Chapter 11 The Zag - Semantics with Complications

11.1 The Lobby Staircase

The last chapter saw semantics as a coming together of three knitworks that differentiated subjects, detailed them, and acted out their meaning - a sort of lobby, where different interests jostled for attention. And as this is the last substantive chapter, I thought we ought to go out on a grand staircase. In particular, the staircase that rises up from the central hall of the Houses of Parliament - lobby for the House of Commons to the left, House of Lords to the right - and up connecting to the committee room corridor. Apt too, as the Commons and Lords chambers are where the main lines of the legislation are laid out - the last chapter covered the main lines of my engineering model of semantics - but the committee rooms are where the detailed scrutiny occurs - and in this chapter we start to look into more details of the model.

Some disclosure: in the late seventies I was part of a lobby group on IT policy - not so much lobbying for something. rather arranging for two sides of a debate to argue their case in front of MPs and Lords to make them better informed. This usually took place in one of the committee rooms so that it was easy for them to attend. At the time, we caught the wave of interest caused by the rise of the "microchip" - the early eight bit microprocessors that powered the Apple 1, ZX-Spectrum and the Altair 8800. I brought in my Nascom 1 - a Z80 based machine with 1K (yes, only 1024 bytes) of RAM - probably the first micro-computer to be taken into Parliament, and shown off in the Grand Committee room off St Stephen's Hall.

What started as a few, small scale briefings - the most important being the lobby to break the telecoms arm of the Post Office away from the Royal Mail - culminated in a large debate about whether the Inland Revenue should buy IBM or ICL computers [11.1]. As a shy, relatively recent graduate, and seeing five past presidents of the British Computing Society contributing, I stepped back after that. Although some years later, following the public announcements of US and the UK cyber warfare units, I wrote a short paper on the foreign policy implications of cyber response, describing a grey zone where the impact of an attack combined with difficulties of attribution were below the threshold needed to launch an overt response - a risk-based differential knitwork.

While not overtly political, this chapter takes examples from government to explore some of the complications when it comes to semantics, choosing examples from Security Classification, self-assessment tax forms, and exam grades. I follow this with a couple of case studies from the only slightly less political world of information standards. Then we go up a level or two (you would be disappointed otherwise) to discuss the semantic processor, the meta knitwork that links together the three knitworks that define terms. Finally I mention the problem of semantic drift, and by that point, hopefully, you will know what this whole book has been about, and we can finally move forward to the Introduction.

1.2 Example - Security Classification

Taking a simplistic approach, a Security Classification is a marking applied to a document or file which identifies the measures needed to keep it secure from prying eyes. Security classifications are used in the access control process to enforce both rules on the security of the system storing the data and constraints on the handling of "documents". For example, handling includes forwarding: you

might be both forbidden (by the rules) and prevented (by the application) from e-mailing a confidential document on a public network. And Yes, we are back in the world of PDM where security classification is classified (other sense) as metadata.

Here I distinguish *SecurityClassification* from *Caveat.* A *Caveat* does not imply additional controls on storage or handling but rather restricts the circulation to a pool of people within the broader group who have the security clearance to read it. For example, the US caveat "US Eyes Only" means that a UK citizen may not read the document, even if they have the necessary security clearance.

When implementing a PDM system, one would add two attributes to each mathom: *SecurityClass* and *Caveat;* create a dictionary of values for each attribute; and build in software controls that check the users' credentials and apply the corresponding handling and distribution rules. All we need do now is check government guidance on the names used for attribute values and the rules that apply to each name.

In 2014, to simplify the system, the UK government issued revised security classifications to cover the whole of government, including both the defence and civil branches. The original five were *Unclassified, Restricted, Confidential, Secret* and *Top Secret.* Additionally they listed *Protect,* which I have only seen in the document that made it obsolete. Having worked for a defence contractor, I had been trained in the procedures for the lower level classifications, and which had always seemed perfectly straightforward. The post 2014 system was trumpeted as simplifying the system to only three levels - *Official, Secret* and *Top Secret* - with *Official* replacing the four lower tier classifications [11.2]. In addition there is *Official Sensitive,* which is presented as a modifier to *Official*, rather than a separate classification.

Firstly, note that *Secret* and *Top Secret* carried across unchanged from the original system, so don't need further discussion - except to note that IT security is so weak and fault-ridden that the best option is to air-gap any *Secret* system from general connectivity, despite that needing multiple computers per user and having to replicate all the cabling. [11.3]

The next decision is how to handle *Unclassified*. If there is an attribute that could be set to *Official*, then it also needs a value for *Unclassified.* It cannot be left blank, as it must be checked each time the file is accessed, and even if the value is Null, then Null is a data element, and the information it conveys is still *Unclassified.* That is, it is logically incorrect to say that *Unclassified* no longer exists as a security classification, even though it is no longer approved as a security marking. The absence of an unclassified marking must be irritating in a secure environment, where any unmarked document is looked on with suspicion.

The guidance on the new classifications [11.2] is presented in three columns, one each for *Official, Secret* and *Top Secret*. However the *Official* column is frequently subdivided to provide separate guidance for *Official* and *Official Sensitive,* and in many cases, the access rules differ. From the point of view of the practical knitworks involved, there are two separate knitworks for *Official* and *Official Sensitive*, and any data management system (which implements the practical knitworks) will treat these as two different security classifications. That is, from a data management perspective, the number of security classifications is five, not three. The cynic in me is apt to infer that "reducing the number of classifications" was civil service smoke and mirrors to satisfy political masters, but which left the original system unchanged.

Moreover, from the human factors perspective, the cue word "Official" has an ordinary, day-to-day

meaning which is unrelated to any security classification. As a local town councillor, I sometimes get an e-mail from a higher tier council (district or county) - such as a notification of grants for community groups - which is marked "official". The contents must be disseminated widely if they are to be useful - no access controls are intended, as if it were *Unclassified*. Such a poor choice of terms has led to different interpretations of the subject knitworks, and in turn led to different applications of the access control rules - you cannot hijack ordinary language to use it in a peculiar way and then expect it not to cause confusion.

11.3 Example - The Inland Revenue

I was on secondment in France when first I heard - an item on Radio 4 news said it was the last day for comments on the Inland Revenue's plan for Income Tax self-assessment. I rang up the IR, but, as expected, the phone lines were permanently busy. I had wanted to say that, in my professional opinion as a specialist in information exchange, self-assessment was technically impossible. To be more precise, unless they published detailed criteria (practical knitworks) in plain English, there was a risk that people would interpret tightly defined technical usages in the flexible way that natural language is used, and one could not be sure that one had answered the self-assessment questions accurately.

For example, since I was working abroad, among the first questions on my income tax return were "was I resident?", "was I domiciled?" and "was I ordinarily resident?" - technical appropriations of ordinary English words. To be fair, in this case they did provide several pages of guidance on how to work out the answer, involving counting the days one was back in England. However, the definitions were given through independent practical knitworks for each term, without setting them in the context of a differential knitwork. Consequently, it wasn't clear if this was an exhaustive set of categories, nor whether they were mutually exclusive. Consequently, each question would need to be assessed independently, rather than working through a single decision procedure.

Self-assessment uses the mechanism of "filling in a form", which, in practice, is a type of data exchange. At one end is an individual with no training in the processes and procedures of the Inland Revenue. At the other, a large organisation which has developed a complex technical vocabulary to run its processes and to defend them from malefactors. These processes and the vocabulary involved can be opaque to anyone outside the IR. Being on secondment, at least my company provided accountants to help us with our tax forms. Anecdotal evidence from them included "we sometimes don't know how the IR will assess something until we get their response back". That is, the terms that determine which branch of a tax calculation to take - the practical knitworks that ground their meaning - were opaque outside the IR, even to professionals - self-assessment was technically impossible.

Earlier versions of this chapter have more examples of such technical quibbles, but I also perfectly understand the need for a precise technical language for tax, especially in a world where accountants and lawyers are employed to wriggle round the words in an effort to avoid tax (and why there is a subtle distinction between tax avoidance and evasion). However, a more systematic way of exposing the semantics of terms is needed - at the very least publishing the subject, differential and practical knitworks that are needed to understand the terms aright. BTW Mr Inland Revenue, you are a beacon of light and clarity compared to many government departments, though perhaps only an old fashion 20W tungsten filament bulb.

11.4 Example - Exam Grades

Most years, when school exam results are published, there is a brouhaha about whether exams have gotten easier and whether grades are being inflated, often accompanied by a statement from some minister for Education about possible reforms - exam grades are political, if only in the sense that "something must be done".

Exam grades provide a data exchange between the education system and industry - an along-process exchange from gaining an education to deploying it. And an exam is, in essence, a classification system, dividing the candidates into the classes "pass" and "fail", or possibly into finer divisions such as "A" to "F" or "1" to "9". Of interest here is the way exams expose some of the more complex mechanisms of classification in a way that the semantics of an ordinary word does not.

The eventual aim of education is for the person educated to be able to autonomously delve into a subject, extend their knowledge and critically assess what they have learned - say, to be able to pick up a book on a subject, identify the facts and methods it presents and be able to use them and adapt them as circumstances change. For example, when a new version of the Highway Code is promulgated, a driver should not crash when they navigate new junction layouts, encounter new situations such as a self-driving vehicle, and adjust their driving follow new regulations.

The exam is set to randomly sample a particular subject knitwork, on the assumption that with enough samples (questions answered), the combined score will be a valid indicator of the candidate's knowledge - the candidates must recall particular facts, apply relevant techniques of reasoning, and apply meta-knowledge about how reliable are the facts and techniques used. All exams are norm referenced within their particular context. That is, each grade is based on the cumulative performance of many candidates. and it differentiates that range from larger or smaller ranges of knowledge. However, in practice, its main role comes in practical knitworks such as selecting candidates for employment. The existence of such hiring practices then create secondary, practical knitworks, where future candidates are motivated to revise or to hire additional tutors and where teachers are driven to narrow what they teach to just the examinable syllabus or focus on techniques designed to give the candidates higher marks from less knowledge.

What do exams measure? Firstly, an exam grade is a proxy for the ability of the candidates within a narrowly defined remit - a Maths degree, an English GCSE, or the ability to drive a car. A proxy because the measurement itself is unreliable - it suffers from an number of confounding factors, including the health of the candidate on the day, the competence of examiners to mark the exam, the competence of the teachers to cover the syllabus, the luck that on the day enough questions will match the particular candidate's knowledge, and so on. Secondly, in school exams, the overall exam grades of all pupils are used as proxies to assess the effectiveness of the school, Consequently, the system is gamed by schools so that the individual exam grade is only a partly a measure of the candidate's knowledge - it is also a measure of how well the candidates are trained to give the school a high effectiveness rating.

As is well known to industrial metrics experts, any metric that is used to reward performance will lead to behaviours that maximise the scores against that metric. Or, there is no way to avoid grade inflation except by investing more money in the design of syllabuses and exams and the full resourcing of marking. And to provide not only a regulator that examines the exam boards, but also a meta-regulator that examines the way that metrics based on exams are distorted because of the way they are used, including the way they are used to reward (or punish) exam boards and their

regulator.

"Something must be done". But other than putting up taxes to put more resources into the system (which will also cause "good" grade inflation), doing "something" will achieve nothing - the problem is the slogan.

11.5 Emergency Response - Alternative Classification Processes

Generally, a classification is based on factors salient to the process/system that is going to act on that classification. A problem arises when multiple actors are involved in the same situation - say, when fire, police and ambulance are all called to the same emergency. Each organization will classify the incident in a different way, to match the response they need to make. The focus in this example is "incident size" - Emergency Responders need to know "how big" an emergency is so they can send enough resources to deal with it. For example, the fire service will need to send more fire engines to a thatched cottage than to an ordinary house fire, although in either situation, the electricity supplier will send the same engineer to make safe the power supply.

This particular issue arose in the development of the Tactical Situation Object (TSO), a basic message designed to provide situation awareness to multiple emergency responders for large scale incidents stretching across European borders. The background is an EU supported project "OASIS", [11.4] not to be confused with OASIS (or OASIS-open), the industrial consortium that publishes standardised XML vocabularies [11.5]. OASIS the project included various European vendors of situation awareness systems plus representatives from various emergency responders from different EU countries, including the fire and ambulance services. The project found that cultural differences added a further layer of complication. For example, the French Sapeur Pompier not only provide the fire service, but also emergency ambulances, while in the UK fire and ambulance services are separate providers. In one meeting, a Swede from the fire service observed that simply having the discussion in English lead them to cast their requirements in a different way to the way they would in Sweden.

As designed, the TSO described the situation in terms of a number of different factors, including the type of incident (e.g. fire, collision, chemical spill), where it took place (urban, suburban, countryside) and so on. It was realised that the TSO would contain enough detail for the different responders to calculate the scale of the incident for themselves - a thatched cottage fire is a major incident for the fire service, but a standard one for many other responders. That is, "this is a big incident" is an inference from a number of different subject knitworks and the supporting differential knitworks. Moreover, this inference is designed to lead to the right practical knitworks of the form "what do I do when I get there?"

The TSO is an illustration of "no one ontology" - that is, the way people, organizations or cultures deal with the world is specific to the needs of that group, and imposing a single system of terms will not meet their requirements - even if it is easier to write the software in that way. The different systems of colour terms (previous chapter) illustrates the same point.

11.6 Archiving - From Classification to Definition

Subject knitworks often come with a variety of baggage - inferences and expectations not strictly entailed by their definition. Cats are not just a small mammal, but a much loved (or disliked) pet, or, in ancient Egypt, something to be worshipped. A classic open-top Bentley is a magnificent machine, a car to roar down the highway, or something to drive with caution as it has 1930s brakes. And an archive smells of a musty room where old documents are forgotten.

In developing standards, occasionally the baggage prompts scope creep, and one has to be more explicit that usual about when one means. Long term data archiving provides an example, with terms like back-up, retention and archiving frequently being movable feasts. Computers and software - especially software - will be replaced or updated many times over the 70 year lifecycle of an aircraft. Consequently, a long term archive for an aircraft design cannot simply be a dusty hanger full of old magnetic tapes - there may be nothing left to read them, the software doesn't run on modern computers, and the old guys who remember how to use the software will be long retired. Consequently, the term "Archiving" in the project LOng Term ARchiving (LOTAR - see ch 7) needed some work.

The LOTAR project developed a taxonomy of four factors that defined the problem: Archiving Objectives, Length of Time, Stored Form and Invariance [11.6]. For example, Stored Form was subcategorized into Detail, Representation and Format, with each subcategory being further subcategorised by specific terms. The scope of long term archiving could then be tied down by reference to the detailed subcategories. For example, the LOTAR scope is focussed on *StoredForm.Detail.Accurate {1}* and *StoredForm.Representation. DerivedRepresentation {2}* where {1} = geometric solids such as cones or spheres, as opposed to approximate surfaces faceted to a mesh of triangles and {2} = a standards- based format, as opposed to a Native Representation or a Presentation (a picture).

The implications of these choices are significant. For example, when archiving CAD, choosing a Derived Representation rather than a Presentation effectively put the focus on storing the 3D model in a international standard for CAD (ISO 10303-242) rather than as the drawings or images derived from the 3D model.

In terms of knitworks, the LOTAR project set out a clear differential system of terms, and provided supporting subject descriptions. The standard developers could then get on with defining each practical knitwork (for archiving a type of design file) without the need to answer queries about "why X is not included". That is, the project defined its terms precisely, and used these definition even though the words used for the terms - such as "archive" - are rather more nebulous. This slightly pedantic approach probably saved hours of discussion as project personnel came and went, and helped prevent the sort of scope creep that such personnel changes or lobbying from vendors can engender.

11.7 Class/Classification Duality

Every data exchange starts with a business process that has been mapped from the real world to a taxonomy of entities and from there to their reification in the computer - the data model. After a data exchange, a second business process - or the next stage in the same process - maps back from that taxonomy to actions in the real world. The people involved must pick the correct entity to

correspond to the thing in the real world that is the subject of the exchange. If not, the two ends of the exchange use the same term to talk about different things or different terms for the same thing - confusion and irate phone calls follow (been there). The job of semantics is to make that mapping explicit and certain and thereby avoid such mismatches.

On the computing side, we start with an information model - a taxonomy of the things that the information system knows about. The information mapping classifies the things in the real world against that taxonomy. That is, the taxonomy should come with a decision process - a classification algorithm - to assign things to the taxonomy's classes. That decision process will take observables and compare them to the attributes for each class - although the fundamental attribute of an apple - that it tastes like an apple - is not observable until it is eaten, Ludwig's system must sort fruit on the basis of observables which do not damage the product. An apple has a shape (round), size (typically 50-60 mm across), colour (red, yellow or green but not brown). A pear has a distinctive shape, as does a banana, while an orange is bright orange. We can assume that Ludwig himself will sort the fruit into the right "fruit type" boxes, leaving the fruit machine to check only the colour. If Zane the robot were to take over while Ludwig is on holiday, then he would have to dig into the software documentation to pick out what attributes to observe when loading the boxes.

The taxonomy and the classification process are duals [11.7] of each other. The taxonomy is a set of classes distinguished by their observable attributes. The classification process uses observables to sort things into the classes of the taxonomy. In terms of knitworks, the differential knitwork (in a given context) will identify the classes of interest, and provide the inference - the decision algorithm - that sorts things into those classes. The subject knitwork will identify each class and the attributes it has, including those that are used in the decision process. However, the decision process may use combinations of attributes - apples are round but not bright orange - implying that the classification cannot be made using only the subject knitworks. Another example

- a van has a single row of seats and a large storage space
- a bubble car has a single row of seats but minimal storage space
- a hatchback has two rows of seats or large storage space
- an Estate car has two rows of seats and a large storage space

Here, no single attribute distinguishes a van from a car. One cannot simply declare a taxonomy without ensuring that there is an equivalent decision process. Zane the robot got it wrong when dealing with a box of wax display fruit, as taste was not an attribute in the data model, nor was there a category *WaxFruit* to check against.

One might also note the political misuse of the taxonomy/classification duality. For example, an immigrant has status in {*legal, undecided, illegal*}, but in immigration rhetoric this can be misused by implying that anyone that does not have the status *legal* therefore has the status *illegal,* which is invalid because the immigration decision process may not yet be completed - their status is *undecided.*

11.8 Semantics/Processor Duality

Not only is there is a duality between the set of classes identified in the differential knitwork and the classification process, but there is also a duality between meaning and the practical knitwork which exhibits the meaning. Here we are returning to OWL, the Web Ontology Language, and will look at three symbol processors and the inferences they make - three practical knitworks that ground the semantics of an OWL model. This is, of course, a discussion on three levels, but level change is not

the focus here.

The lowest level processor is the character stream tokeniser. This takes a text file containing an OWL model, and processes it character-by-character to output a stream of tokens: RDF predicates, OWL vocabulary terms, and so on. The human equivalent is recognising the words in a text message, but before we think about what each word means. The meaning of the character sequence "C", "L", "A", "S" and "S" is that this pattern of letters is an OWL vocabulary item *Class.* But *w*hat the term *Class* might mean is not in the scope of the tokeniser, but of the OWL reasoner.

At next level is the OWL reasoner itself. This operates on the RDF triples using the OWL vocabulary according to the semantics of the OWL language. For example, if a *Class X* is defined as having a property with the name *Size*, then an occurrence the vocabulary word *Subclass* (in natural language, *Y is a subclass of X)* can be used to infer that the subclass *Y* also has the property *Size.* The OWL reasoner assigns no meaning to the term *Size* - it is merely a pattern of characters making up a property identifier. However, it does apply a meaning to *Subclass*, and uses it to infer - assigns the semantics - that a *Subclass* has the same set of properties as has the *Class*. *Subclass* is not merely a pattern of characters to the OWL reasoner, it has a meaning that the reasoner can interpret. And it is the semantics of the OWL vocabulary that give the Semantic Web its name, not the model identifiers - such as *pizza* - used in the OWL file.

The third level is the system (or process) in which the OWL model is used. This is not part of OWL itself and OWL assigns no meaning to this model. The meaning for identifiers comes from the mapping of those identifiers to elements in the process - from the information they convey. If you are running a pizza delivery service, then one of the properties of the *Class pizza* is its *size* - choice of *small*, *medium* or *large*. If the customer orders a *large pizza*, then the customer should get a large pizza - the meaning of "large" here lies the behaviour of the cooks who make the pizza. If your partner sets up a cake business, they could use your OWL model unchanged, but interpret the class *pizza* as referring to a *cake*, and deliver a large cake if that's what the customer orders. Note that the size of a large cake is unrelated to the size of a large pizza - the meaning of large is defined by the business process, not the model vocabulary. Of course, the user interface for the cake business would have to print "cake" every time it references the class *pizza*, but that will cause no confusion, except possibly to the programmer employed to update the software.

That is, we need to separate out three levels of inference:

- the symbol level, turning a symbol stream into data into OWL vocab - "class pizza" consists of two sets of five letters to be treated as two vocabulary elements of the OWL language;
- the OWL language converting OWL vocabulary into description logic predicates - "pizza" becomes the name of an OWL class;
- the business process, where the chef makes the pizza ordered.

(And there is a level change when moving from one level to the next.)

That is, there is a duality between the semantics of a term and the semantic processor that operates on the term. Or equivalently, semantics is a relationship between a set of terms and the practical knitwork which is exhibited in the behaviour of the system. Cat hears "Walkies" and yawns, but dog barks, heads excitedly to the door.

11.9 The Semantic Processor - the Fourth Knitwork

The knitwork, as currently conceived, is a multi-layered, conceptual interlinking of facts and inferences, with complex knowledge being built up by connecting many simple facts with simple inferences and chaining many simple inferences together. Moreover, facts come with "metadata", for instance. when and where did the facts come from, how reliable were the sources, have they been verified (and how reliable was that verification). In order to focus on particular areas of knowledge, the idea of focused, sub-knitworks has been quietly slipped in - here the differential, subject and practical knitworks. Whether they are physically separate or simply virtual boundaries within a single, overarching knitwork is yet to be thought through.

Therefore, when I say that the semantic processor swaps between the differential knitwork and various subject knitworks, from the view of a computer implementation, currently this is no more than a change in focus on what is being discussed - virtual boundaries. In current software technology, at one point the software is applying a classification algorithm, at another point it is focused on inferences about a particular subject. The role of the "semantic processor" is, in effect, performed by the programmer, who ensures the system performs the functions at the right point in the process/algorithm.

A more general concept of a second level of knitwork can be illustrated via a "product explorer" - exemplified in a maintenance system. This system works across multiple applications, accessing many product structures, manufacturing records, and integrated vehicle health management in order to build up a picture of a maintenance issue. Example: a ship's fuel tank is identified in the systems product structure; the subsystem "tank" is not part of the physical product structure, as the walls are built independently in different sections of the ship, but it "appears" as different sections are welded together. The network address of the corrosion sensors are recorded during the build process and linked to the wall each is installed in. Finally the IVHM system records - over many years - the history of corrosion. The question "What is the state of tank 123" is answered by chaining the response from one database to the query for the next - the system "tank" to the physical tank walls to the corrosion sensors to the actual readings. This would include mashing together the data source descriptions of each application to find a database query (across multiple applications) that would provide the data for the next step in the task. Here the first level of the knitwork is the data, and the second the database schemas of each application - the second level knitwork searches across the schemas to build the queries needed and mashes the data together [11.8]. And a third level is knowledge of the systems available and how to query them.

Or imagine sending a PhD student off to do a literature review in order to find the papers needed for the next stage of their research, including the papers referenced by the papers found. Facts used intelligently to uncover new facts. Currently the "semantic processor" is a concept needed to make this model work, but open the box and you will find only a note saying "something magical happens here". The "product explorer" gives a first hint at how the magic might work.

On a personal note, to become a writer of standards, particularly of the definitions section, I had to upgrade my own fourth knitwork so that I could explicitly invoke techniques that focussed on the meaning of a term, differentiate it from related terms, and which took account of the various ways in which the term was used. One of the places I used it was helping to tie down the scope of the LOTAR project, another was in understanding that the TSO could not create a definition of "incident size" independently of the practical knitworks of particular responders.

11.10 Level Change Between Semantic Knitworks

A subject knitwork for a particular fruit will describe the features of that individual type of fruit, give recipes for using it, and possible other uses, such as the use of a lemon to make a battery. The differential knitwork, as well as providing the classification algorithm, will describe common features of fruit, such as the presence of seeds, what to do with spoiled fruit (compost bin) or even tax rules applicable (0% VAT). Although the meaning of "apple" is a reference to a particular type of fruit, this meaning is always located within a series of contexts, some associated with fruit in general, some linked with apples in particular, and even some specific to the variety of apple.

Example context - jam making: Jam making recipes follow a similar formula for all fruit - fruit, water, sugar, boil until it sets [11.9]. But there is a difference between a low pectin fruit (e.g. blackberries) and high pectin (e.g. gooseberries). That is, we specialise a higher level concept (jam recipe) according to a lower level subject knitwork (the pectin content of each particular type of fruit) - the practical knowledge invoked by "I've bought some sugar to make {x} jam" might bring a different "you need to add..." depending on the pectin content of {x}.

This might sound odd - what has jam making to do with what it means to be an apple? But that this is looking at "meaning" from the perspective of language, which puts the focus on individual words, each of which is explained in isolation. Look through the other end of the microscope - language is a technique for linearising a knitwork, so although our starting point is a single spot of knowledge, it is connected to many different contexts and connotations. The word "apple" is a condensation point from which we tease out all the things that the "apple" means to us - all the things we do with apples. A wax display apple does not taste of apple, cannot be eaten, does not go rotten, and, indeed, would not get mentioned in any discussion of apples except as a counterexample. If we focus our knitworks down to what any specific (computer) system might need to know, then when our robo-chef talks to Ludwig's fruit machine, we are merely relying on a concurrence of labels? And should Ludwig accept liability for gumming up the robo-chef if it did not specify "fruit apple" rather than "display apple" - the latter concept alien to a robo-chef?

Level change becomes more obvious if we look at the practical knitworks involved. Compare "the grabber picked up an item from a box associated to a fruit selection button and dropped it in the delivery chute" and "the machine gave Alan the green apple he asked for". The story of the first sentence is about the mechanics of a machine. The second storey is about an "intelligent" machine understanding what Alan wanted - indistinguishable from a human shop attendant (other than being a square box with buttons). Here the mechanical level is used to explain how a machine can respond at the same level as if it were human.

Actually, this is a very poor story of level change, but primarily because the machine embodies such a limited knitwork. For a more extensive knitwork, we might look at the level change between the neurons in Ludwig's head and the action of Ludwig when serving a customer. That is, we might wish to explain Ludwig's behaviour in terms of his brain. However, no neuron can taste an apple (an integration of several types of taste and smell sensors), nor even tell if it is green (function of the cone cells in the eye). But we can explain seeing green or tasting an apple in terms **systems** of neurons and sensor cells - subsystems of the body. And if we understood the subsystem architecture, we might work out the mechanisms of a higher level subsystem "identifying an apple", and thence its contribution to "knowing about apples". Reminder - level change is about how humans think; I

could not possibly think about the way each of my neurons is working, since I use many neurons to think about just one of them - I would need more neurons than I have neurons. So I need level change to think about brain function in a more "condensed" way. And remember also, level change is about relabelling components of the knitwork - the concept "apple" does not mysteriously appear as we go up from the level of neurons, it is a deliberate relabelling so I need fewer neurons to think about what the brain is doing.

But, to return to the practicalities of exchanging data between computers: a three-four-five triangle is a practical short cut to set out the corner of a rectangle when laying out the foundations for my shed: no need to discuss Pythagoras. Similarly "semantics" - the meaning of a term - is a heuristic, a practical short cut for matching up the knowledge of two agents. Level change is an acknowledgement that need only think about one subset of the knowledge at a time, and provides a convenient heuristic for creating the knowledge groupings. And we think goldfish are short of brainpower!

This re-framing of meaning as being related to a knitwork rather than to language - language being secondary to knowledge - is unsurprising from the viewpoint of communicating with each other. We evolved with a vocal system that could make about 15,000 different sound units (syllables), and to speak at 3 to 7 syllables a second [11.10] - a data rate of less than 100 bits a second. However we can infer a great deal more from context, including our understanding of what the other person knows or is thinking about (theory of the mind). But although "Message 42" is only five syllables, if it points us to reading "War and Peace" that (eventually) becomes a great deal of knowledge shared. That is, the knowledge communicated can be much greater than the data rate of our verbal communication channel. Which is why, of course, I cite references rather than reproduce the thousands of pages of brown banana text that went into the thinking for this book.

11.11 Semantic Drift - An Apology

There should be a section on semantic drift here - the way the meaning of a term can change - change over time, from place to place, or from organisation to organisation. But word count got in the way. Read up on Ring Species [11.11] and herring/lesser black-backed gulls.

11.12 In Summary - A Rose is a Rose is a Rose

Or at least it has been since 1551 [11.12], when Matthias de l'Obel, a member of the sixteenth-century Flemish School of Botany, classified a Dog Rose as a rose.

Semantics, like information, is a mapping from knowledge elements to terms (data). Specifically, in a subject knitwork, a term maps to the nexus of things that are specifically linked to the term, while a differential knitwork lays out all the different terms we should avoid getting our term confused with. And in a practical knitwork, the meaning of a term is exhibited by the behaviour of the system that interprets the term. The compound "three green apples", either when spoken to Ludwig, or as buttons pressed on his fruit machine, will deliver me three green apples (physical knitwork); pear would be the wrong fruit (differential knitwork) but an apple tastes of ... well, apple (subject knitwork)[11.13].

So when in data exchange we talk about the semantics of a message, this is a short cut to saying that there is a reasonably complex and arduous process by which we verify that using a term in a data exchange message will have the same effect in the receiving business process as it would have in the business process that sends the message, If it is used in a peer-to-peer exchange, then I will see the same thing on my computer as you do on your computer. If it is an along-process exchange, then you do the same thing as I would have done if I was running the process myself.

Of course, it is fraught getting from the jelly of natural language to a steel block computer term. Different assumptions as to the context may be made, leading to the same word having different meanings to different organizations. Or politicians are happy to elide similar terms to their own advantage. Which why checking semantics is a complex and arduous process. And in the end, we computing geeks just do the data - its up to you, the users, to use the same words in the same way.

In many ways, this model of semantics not substantially different from what philosophers and linguistics specialists have been saying for millennia. But then they didn't have a machine that would operate on language - language was something unique to humans. This approach brings together three different mechanisms - subjects, differentia and practical knitworks - and links them into a single semantic model - a system to create an artificial intelligence. However, this can only be a partial model - the meaning of language is intimately founded on our experience of being human. The answer to "describe the taste of this apple" will expose limitations of such a model.

Notes and References

11.1 At the time, the main contenders were IBM (a US company) and ICL (then still a British company - later bought by Fujitsu), both of which had some 40% of the UK market. Both claimed they were good British companies, for example, by investing in R&D in Britain. I spoke to an MP after the debate who said that the British economy relied on exports, and for the same UK market share, ICL exported ten times as much as the UK arm of IBM - ICL got the contract.

11.2 Based on Department for Business, Innovation and Skills "Government Security Classifications: HANDLING INSTRUCTIONS and GUIDANCE for BIS staff" V2.1.1 Feb 2014, however, different search terms give different results, and the GOV.UK site suggests that it found over 800 pages of results for "security classification".

11.3 As any hacker knows, the Von Neumann model, in which the same memory is used for data and software, is inherently insecure. However, when the model was developed in 1945, memory was measured in bytes and was very expensive.

11.4 As cited in Fedra Henriques, Delfim Rego "OASIS Tactical Situation Object: a route to interoperability" SIGDOC '08: Proceedings of the 26th annual ACM international conference on Design of communication Pages 269 - 270 https://doi.org/10.1145/1456536.1456593 (accessed 16/06/25). The original papers (now 404's) were:

- Published: 22 September 2008 Overall presentation of OASIS, 2006, OASIS Consortium, http://www.oasis-fp6.org/documents/Oasis.pdf
- Definition of the OASIS Tactical Situation Object, 2006, OASIS Consortium, http://www.oasis-fp6.org/documents/ Ref. OASIS_TA21_DDD_041_DSF
- Data dictionary of the OASIS Tactical Situation Object, 2006, OASIS Consortium, http://www.oasis-fp6.org/documents/Ref.OASIS_TA21_DDD_041_DSF

My comments are based on the meetings of the CEN Workshop agreement on the TSO, for which I

was vice-chair, and which I believe became the basis for ISO/TR 22351:2015(en) "Societal security — Emergency management — Message structure for exchange of information" https://www.iso.org/obp/ui/#iso:std:iso:tr:22351:ed-1:v1:en Accessed 16/06/2025

11.5 For example, see OASIS-open "Emergency Data Exchange Language (EDXL) Hospital AVailability Exchange (HAVE) Version 2.0 " March 2019, Accessed 17/06/25

11.6 See the references in Chapter 7 - the scope definition is in LOTAR Part 3 "Fundamentals and Concepts".

11.7 Duality theory is a branch of maths, for example, in a dual space, the duals of points are lines and lines become points. Proving a theorem about points and lines in ordinary space also proves the dual theorem about lines and points in the dual space. Cf Wikipedia "Duality (Protective Geometry" https://en.wikipedia.org/wiki/Duality_(projective_geometry) Accesses 16/7/20205 (Brown Banana) The only problem is defining what is meant by "theorem".

11.8 The technique of finding chains of data is based on the Halevi algorithm, but modified to use a subset of OWL as the data descriptions. This means that queries also process the sub/supertype chains. However, the product explorer needs to be more intelligent than that, as it must also work its way up and down BOMs and must deal with the different ways that BOMs are represented in Product Design and IVHM and also the way the CIMOSA-CBM standard is "double modelled" by defining abstract data structures instantiated to entity models via reference data (too much information). Thanks to Ian Horrocks and Boris Motik for flagging up the Halevi algorithm and the subset OWL needed for the algorithm. (brown banana)

11.9 Ministry of Agriculture, Fisheries and Food "Home Preservation of Fruit and Vegetables", HMSO London 1927. My copy, 13th Edition 1971 (4th Impression 1979) pp15-30 (yellow banana) Last published 1989.

11.10 An unreliable estimate. The search "how many many syllables are there" gets an figure of 15,831 separate syllable sounds (https://www.quora.com/How-many-syllables-exist-in-the-American-English-language#:~:text=I%27m%20going%20to%20give,syllables%20grl%20or%20grr%2Dell.) or 14 bits of information per syllable. "How fast do people speak syllables" gets a general answer or 4 to 5 syllables per second for English. Hence 70 bits/second - say 100 bits/second to give an order of magnitude. Note that this is much less than the bandwidth needed to transmit the sound of speech, where Voice Over Internet needs something of the order 100k bits/second.

11.11 See, for example, Wikipedia "Ring Species" https://en.wikipedia.org/wiki/Ring_species accessed 17/06/2025

11.12 Wikipedia "Rosa canina" https://en.wikipedia.org/wiki/Rosa_canina, accessed 29/04/25

11.13 What does an apple taste like? AI Response "Apples generally have a sweet and tart flavour with a crisp, juicy texture. The specific taste can vary depending on the apple variety, with some being sweeter, some more tart, and others with subtle notes of spice or floral flavours." No wonder Zane the Robot refuses to use AI. As posed to Google 16/06/2025