digitally related problem, it does not mean that the solution is also digitally related, such as with cybersecurity software. The best way to improve cybersecurity is probably still with human domain expertise and their mental modelling capability. But supported by digital tools.

A cybersecurity domain expert could begin by insisting, as far as possible, that the digital systems they look after are transparent – easy to understand, and easy to understand how their security systems work.

Running through the observe / orient / decide / act framework:

Their tools for "observe" can include tools which tell them if there is something unusual happening on the network. What other sort of hacking problems similar companies are experiencing. They can maintain an understanding of how well set-up their systems are to detect hacking, and how easily they would be able to quickly remove a hacker and get systems running again afterwards. And whether employees in the company are creating security risks.

Their "orienting" involves working out if they actually do have a hacking problem.

The easier this is to do, the easier it will be to take action against a hacker, for the "decide" and "act" components.

This same approach can also be used for the challenge of regulating cybersecurity. Instead of providing prescriptive, complex lists of measures companies must adopt, the regulatory agencies can demand that all of the above is being done, and send inspectors to determine whether or not they agree.

This same "risk based" approach is also used for regulating the management of risk in the oil/gas and shipping industries.

Bringing AI into the domain expertise world

If we are using AI as part of digital tools, it would probably come into the ‘orient’ part of the OODA loop, how we understand what is going on beyond what the data is directly telling us.

All of the variants of AI, such as probability modelling, deep machine learning, statistical analysis techniques, can be useful in helping a domain expert “orientate” in a situation and make better decisions. They can show the big trends, what is driving what, and what would be the best choice to make.

AI can work with more data than people can, and be programmed or manipulated to build models which show us useful insights, such as how much wearing masks on underground trains stop us catching coronaviruses.

There are some domain experts, in fields like oil/gas and pharmaceutical research, who have capability to design their own AI tools. But it is more likely that an AI expert designs the tools, working together with a domain expert.

Discussing AI for domain experts is a challenge in itself. AI specialists are interested in making better AI, so may have limited interest in how the domain expert sees the world.

For domain experts, the potential of AI may be misunderstood or overhyped. It is too easy to shut a conversation down by mentioning AI, which only the AI specialist understands. And the domain expert may (probably wrongly) believe that the AI threatens their employment security.

The limits of AI are not very well understood, and it is hard to discuss them without appearing like an AI opponent. And the limits are changing all the time. But they nonetheless exist.

The hype itself causes a problem, with cycles of unmet expectations and cynicism, while people fail to notice the slow but steady progress actually being made.

A domain expert would need to understand the logic of an AI system to accept it, or benefit from it. Just as we get a rough understanding of how Google Maps works, when it calculates the best route for us to drive taking different factors into account.

Some technology companies, such as C3.ai, are developing AI platforms with the goals of making AI systems easier for a domain expert to understand and implement. This is surely a worthy goal.

The supporters of the computer-does-everything world view

Another way to understand why it is difficult to implement digital technology orientated around the needs of the domain expert, is to look at the people with the opposing belief. That digital systems should be orientated around what the computer is capable of doing.

In this world view, AI systems continually improve, and people become more and more redundant. But because there is no overall control over the AI systems, the world eventually becomes uninhabitable. Everything eventually becomes controlled by what the computers are programmed to do.

Where there is a conflict, the most survival-orientated logic wins, so decision making is programmed to prioritise survival of the digital system and the company behind it.

Let's look at what that means for our three challenges – climate, cybersecurity and cash distribution.

So for climate, we can expect to see decisions driven purely by short term economics, and since there's no price on carbon emissions, or likely to be one big enough to change behaviour, that means a lot more CO2 going into the atmosphere, and no penalty for wastage. We can try to beat that by focussing on novelty, such as with fancy electric cars, solar panels and batteries. But will new products alone fix the problem?

For cybersecurity, we can impose unlimited cost and inconvenience on our 'users' because that comes at no expense. So requirements for complex, unique, passwords, which must never be written down. We blame our 'users' if they inadvertently reveal their password, even when due to a sophisticated phishing. And we won't make much effort to make digital systems transparent, because that only benefits people, not machines. Machines can cope with messy software so long as it works.

For cash distribution, the digital systems operated by the banks will take care to ensure they never lend money to people where there is a risk of not getting it back. The digital systems operated by social services will be as complex as possible, since complexity comes at no cost to the machine.

If you don't like the sound of this world, let's think harder about how to make digital products which can let domain experts make the problems of climate, cybersecurity and cash distribution go away.

Platforms and the integration challenge

Introduction

The digital technology market for organisations, and the products companies make, is conventionally carved out vertically, in that software products are created for specific ‘verticals’ and presumed to give them everything they want, such as a ‘full stack’ set of software tools for oil and gas exploration and production.

There are some ‘horizontal’ products presumed to provide for all the needs of a department which is present in multiple industries. For example, manufacturing and purchasing departments in all industries can use the same ERP software.

What would probably work better for domain experts, and something we are already seeing in many areas, is a division where big companies make horizontal products for providing fundamental tasks for digital technology. Then applications are built on top of these giving domain experts what they specifically want, gathering together data in the most useful way, which can be customised sometimes by individual users.

This structure reflects what we see in the real world. For example, looking again in football, we have some big horizontal ‘platforms’, companies building and managing stadiums, the football leagues, the TV programs, the ticketing systems.

The domain experts, the players and their coaches in this case, have individual systems tailored to them. They don't need a deep interaction with any of these platforms.

Today, we see platforms for domain experts in today's housing industry, including the banks lending money, governments managing of land and approving projects, big development firms, big firms supplying materials and equipment, large scale construction, big databases of properties for sale. These serve domain experts such as property developers, architects, lawyers, specialist builders, interior designers and estate agents.

The evolution of our businesses into platforms and domain experts has happened slowly over hundreds of years. In the past, the world did not divide into platforms. Many people built the house that they live in. Transportation meant a coach and horses where the same person built the coach, fed the horses, drove the coach and fended off highwaymen.

So we are now seeing the same trend happening in the digital technology world. In the past, a single software company might have provided for all of a customer's digital needs. In the future, it is more likely to be put together with a number of platforms and specialist tools.

A small business today might use multiple digital platforms, such as for banking, website hosting, sending e-mail, making documents, collaborating, payroll, as well as other platforms related to the domain. It gathers data together from all of these places to help the domain experts to observe, orient, decide and act, both in the work the company does for its customers, and the management of the business.

This trend has much further to run. With the help of digital technology, we can imagine an oil and gas company turning itself into a ‘platform’ providing energy and CO2 mitigation services to its customers, with the desired mix of cost, convenience and CO2 emission, for the customer’s needs.

We can imagine a shipping company becoming a platform to its customers, such as an oil company or supermarket, providing the service of moving goods around the world when they are needed.

In all of these examples, the platform is changing relatively slowly, and does not need much flexibility, but the domain expertise world which runs on top of it is changing quickly. The domain expertise world is also continually evolving its processes, and the situation awareness it needs to do this. The challenge is connecting these together.

Platform approach aids clarity

A benefit of the platform-based approach is that people can see what is going on much more easily.

Just as in housebuilding, the architect, developer, lawyer, specialist construction company and estate agent can see what is happening within their own domain easily, because they are working with tools designed for them.

There is no lack of transparency anywhere, except where commercial confidentiality specifically calls for it.

This clarity helps the “observe” part of OODA. It also helps the development of mental and digital models which help us “orient” and improve our decision making. We can see faster if we are about to make a decision which would not work, or if it is not being implemented very well.

This clarity also democratises the digital technology – it is far easier for other people to get involved, start to build their own understanding which gets richer over time.

How to approach the integration challenge

The hard part of this is that domain experts are generally seeking to get data from a number of different platforms, not just one. They may want to use tools designed by themselves, or a small software house with domain expertise, rather than tools made by a big software company. So we need data integration.

This book is not going to get into technical aspects of data integration, but before getting into that, it is important that everybody recognises how important data integration is, and that it is useful for a domain expert to be able to define how it is done.

Any data system can make itself easy to integrate with other systems so long as the people behind it want to. Just as every person can make themselves approachable to others, so long as they want to.

In the world outside of software, business customers try to make their requirements and instructions clear and straightforward; suppliers try to make customers feel they are getting what they need without being overburdened. Suppliers want to present themselves with a straightforward offering free of mess. These are approaches to business integration, but not often applied to digital integration.

The goal is to provide data to a domain expert in a way which is useful to them. Every domain expert puts data together in a different way. A starting point is to look at how the domain expert is already putting information together, but maybe struggling to gather it from different sources.

Consider architecture. An architect is, in a way, an information integrator and modeller, working together with the ideas of a building company or owner, and understanding the constraints of house construction, regulation and cost, to come up with a workable plan which can be passed onto a construction company to make a desirable building. An architect will have established "OODA" processes – what they observe, how they orient, how they decide and act.

So how can we build digital systems which put data together in a way which a domain expert wants, which can also work as a commercial operation?

One pathway could be to have an intermediary company which focusses on handling and integrating data in a way which makes sense to a domain expert. A company developing this business model with some success on a large scale is Cognite of Norway.

We can have domain experts directly advising how they want data to be brought together to give them the situation awareness they need, perhaps working together with a digital expert. We could have a small business employing both domain experts and digital experts.

Domain experts do not just have one integration model. Any domain expert could have thousands of different integration models they use, for how they keep situation awareness about a certain area of their work, or what to do when a certain event occurs. Just as footballers are continually integrating information about the ball, players in their own team, the opposition players, and themselves, with thousands of different models in their heads.

A domain expert will also be considering what is driving what. A bank statement is not just a list of transactions, it reflects the company's sales and purchases. If the bank balance is not satisfactory, then there probably won't be any information in the bank's own system which can show how it can be improved.

A common problem in 2021 is that operators of big software systems just make it too difficult for smaller software tools, with domain expertise embedded, to integrate with them. The large software company may say they have APIs and adaptability, but it works out very expensive in practise.

Platforms which are commonly used together can develop better integrations between them over time, including understanding how their neighbouring platforms work, just as we get to understand our own neighbours.

Data standards and APIs can be very useful, and should of course be used when available. But if they are not available, it may be very uphill work pushing for them, particularly if the reasons are commercial rather than technical. Such as companies do not see they have a benefit in making their products easier to integrate with. Or there are competing ideas in an industry about how a process should be followed, which obstructs people agreeing on a standard.

Standards are easier to adopt when multiple companies already use data together in the same way, or follow the same process.

Reducing platforms to their “schemas”

Platforms are easier to integrate together when they can be simply defined. In the way that footballers don’t need to know the details of the football stadium they are going to play in, they just need to know it is a football stadium.

There are many examples in life of platforms defined as schemas. We can know nearly everything we might want to know about a new laptop just from knowing the processor speed and the amount of memory it has, and we can easily fit a new hard drive.

The music notation system is a schema for how musicians should follow music written for them, so their parts fit together, and they play what the composer intended.

A country’s national constitution is a schema for how the government should work, so that we don’t need to know all the intricate details of everybody’s working life to know what they should be doing.

The human-needs-domain-expert world is low on schemas for digital technology as of 2021. But it can be very high on schemas for its world outside digital technology, such as with key performance indicators widely used in an industry to measure performance.

Someone who works in a company as a digitalisation manager should perhaps eventually work entirely in the world of schemas, and not need to get into the depths of technical products in any part at all.

A world with no domain experts

As we come to the end of this book, it is time to try to imagine a world where all of this fails. Domain expertise gets progressively less important in our world, as increasing faith is made in computers to make decisions.

In this world, computers do not have a balanced view. They are likely to make decisions to follow goals they are programmed to follow, that are unlikely to be adapted to suit the needs of the wider world.

People all still make models and have judgements, but whether these judgements count or not depends on your wealth. We have industries dedicated to serving the opinions of wealthy people.

The biggest companies are the ones which create the algorithms - the big tech companies - because although they may not be in charge for now, everybody imagines they will be eventually, since computer decision making is all that matters. So all the investment goes in that direction.

There is no pressure on these companies to build technology which people can understand, particularly their 'users'. The opposite in fact, they see competitive advantage in making things hard to understand. Even making their core business models hard to understand, such as when they make money from creating useful products but pretend to us that their competitive advantage is from advances in computer decision making. Or are they playing more to shareholders than customers, and shareholders believe that computer decision making is the future? Hard to tell.

If you are weak in society - whether mentally, financially, health-wise, there is no logic from anyone else from standing up for you, you have no money, no influence, no capability to improve the world for bigger companies.

If you have influence in society you may want to use your influence working for causes you believe in. But you can also see that the future is increasing amounts of AI driven decision making, and you don't particularly like that. Perhaps living on Mars sounds appealing.

Cybersecurity problems get out of control. Nobody, outside the people who create the code, even understand the code, and perhaps not even them. It is fine if everything works. But when hackers do find a way in, perhaps because people are using software products which are not up to date, or because the effort of a state puts enormous resources into finding a pathway, it is unstoppable.

If you are a person with low influence in society and faced with software where you don't understand the logic, like a difficult to complete application form, don't expect anybody to listen to your complaints. The programmers are in charge or feel that they are. Everybody else has to work around them. You will just need to figure out how to use it.

Accidents may happen through holes in the systems. But to fix them, the people who build the systems would need to acknowledge their failures, rather than blame the people who 'use' them, which is not something likely to happen, when the people who build the systems are all powerful.

There is no point in being a human expert and arguing for the strength of human expertise in this world - everybody knows the days of human decision making are numbered.

When it comes to climate change, or fair distribution of cash in society, forget it. With nobody in charge fighting for the general 'global good', everybody's decision making leads to more emissions and more inequality. Energy is much cheaper and easier to manage when it comes from fossil fuels than renewables, and nobody can see the CO2 or make you pay for it easily. Wealthier people can get more for lower prices because everyone wants their business. Poorer people pay more for everything because nobody wants their business. Inequalities get worse.

There is no motivation to help our children to develop into strong, confident individuals, because we can see they just end up feeling powerless against the machine, like their parents do. And no motivation to have any journalists, who try to understand what is going in on society. Nobody sees the point of even reading what they have written, let alone paying them for it.

Ending

But the reason this version of the world will never happen is that it has nobody to fix any problems in it.

There will always be problems, and always need to be people to fix them.

Some governments will be inclined to cover up all of their problems, rather than fix them. But not all governments.

Computers cannot fix problems themselves because they can only do what they are programmed to do. Yes, they can be programmed to fix specific problems which they have come across before. But you may have noticed problems have a way of being unique every time.

Only people can work out a solution, or an approach to a solution, in a completely new situation, including a situation where there is no data.

And only people can care about the world in itself and the health of everything in it, if it comes at even a small cost to themselves personally. That defies individualised logic so is something a computer could never do.

Computers cannot manage their own cybersecurity in the face of ever-changing threats, fix climate change when that would involve imposing costs on the wealthiest in society who use the most fossil fuels. Computers could never put together an education program, or even operate a farm. Not to mention keep our ships in operation, and balance electric grids.

Computers cannot regulate safety and security, something our society increasingly needs. That demands firstly that the systems to manage safety and security are clear to understand - which is perhaps a pre-requisite to even being able to regulate security - and puts a burden on the people who create the digital systems.

We are not asking here what computers can do. Computers can do a lot and will always be able to do more. We are asking what society actually needs and showing that people rather than computers are often better placed to deliver it.

And so long as we have a government, or business organisation, capable or forced to acknowledge their problems, they will put people in charge to fix it, and these people will demand clarity, from all of the people working to keep the big machine of human needs running.

And the people who keep the machine of human needs running – the human needs domain experts - will in turn continue to feel comfortable demanding clarity, transparency and integrat-ability from everyone, and everything they deal with, including their digital systems.