Chapter 9: An Emergency in a Smart City - Level Change in Action

9.1 Levels In Every Direction

In "Relativity" [9.1], the Dutch artist M.C. Escher drew staircases rising in multiple directions, staircases that not only went up and down, but also in and out and left and right. Little did he know that he was creating the perfect illustration for this chapter. However, unless you are trying to con clueless executives into giving you money [9.2], you can't reduce a systems integration architecture to a single picture. In fact, you'll need a whole, bloody methodology's worth.

A Systems Integration Architecture has to resolve a tangle of physical systems, data patterns and contradictory information mappings into a coherent portfolio that will bring many teams together to allow their disparate systems to work together as if they were a single design. In the 1990's I was involved in multiple R&D projects, each intended to develop a general solution to integrating "islands of automation". Every project reported success - but none developed the general solution promised. My analysis [9.21] identified a shopping list of 10 different dimensions to be included in the solution, noting that each project had tackled only three or four of them. Here I work through a couple of the dimensions, but only in order to flesh out the concept of level change and its application.

In IT culture, a single systems analysis/design **method** is a way of collecting and writing down a particular type of fact about the system-in-focus, such as which physical systems are involved, the interface standards to be used or the types of people who will interact with it. A **methodology** is a raft of such methods chosen to create a single (multi-dimensional), coherent picture of the system. Frequently, methods share facts, although they put them into different contexts, and computer-based tools are needed to propagate changes into all contexts that may be affected.

"A whole bloody methodology's worth"? Because most methodologies saddle you with huge volumes of tedious detail - at three volumes, the training manual for SSADM is admirably brief - for a methodology [9.3] [9.4]. To experience the level of tedious but necessary detail, try the 600 odd pages of check lists in the SABSA security methodology [9.5].

Even a single method may describe a staircase connecting views on multiple levels. But because each method is describing a different aspect of the system, the staircases they provide go off in different directions - into different dimensions. Consequently, what may be a single level from one viewpoint may involve multiple layers when viewed from a different perspective. That is, levels and layers provide one means of organising the detail about a single topic, and one talks of dimensions or perspectives to focus the discussion in other ways. And to bring things back together, one tells stories - works through examples - to bring out the links between the different perspectives, levels and layers.

Or perhaps, for a single, complex idea, divide a book into chapters, each chapter covering a single term of the idea, use notes to wander down rabbit holes, and include personal anecdotes as reminders that technical discussions about knowledge modelling are there to solve human factors issues about how much our brains can process in one go.

This chapter works through one example, based on building an Emergency Response Situation Picture in a Smart City [9.10] while interpreting the Smart City through a multi-level reference model for Integrated Vehicle Health Management (IVHM) [9.11]. Different sections of the chapter will go in different directions, though not all in the same 3D space. Perhaps illustrating how you could build Escher's staircases if you were not constrained by the gravity?

9.2 The Situation Picture

Start simple, start with a story. It's a spring tide, weeks of heavy rain have lead to the Thames bursting its banks and there is a strong easterly wind, so expect high water levels in the Thames Estuary. But, oh no! Someone has pranged the Thames Barrier, and it can't close and stop the tide - London is in for a major flood. Somewhere in an anonymous 1980's building near Victoria Station, the occupants of a windowless meeting room are being chucked out and the room readied as the UK Gold Command for the Fire and Rescue Service (emphasise rescue). From here, the various tier 1 responders - police, fire, ambulance, hospitals, port authority, railway police, etc. - will be co-ordinated. And possibly tier 2 responders as well.

Some 20 years ago, much of the co-ordination would have been through voice calls and video-conferencing. The policeman in Lower Thames Street would radio in rising flood levels to the City of London Police control centre, while the bobby in Flood Street would report to the Metropolitan Police. Their reports, possibly sent via Silver Commands, are eventually collated in Gold Command. Gold Command will probably have a hot line to the UK government, though possibly not the usual Cabinet Office Briefing Room [9.12], which is rumoured to be in a basement close to the Thames. Integration was more about knowing who to ring and chains of communication than computers.

The scenario we will be exploring imagines we are in a Smart City - interconnected computers and smart sensors everywhere. Somewhere in a corner of Gold Command someone is running a simulation predicting water levels street by street, while the rest of their team update the model with reports from the police and social media. Failures in electricity substations and traffic lights - sensors no longer reporting - are used as a proxy where there are no other data. Meanwhile in another corner, movements of phones and positions of buses are used to assess traffic flows along evacuation routes and to find access routes for emergency services. On-street cameras are used to identify the causes of snarl-ups, although people using live data from their satnavs are causing secondary traffic jams in unmonitored side streets. And in another corner...

A number of assumptions are needed to make this scenario work. Firstly, assume that complete communications stacks are in place - in the OSI model, this means application-to-application data transfers actually work, and not by some user retyping data from a different screen. Secondly, assume (big assumption) that there is an adequate trust infrastructure, so that, for the most part, information flows are not blocked by the lack of a decryption key or not being able to validate that a sender is who they say they are. Moreover, assume at the business level, organizations know how to work together - knowing that the fire appliance that is flashing blue is the local (bronze?) command, or knowing the civil servant from COBR has (or has not) the authority to order a large-scale evacuation and override senior police commanders. And assume...

In fact, assume that we can abstract our system to a collection of data sources and data consumers, and our objective is to connect them together so that we can extract knowledge about the situation - a Situation Picture (SP). Our design is about how we aggregate the data in meaningful ways: how we aggregate a 5v signal on a fuel tank sender with an 120 degree signal from a temperature gauge with a 12 KHz reading from an accelerometer, and correlate them with last months readings, the planned route for the fire truck and the observation of rising levels of panic on social media. And don't assume that everything - from sensors to communications links - is working. Having the data from a Smart City is only the start of the problem. The architecture below is designed to take this problem forward.

9.3 The Physical (to Functional)

The physical - sensors, networking cables, data centres and so on - describes both the physicality of the particular components and links the components to their physical location. A traffic sensor is not just a box of electronics, but also sits under a particular road junction - say where Gloucester Road crosses the Cromwell Road in West London. Cromwell Road is usually a very busy three lanes from the West End out towards Heathrow. A sudden drop in traffic might mean a sensor failure, an accident, or a royal convoy driving to Windsor [9.13] - the last follows from site specific knowledge.

People find the physical useful as a way of grounding the equipment description, but we will pass quickly from a physical view to one of several functional views. In mapping terms, we are in Tube map territory. A sensor drawn as a circle mimics a station - a function tied to a physical location - while the communication links become tube lines connecting circles - or perhaps ovals where were several communications routes come together in a switching hub (Londoners, think Camden Town). Our physical map will also include things like data centres - although the actual tube map doesn't care what passengers do at the top of the escalator. (Figure 9.1)

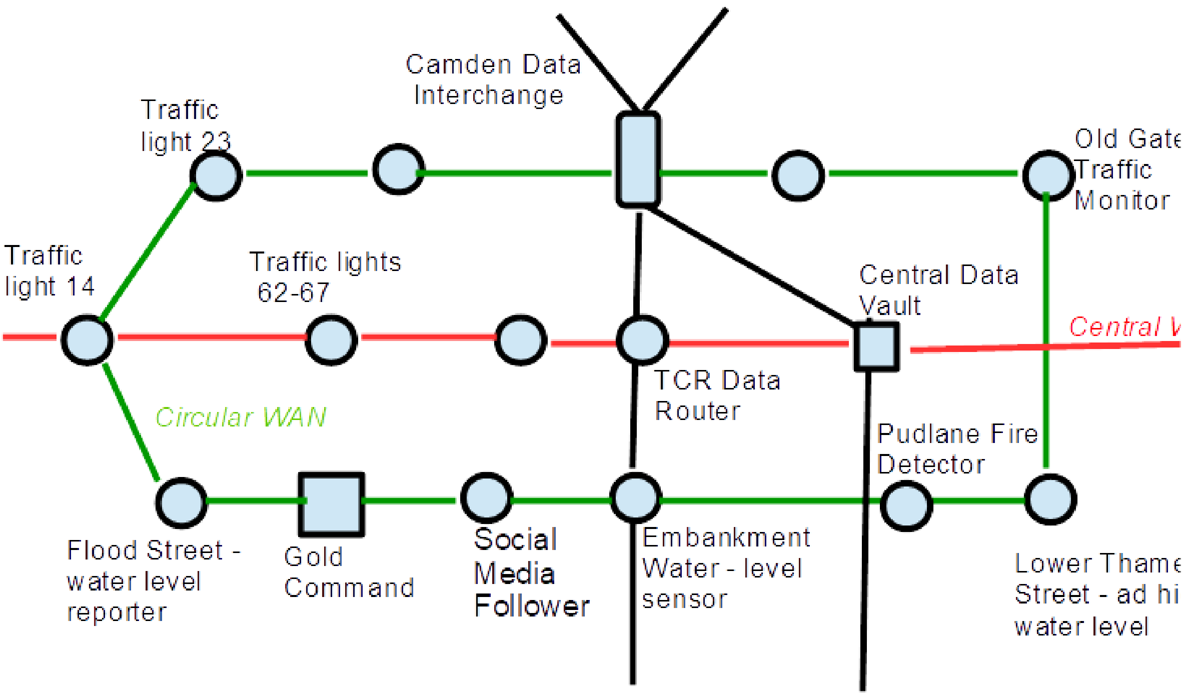

Figure 9.1 Physical to Functional

From a design viewpoint, this move from the physical to the functional is an abstraction - a level change - with important consequences for maintainability. At the functional level, all sensors will be treated as interchangeable, as are the communication lines. That is, the functional design will impose a series of interface standards (remember the OSI model), and replace a hundred thousand device specific interfaces by a single family from a standard interface. You yourself do this - you have a single nervous system that tells your brain if you are hungry, how heavy a brick is, or how cold is the water - different sensors connected by a single nervous system.

Not that you would replace a sensor measuring street-level pollution with one that tells you whether Tower Bridge is open - that wouldn't make sense. But this does point to a functional requirement - to maintain a register of sensors, both to record what the sensor measures and where it is measuring it. A little delving and the scope of the register expands rapidly: how accurate and precise are the measurements? when was the sensor last calibrated? how often does it report? and do we need to look at sequences of reports? That is, we will need to abstract physical aspects of the sensor into a systems model - it is the physical system that explains the signals the sensor produces, but the systems model describes how to interpret them. But we can still impose communications standards to make sensors interchangeable from the perspective of the physical communications system.

Sensors are just one part of the network. Another is computers - or rather, the information they store. This may be an asset register, listing the types of equipment installed - and possibly their availability and/or maintenance schedule. Personnel databases provide lists of people, their qualifications and their skills. Or Wikipedia for background knowledge, social media for (unreliable) reports of what is happening in a neighbourhood, street cameras, the text message alert system, and on and on.

Again we can abstract away from the physical box in the corner, although if we were adding it to our tube map we might give different physical nodes different symbols - a square for a data centre, say. This tells the system maintainer that the node not only produces data, but also eats, digests and stores it. And, from the viewpoint of our situation response, it is the big square that maintains the situation picture for our responders.

9.4 A Functional Model of the SP

The functional goal of the system is to collate and integrate information about "the situation", and make it available as a Situation Picture. The "functional model" shown in Figure 9.2 has the role of identifying and differentiating the key functions of the situation system.

The central block is the Situation Picture (SP), a virtual database [9.14] containing the facts of the situation, together with knowledge of how to use those facts. Data gets pumped up into the SP from sensors, corporate systems and so on. Once data is in the SP, it is pumped up through knowledge engines which collate the data, check it for consistency, or make predictions about how the system will develop. The knowledge thus extracted returns down into the picture for viewing and further use. Viewing platforms (specialist applications) look down into the SP to provide situation views, which will typically be tailored to a group of responders, based on the situation elements that directly affect them, the service they represent, and their level of expertise. Some of these viewing platforms will take a summary picture and distribute it to responders outside of the command hub. A tunnelling viewer (not shown) is one which can drill down through the SP to the original data sources to find data not replicated to the SP.

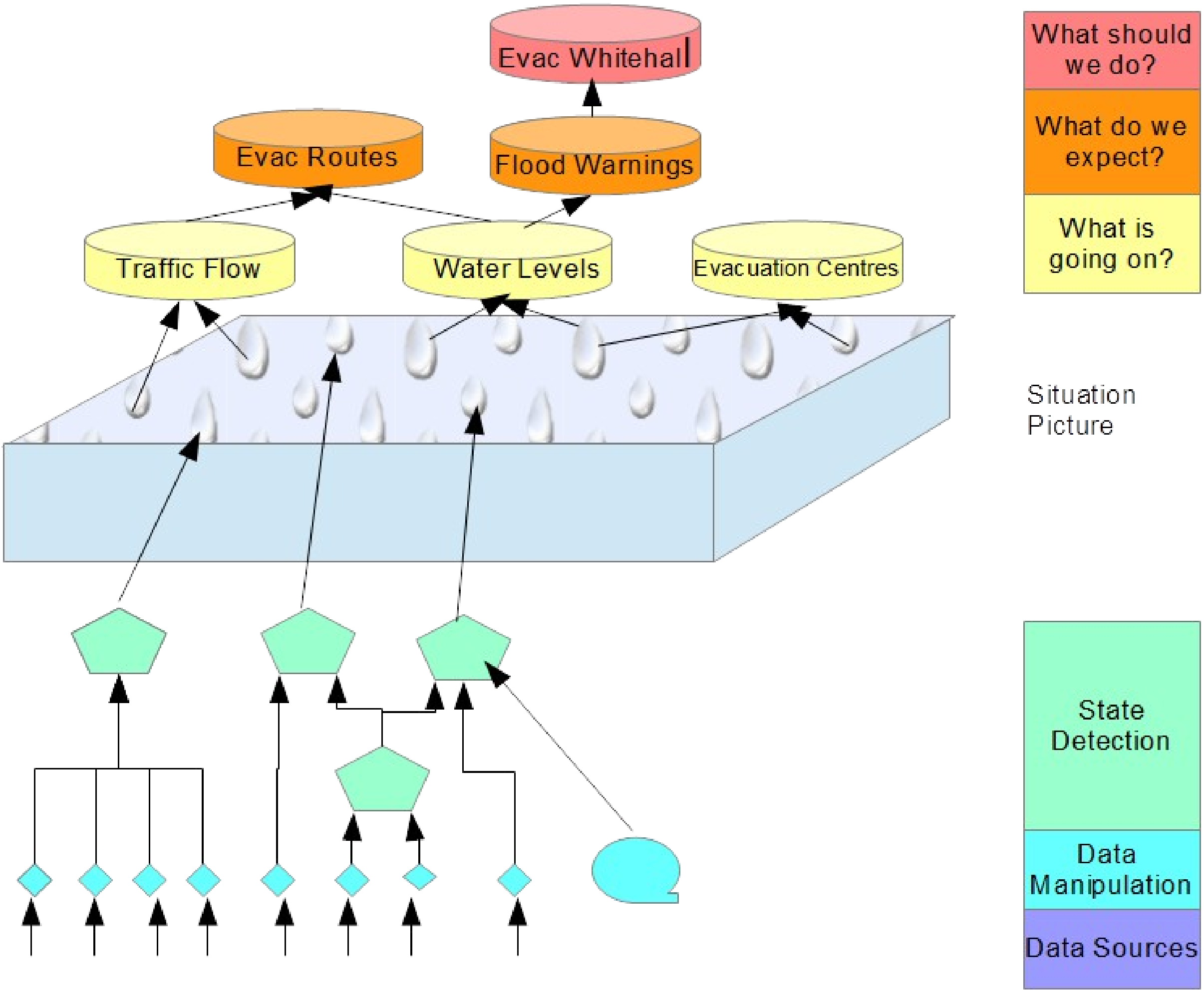

Figure 9.2 A Functional Model of the Situation Picture

Let's ground this in an example scenario. The Gold Command centre is activated on the expectation of a very high tide, and have also heard on the news about a ship running into the Thames Barrier. The Situation Picture system has been activated, and is pumping in data from water level sensors (including an officer phoning in readings from the Embankment wall). Two simulations developed at two different university colleges are sucking data out from the situation picture, making estimates of the likely river levels, and starting to colour in maps of possible areas of flooding, predicting 1, 3 and 6 hours ahead. Sand-bagging teams from the local authorities (secondary responders, who get a summary view of the SP) are being readied. Police stations and patrol cars in high risk areas are sent a tactical situation picture saying "Risk of Major Flooding Event - prepare to evacuate" together with a map showing the areas that are predicted to flood (from the universities) and the location of evacuation sites (from existing contingency plans). And an alert has been sent out to the public on their mobile phones and traffic is being monitored to see where jams are building up.

A complication that is immediately evident is that the SP must cope with many different information lifecycles, from many updates a second from a traffic flow sensor to several times an hour for the flooding simulations, and to months between contingency plan updates. Also, it may not be practical - or desirable - to replicate the entire content of a data source to the SP - that data is going through its own lifecycles within its original source. In such cases, a tunnelling viewer gives access to the data in its native environment, pulling out a snapshot of the latest state of the data when it is needed. Stairs going leftwards, beyond the scope of this brief discussion.

Also beyond our scope are the human factors stairs going diagonally through the arch - how does one ensure that the various users are familiar with the system. The basic system must be useful day-to-day to ensure that when used in an emergency, users are familiar with it - for example, the fire service will use a tactical situation picture every day showing where fires are, how big they are (bin fire or tower block) and who is attending (just us, or forty crews from across the area).

And spiralling through a dungeon dimension is the whole business of data reliability. This ranges from basic assessment - is the sensor broken? is the source reporting a fact? an opinion? a rumour? - to dealing with deliberate attacks on data integrity such as fake news, hacking or creating the appearance of a traffic jam by towing a hundred mobile phones in a handcart.

9.5 Using Levels to Structure the Situation Picture

So far the Situation Picture has been described in terms of the functional components surrounding a pool of data, such as data and knowledge pumps. But to this point the data has been just a big swimming pool, not unlike our knowledge ramen. In this section I start to put some structure into it, but structure beyond the usual entity-relationship gubbins of an information model. You should not be surprised to learn that, like Escher's staircases, this structure is elaborated in multiple directions.

{Read the next three paragraphs as if you were James Bond driving a tank in a high speed car chase through a city - lots of detail rushing by, smashed cars, explosions, and you have no idea exactly where you are or who is chasing who. Read for the thrill of fast-paced action, rather than the detail of each skid and squash.}

First, use the meta-model from Integrated Vehicle Health Monitoring (IVHM) [9.15]. This provides a way of consolidating different types of data manipulation and analysis into knitworks with a particular purpose. The IVHM model is based on a hierarchy of levels, starting with sensors and rising up to the level of planning vehicle maintenance - comparable here to the need to work from sensors to emergency planning. In this section this is treated as two separate problems: a basement in which sensor data is consolidated into a system in order to expose it to the SP. Above that is the conference room, in which system states from the SP are brought together to build situation awareness - what just happened? what is happening now (and when will it finish?), and what is about to happen (and when should it start?).

Second, while the IVHM model gives clues to the methods we should apply to each level, we also need to know where to apply them. That is, the SP shows the state of various systems/processes, but each of these systems need to be identified and put in context. This is tricker than a simple catalogue because *system* is a recursive concept. A car (as a functional system) contains various subsystems - engine, braking, lighting and so on. Individual systems have their own subsystems - the engine has a fuel pump, a subsystem controlling timing and ignition, a cooling subsystem, not to mention the mechanical system combining engine block, pistons and crankshaft. The health of each of these subsystems feeds into the health of the overall system (the car), as is shown by the array of fault warning lights hiding in the dashboard. To link into information models developed in industry, call this structure of systems within systems a *product structure*, and then follow though to the way that a product may have multiple product structures [9.9] with complex mappings from one to another.

System models create views on the SP which are themselves level changes, explicitly adding a "system" relating the components made visible in the SP. That is, the inferences applicable will make use of knowledge about the system as a whole which will go beyond the data for the components. Driving a car (or tank) is not simply a matter of knowing if the the windscreen wipers are turned on.

9.5.1 The SP Basement

Data pumps in the basement extract the basic information from sensors and sources. They pump the data through three levels: Data Acquisition, Data Manipulation and State Detection. The IVHM model is oriented to physical sensors in a vehicle, and needs only limited reinterpretation to apply it to the Situation Picture - a change to the mapping that binds the model to the real world.

At the level of Data Acquisition, we are collecting the basic facts: the raw signal from a sensor or an entry in a database. At this level we need the knowledge that describes the sensor/database. This should be contrasted with the Data Manipulation level, which converts the raw data into attributes in the information model - the output of the data pumps.

Example 1: the sender on a car petrol tank will output a voltage proportional to the amount of fuel in the tank (the Data Acquisition Level). Knowledge of the range of voltages and their relation to the fuel level will be used by the Data Manipulation level to drive a fuel gauge to show a value between full and empty and to turn on the "Fuel Low" warning. The fuel will slosh about in the tank, so the level at the sensor will change many times a second, whereas the person viewing will want to see a definite level which does not wobble about - data manipulation will smooth out the signal (a signal processing function). That is, the Data Manipulation level transforms the data from a source specific view to a common view.

Example 2 & 3: the arrivals board (a database) at an airport will show local time (data acquisition level) which needs to be converted to the time zone used in the situation (data manipulation). Or image recognition will have recognised a photo (an array of pixels) as showing a car and pick out its registration number.

These two levels have two roles. The first is to isolate specialist knowledge about the sensor and its signal away from the situation picture and then present it using standardised information to merge the data into the SP. The second is to compress the information, both in quantity and over time, to match the capability of the SP and the systems (including people) that process that information. (In one job, my goal was to compress four 10MHz data streams into a single symbol to warn a pilot of approaching obstacles).

In addition, because it is closely linked to the data source, the Data Acquisition Level also provides knowledge that might be classed as provenance and trust metadata, such as data sensor used, sensor configuration and calibration, as well as context data such as what system the sensor is used in and where it is located.

The next level up (but still in the basement) is State Detection. Essentially, this brings together multiple sources to create state vector for a system or a process. This can then be transformed into a view matched to the user, for example, by summarising the system state as "working" or "failing. For example, my car is "working" if the engine is running, the fuel gauge is "half full", and there are no red lights on the dashboard. I may be stuck in a traffic jam or had a wheel clamped, but these are out of scope for the state vector. Again, State Detection helps match the data sources to the data consumers, providing data volume and time compression. As importantly, it provides knowledge compression - all I need to know is that my car has stopped working, but the garage fixing it will extract many different measurements from the engine management system.

And this points to another staircase running off to a side dimension - the product structure - the state for a situation may well be a combination of the states of multiple systems: I have forty fire and specialist tenders at the scene, all of which are working normally, plus a local silver command which is in touch with gold command. The system is working and situation is under control - though maybe not the fire.

And, by the way, there are complications. Firstly, a single data source may feed into many different systems models. Systems models may take the information from a number of lower level systems and may also feed multiple higher level systems. And the whole is not just a snapshot of the current situation, but involves continual monitoring of the systems, both to learn their typical state, and to flag up any deviations from normal - when I drive my car, I quickly notice when the oil warning light suddenly comes on.

9.5.2 The Information Model Conference Room

The next three levels of the IVHM model are Health Assessment, Prognostic Assessment and Advisory Generation - in terms of our situation awareness model: what is going on, when is it going to change, and what are the options for what should happen next. The focus here is not getting information into the SP, but using the SP to build up knowledge about the situation.

Rename IVHM's "health assessment" as "current situation" and we can see that the goal of this level is to work out what is going on. That is, this level is not looking at individual systems, but bring together the most important systems to create an overall picture: which areas are currently flooded? are there enough rescue crews in attendance? is a civil disturbance contained? are evacuation routes open? Summarise the Health Assessment level as answering the question "How bad is it?"

Going up to the next level, the Prognostic Assessment is there to predict how the situation will develop in the short term: if the flood has got this far, will it rise to overwhelm flood barriers? is the fire likely to spread? is the civil disturbance spreading organically, or does there seem to be a focus? will the numbers of evacuees exceed the capacity of active receiving centres? These are the sort of make-your-hair-go-grey questions that the gold command teams will need to grapple with before things go pear-shaped. Summarise repurposed Prognostic Assessment as answering the question "Will the situation to go from bad to worse?"

And we can summarise Advisory Generation as "If you do this, you can fix the problem". In the IVHM model, the *state* of a drive shaft might be "working with high levels of vibration", the *health assessment* will be " acceptable vibration levels exceed", the *prognostic assessment* might be "the drive shaft bearing will fail within the next month", and then the basic *Advisory Generation* should come up with "put the ship in dry dock as soon as practical" although the sophisticated one will say "the dry dock is 8 days sailing away - complete current operations within two days and sail there directly" plus "send instructions to the dockyard to prepare to receive a ship for maintenance". In the situation model for our flood, Advisory Generation will throw out messages such as: build a temporary dam across the road under the railway; put crews in place to spray cooling water on the petrol storage tanks - there isn't enough time to empty them; deploy the riot squad to block the road to the town hall; activate the emergency response teams in the next county and get them to open emergency shelters.

Are these levels enough? We once had a disaster in the hanger where I worked - exceptionally heavy rain overwhelmed the drainage system. We needed a Jekyll and Hyde disaster manager. Dr Jekyll would be a careful, systematic person who would have spent years working through different scenarios, reviewing interlinkage and knock-on effects, and developing detailed plans to respond. Mr Hyde would be a wild, innovative character, acting immediately, cutting through bureaucracy and process, a person who would pay out on the corporate credit card before he had the three signatures needed. However, Mr Hyde would not have sat down and read the response plans in advance. Although the Advisory Generation system would hold the details of Dr Jekyll's plans, the human factors problem is to get Mr Hyde to read them at all - yet anther staircase spiralling off into the human factors dimension. I'd worked with our Dr Jekyll, although watching him stride through the hanger, I suspected he might do a passable impression of Mr Hyde.

PS While writing this chapter (October/November 2024), there were major flash floods in Valencia. The governments (National and Regional) are being criticised for acting too slowly. Add another dimension to the Emergency Response system - updating dynamics of the processes needed to stop people acting wildly, but which also need to respond to events much larger than ever imagined - roughly, the meta-process that says "this is a mega problem, increase Mr Hyde's spending limit by a factor of 100!".

9.5.3 Product Structures and the SP

Did you read the appendices to Chapter 1? If not, you might like to nip back to look at the Bill of Materials example (1A3). As a reminder, a Bill of Materials is a tree of parts, sub-assemblies and assemblies that join together at the root to provide a list of every part that goes into a product. It is called a *product structure* because it describes one way the product is structured. It is a structure of physical assemblies, and these have their counterparts in the assembly processes - each stage in the process brings together the parts listed to build up one of the assemblies.

In system design, product structures are based on the parts that come together to build systems and subsystems. This is not the same as the physical assembly structure, If you order a new heating system for your house - boiler, radiators, and the plumbing connecting them - you will rue the day when you ordered it from a company that delivers it fully assembled - it won't even fit on your drive. System design ensures that all the pipes connect together and there are no leaks. Physical design delivers the physical assemblies that fit through your door. Design involves the separate development of all the different product structures needed - physical and systems design are just the start of the list. [9.16]

The IVHM model acknowledges that signals are brought together into systems, and that some systems are contained within other systems, but the information models that result [9.17] reflect only the six level model, not the system/subsystem structures that are overlays to it, nor the way that systems interact as a systems-of-systems with its own whole systems behaviour. Industrial models such as STEP [9.18] are somewhat more helpful, in that the first level of information structure below *Product* is a *View*, and each *View* has its own product structure. It even allows for a Support Processes view, in which the product structure contains *Slots* in which a particular *Product* (subsystem/assembly) can be slotted in - in an organisation structure, at the end of a shift, a new team with the same skills will slot in to replace the team who have been working through the night.

One use of such product structure comes when dealing with subsystem failures. For example, does the failure of a traffic sensor at a road junction affect only the reporting of the traffic flow? or does it affect the traffic light sequencing, and hence the whole junction? and how does this affect the routes that go through the junction? And are there other sensors that can fill in the missing information? or even take over from the failed sensor? Product structures go off into other dimensions (with associated level changes) that help answer these questions. Moreover, the structures will be part of the knowledge used by knowledge pumps to feed things like simulation tools.

9.5.4 The Situation Picture - Iteration 2

So, what does the SP look like once we have added the IVHM model and Product Structures? Imagine the "floor" of the SP as being a mosaic, with each of the tessera representing a *System* - recording the system *state* together with its context and provenance. On top of this are laid mats and carpets - say the "current situation of the road network" mat laid over the tessera for traffic sensors, or the prognostic assessment carpet showing the expected rise in flood levels which bring together water level sensors, geography and reports from social media. Let me elaborate further.

Bringing together a collection of individual system states does not explain the whole system behaviour. For that, there needs to be an explicit system model fed by the states of the component subsystems - a knowledge pump sucking in many system states and pouring back a system-of-systems state. The floor of the SP is composed of all the system states which are relevant to building the situation picture. These may be as simple as the output from a single sensor, transformed into a standardised measurement, set in the context of its location and its sensitivity, supported by the provenance of system test records and the time of the reading. Or this may be as complex as the whole city traffic flow, combining many different systems using not only their current state, but also the recent history of the systems. The floor of the SP is itself a multidimensional structure, but its up/down runs at 90 degrees to the IVHM model.

There is a design decision dictating which system states appear on the floor of the SP, and which are consolidated into a higher level system. This decision may be constrained by the information exposed by the contributing systems, the capacity of communication links, or the ability to process the information [9.19]. It is oriented by the inputs required by the various conference room assessments. In some cases, this may mean exposing both a collection of systems and the system-of-systems they contribute to, whilst in others, only the system-of-systems may be relevant. Even where systems are not exposed on the floor of the SP, tunnelling viewers may provide such detailed insight.

This further implies that data pumps may have functionality beyond simply sucking up data from the environment, but some may actively consolidate systems data, either before it gets to the SP level, or by sucking data from other systems on the SP floor.

The responders using the situation picture will not usually look at the state of individual systems, but at views consolidating them into "the current situation", "predicted developments" or even "suggestions for immediate action". These may cover as little as a single topic in small area or an all-encompassing picture. The tricky (human factors) problem is to match the scope and detail to the capacity of a responder to deal with it. For example, the Tactical Situation Object [9.20] is a whole incident picture summarising the incident into a small number of details - sufficient for a fire crew or police car arriving at a scene, but not enough for detailed planning at Gold Command.

Compare this with an in-theatre military command post: individual teams specialise in planning troop movements, or fire control, or logistics, or casualty evacuation, or etc; the command tent brings their work together under the guidance of the senior commander and their aides. And there are teams that will take over to give 24 hour cover. Their "conference room level assessments" may range from simply automating the clerical tasks - collating the location of all medical staff - up to AI based tools proposing detailed plans. The SP provides a tiling of all the information sources, oriented to meeting the information needs of each team, and makes their reports available at the commander's planning meeting. And running off in yet another orthogonal dimension is the organisational structuring of the teams involved.

BTW, the fact that we are quickly getting into more than three dimensions is not a problem. One of my colleagues in Maths Modelling (up half a level from me) used to work on ten dimensional visualisation.

9.6 So What?

From the perspective of working with knowledge, the Situation Picture looks like a hot tub full of ramen - nowhere near as big as a swimming pool for knowledge in general, but churning furiously - a picture taking in data from many sources, mixing them together, and in which several people can immerse themselves. Too big to eat as one bowl full, but not so big that one could not imagine ladling it out - say a couple of thousand one litre bowls, served at one a minute by each of you and an assistant during a music festival, with a third partner taking the money - empty in time to watch the headline act.

And that is precisely the point. How can one organise a complex problem into portions small enough for a person to digest in a reasonable time? Function, level and product structure have been used to create a concept design that could lead from initial idea to completed project. The structures are there to identify small sub-knitworks (bowls worths) that can be partitioned out to different implementers.

Say you assign a single sensor to a junior engineer: they should document the measurements made, transform the raw signal into a common standard, expose supporting information and generate provenance. Possibly a couple of days work the third or fourth time they do it, but a couple of weeks the first time through. The senior engineer supervising will allocate work based on the effort required for each sensor and on the knowledge and capabilities of the junior engineers - including putting together a reading list for learning-curve-climbing for the least experienced.

Our senior engineer comes from a group specialising in sensor technology, so allocating them the lower two levels of the IVHM-based model stays within the scope of their knitworks. However, a systems engineer will be needed to develop a behavioural model for each system and allocate sources to its state vector. They will use knowledge of the system as a whole, including what the sensor values indicate - 300 degrees from an automobile air temperature sensor means either that the sensor has failed or the car is on fire. Higher level specialists will merge the models of subsystems to develop their whole system properties. The development of health monitoring and prognostics level is likely need teams containing multiple specialisms - team who can merge multiple knitworks.

My role as the original architect was to breakdown of the overall concept into team-sized knitworks - as the design this chapter sketches out. The initial idea took an hour to sketch out - in a commercial organisation, one doesn't get much time to work through ideas. If the bid had been funded, I would have started by taking several weeks to work through a multi-view methodology - probably MoDAF - while the team was set up and briefed. Cost wise:

- Initial idea* - £70;
- MoDAF architecture* £25, 000;
- Project budget - £1,000,000.

* calculated at 2014 rates, but excluding sunk costs of nearly forty years experience and £500,000,000 to develop the industrial standards reused.

But you do need a coherent technical approach to ensure the million in funding is well spent. And, as I have noted elsewhere [9.20], historical integration projects often only addressed three or four out of of the ten problem dimensions. Level change in multiple dimensions is the way I would organise such a project.

To summarise the knowledge structuring dimension in an example:

- Knowledge of electronics gives a signal 10v;
- Knowing the signal and the sensing mechanism gives a detected water level of 2.1m
- Knowing the sensor system location gives a water level of 105.2m above datum [9.21], and the history of the last ten readings shows it is increasing at 0.02 metre/minute
- Flood defences work up to 105.4m => health assessment of "Flooding expected very soon";
- Knowing also the time to peak high water gives a prediction of 1m of water on Whitehall;
- Advisory: evacuate Parliament and ministry buildings now.

Explaining to the Prime Minister simply that "We have a 10v signal, shut down the government" is unlikely to be career enhancing. Although going through the detail at each level, and explaining how one level reinterprets the data of another may well exonerate you at the public enquiry after.

Overall conclusion: level-change makes sense, and is a useful tool for dividing up complex problems into smaller knowledge domains. It also avoids magical thinking - that box on the diagram which says something vague, like "system dynamics", but really means "something magical happens here". Several times in my career I was lumbered with "doing magic".

Level change is one of a number of technical devices used to structure knowledge - partition knitworks - in order to match it to the capabilities of the human brain. The difficulty is that we change level all the time, like breathing or walking, so that we barely understand the mechanism involved - until now. End of chapter - relax intercostal muscles (breathe a sigh of relief), tension calf and thigh muscles using you inner ear to adjust the rate of motion and maintain stability (stand up) and make a cup of tea.

Notes and References

9.1 Copyrighted - you're intelligent, so look it up.

9.2 I reviewed some 60 or 70 "systems diagrams" for a client. They were essentially only adverts, and could roughly be classified from the number of blocks shown:

- 4 block - we have an idea, give us some money to develop it;
- 8 block - we mostly do the big block in red, give us plenty of money to develop it;
- 16 block - this is too complicated for you to understand, give us a shed load of money.

Such diagrams can be useful, but only if the blocks and the interfaces between them mean something specific.

9.3 [Yellow Banana] The National Computing Centre, "SSADM Version 4, NCC Training" 1991

9.4 Although SSADM became the required methodology for my department, I used it only once. SSADM implicitly assumes that you are starting from a vague user wish list and are going to end up with a large database. However, provided you justify it, you can omit parts that you don't need. I was building a system to merge message elements produced on different computer systems, so didn't need the initial requirements methods, nor... Well, in the end, I only used one diagram (and, yes, that was taking the piss). Dissatisfaction with such methodologies has led to a proliferation of alternatives such as Spiral Development [9.6] and even anti-methodologies (sic) such as SCRUM[9.7] and Extreme Programming [9.8]. But use with caution [9.9].

9.5 SABSA "Executive Summary" https://sabsa.org/sabsa-executive-summary Accessed 20/1/25 and [Brown Banana] Sherwood N, "Enterprise Security Architecture" Originally published 2005, CRD Press (610 pages) Warning, this book works systematically through all 36 cells of a Zachmann Framework, For Zachmann see Wikipedia, "Zachmann Framework" https://en.wikipedia.org/wiki/Zachman_Framework accessed 19/11/24, but there are many other sources.

9.6 Wikipedia "Spiral Model" https://en.wikipedia.org/wiki/Spiral_model Accessed 19/11/24

9.7 Wikipedia "SCRUM (Software Development)" https://en.wikipedia.org/wiki/Scrum_(software_development) Accessed 19/11/24

9.8 Wikipedia "Extreme Programming" https://en.wikipedia.org/wiki/Scrum_(software_development) Accessed 19/11/24

9.9 Barker S. "Aircraft as a System of Systems" SAE Warrendale 2019, see particularly Ch 3 "Model Maturity"

9.10 Wikipedia "Smart City" https://en.wikipedia.org/wiki/Smart_city 19/11/24

9.11 wikipedia "Integrated vehicle health management" https://en.wikipedia.org/wiki/Integrated_vehicle_health_management Accessed 19/11/24 and [Brown Banana] Mimosa "Mimosa OSA-CBM" https://www.mimosa.org/mimosa-osa-cbm/ accessed 1/11/24 and OSA-CBM = "Open System Architecture for Condition-Based Maintenance"

9.12 Wikipedia "Cabinet Office Briefing Rooms" https://en.wikipedia.org/wiki/Cabinet_Office_Briefing_Rooms Accessed 19/11/24

9.13 Having lived for several years in Central London, I have wondered how self-driving cars would cope with the occasional blowing of whistles, usually a sign that the police are setting temporary road blocks to let a diplomatic convoy pass.

9.14 Virtualisation separates the function of a system from the physical hardware that provides the function. This can mean that what appears to the user as a single database is actually distributed across multiple computers, possibly located in several different data centres. Cloud computing is a suitably vague term for this. Benefits may include the ability to continue to operate even when some of the hardware systems are down, or being able to access multiple physical databases through a single query. One useful discussion for interoperating databases is OAIS (Open Archival Information System CCSDS 650.0-B-2 https://public.ccsds.org/Pubs/650x0m2.pdf Accessed 19/11/24)

The level change from the physical to the function relies on abstraction to make the change, but also facilitates virtualisation - the mapping from a function to the hardware that it runs on "happens magically", meaning that it is someone else's problem to think it through. Conversely, it means we can pretend that mapping does not happen - that a particular function runs on a particular computer, and that we can imagine it is one big computer rather than lots of different computers chatting together.

9.15 [Yellow Banana] ISO "Condition monitoring and diagnostics of machines — Data processing, communication and presentation — Part 1: General guidelines" ISO 13374-1 First Edition 15/3/2003 also at https://cdn.standards.iteh.ai/samples/21832/4f282cf6f5594b73be0bbca7590719f1/ISO-13374-1-2003.pdf Accessed 19/11/2024 (or see Wikipedia)

9.16 Those will an addiction to detail can read my book [9.9], particularly chapter 6.

9.17 [Brown Banana] MIMOSA OSA-CBM 3.3.1 29/6/2010 https://www.mimosa.org/specifications/osa-cbm-3-3-1/ Accessed 19/11/2024

9.18 STEP, ISO 10303 is very Brown Banana, so see my book [9.9] instead.

9.19 For example, IVHM data for a single transatlantic flight of a commercial jet may run to petabytes

9.20 ISO "ISO/TR 22351:2015(en) Societal security — Emergency management — Message structure for exchange of information" based on CEN workshop agreement CWA 15931 published in March 2009. Accessed 21/01/25. I was vice-chair of the original CEN workshop.

9.21 Barker S. "Here be Dragons: An Integration Shopping List" IEE Computing and Control Engineering Journal 12/2000

9.22 https://www.tideway.london/media/1302/site-plan_victoria-embankment-foreshore.pdf Page 19