APPLIED SEMIOTICS

R STAMPER

Rapporteur: Andrew Blyth and John Dobson

APPLIED SEMIOTICS

Ronald Stamper
School of Management Studies
University of Twente
POBox 217, 7500 AE Enschede
The Netherlands

So far we have introduced the theory of signs as a means of bringing some order to the plethora of information-related concepts. Now we turn to the practical application of this socially-oriented view of information systems.

Semiotics exposes the inadequacy of a purely technical approach to the subject. But it also furnishes us with practical tools for building computer-based systems that are more complete, more reliable, easier to maintain and of higher quality from a business point-of-view, and to do so more quickly.

Sone years ago most of the systems we built were rather small. They served a culturally homogeneous group of people who shared meanings usually because they operated under the same hierarchy. Today, the growth of inter-organisational system is taking away that cosiness. Systems are becoming global, transcending industry boundaries as well as company boundaries. These changes remove the illusion that people and society are merely a side-show for computer science. My argument is that computer science has nothing to lose by adding the social dimension, indeed there is much to be gained, even in the most practical ways. Indeed, we can improve our system specification formalisms, our software development tools and our methods of problem-solving.

I shall illustrate this point, firstly by showing that semiotics leads us to improved specification methods, secondly by introducing the Normbase concept and thirdly by showing that it leads to new methods of analysis that bring some significant benefits.

A Formal Semantic Dictionary

Every dictionary is constructed with the intention of listing the meanings of words. But they are of limited value in the computer field because they are informal and often circular in their definition chains. If we could have a formal semantic dictionary then the data to be exchanged between computers could be organised in compatible structures without the need for some data-administrator to impose an arbitrary solution. And data exchanged between people, via computers, would have a better chance of being interpreted uniformly so improving our effectiveness when conducting all kinds of practical affairs in manufacturing, trade, health care, government, law and so on.

If we wish to act in a coordinated way, each of us must use each word with the same meaning, that is, in the sense of relating the words to actions in exactly the same way. Semantic analysis is intended to achieve this with the precision that the domain requires. The Ontology Charts which semantic analysis generates provide a basic level of uniform semantic structure. Examples are shown in Figures 9 to 15.

Later, I shall explain how such chart may be implemented directly as a schema for a Semantic Temporal Database or for a Normbase. However, regardless of any implementation, these semantic structures can be used for building a formal semantic dictionary. A resource of this kind would be valuable for developing effective, global information systems which the new telematics makes possible.

To understand how Ontology Charts can supply the materials for a formal semantic dictionary, consider what they express. Each arc in a chart asserts that the existence of an instance of the element on its right can only exist during the existence of a suitable instance of the antecedent element on its left (see any of the illustrations above). Thus each element in a semantic schema is placed in relation to others which constrain it meaning very closely in terms of how it relates to human behaviour. If we take one element and trace back to the root agent (Society, the fount of our commonsense knowledge) we select the lattice structure of elements and dependencies called its 'stem'. The stem always contains the agent or agents responsible for the behaviour named by the element. For an instance of an element to exist, appropriate instances of all the elements in its stem must exist. Remember that the element is known to us only as a repertoire of behaviour which allows the agent involved to act in certain way either alone or in conjunction with other agents.

For example, Figure 9 illustrates many points about the dictionary. A *conference* can only exist during the existence of the *learned society*. In the dictionary, we give the antecedent of *conference* as a *legal person*, not necessarily a *learned society*. The reason for this is that, someone or some organisation must be able to take contractual responsibility for the conference arrangements. The *legal person* will only exist by virtue of the *nation* or *nation-state* which provides the legal framework for the contracts involved. Many people would expect the term *conference* to be used as *meeting* is use in the figure, but that is only the brief event when people assemble. Note that a *meeting* is not an agent from a legal point of view although it might informally be regarded as one. The dictionary would give this reading and point to *conference organisation* as alternative for *conference*. The difference is that the gathering of people constituting the meeting would not be competent to enter into contracts. A person will be registered with the *conference*, in a contractual relationship, but *participate* in the *meeting* as a physical activity. We have already noted one semantic defect in the chart: the placing of *conference* as the antecedent of *work* is wrong because we have established in our society, enshrined in copyright law, the idea of a literary work as a legal entity defined within a national jurisdiction, so the work does not cease to exist just because a conference

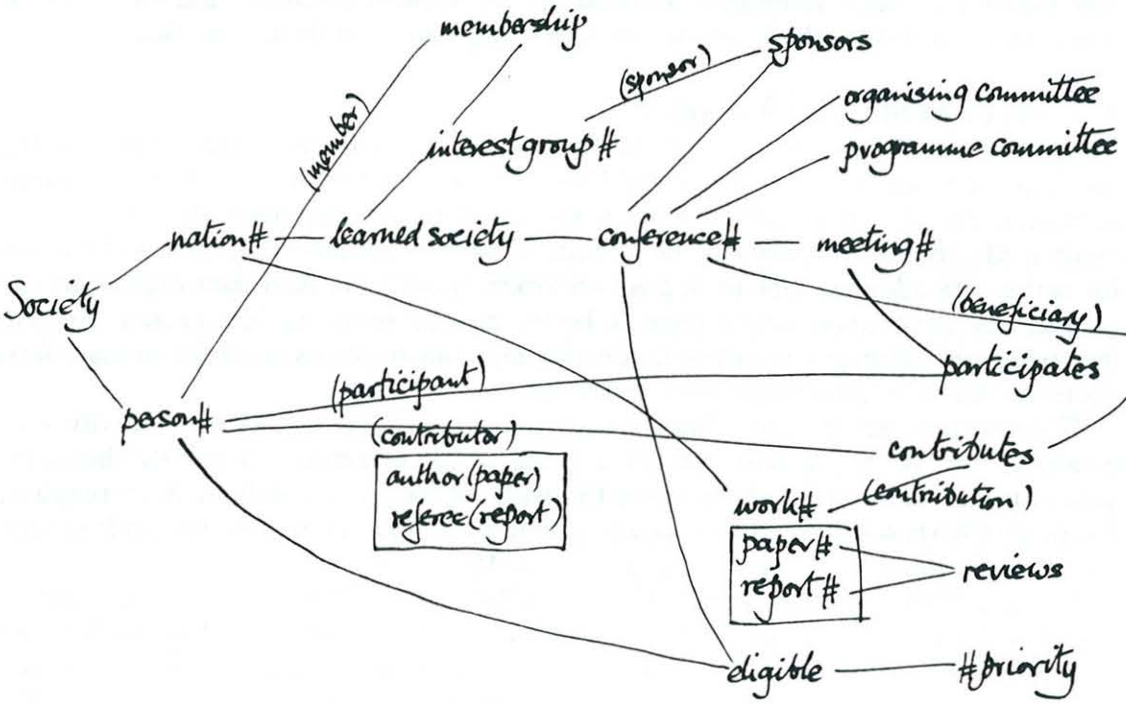

Figure 16: A revised version of the chart for organising conferences

is wound up. Making this adjustment would lead to confusion about *contributes*, the stem of which, in Figure 9 makes it clear that the person (*contributor*) intends the work for the *conference*. In the revised Figure 16, we have *nation* as the antecedent of *work* and *contributes* becomes an illocutionary act (an intentional act performed with a sign, in this case the work) which must be addressed to another agent. Figure 14 on pragmatic aspects of semiotics has already introduced this idea. In the dictionary, *contribute* would appear as one of the 200 or so illocutions, which all exhibit some generic semantic features but differ in detail (See Searle and Vanderveken 1985).

Thus far, it is clear that every element in the Ontology Chart must have associated with it the following attributes: an identifier, the one or two antecedents and the times bounding it period of existence. These provide the basic but limited definition of meaning. One can imagine that almost all organisations involved in running conferences would be content with this model. However, each nation would have different laws for incorporating (starting) legal persons and for striking them off the register (finishing their existence), each learned society would determine the existence of its interest groups and of its membership in its own way, the ad hoc organisation for the conference would have its own, local norms governing eligibility, priority, participation, the constitution of its committees and so on. Thus the semantic dictionary could also provide some guide-lines on the authorities governing starts and finishes. To do so might be going into too much detail for general purposes but a specific problem domain would require this level of analysis. A general purpose dictionary would provide only the ontological dependency structures.

A Canonical Semantic Form?

Ideally, the elements in any Ontology Chart should be re-usable structures for analysis and design purposes. Can this be achieved? Is it desirable?

There is the problem of different languages. In formal semantic dictionary language should not matter. The semantic structures it presents represent behavioral repertoires - the natural language labels attached to them are incidental. Whatever language people speak, the same behaviour pattern associated with a word will have the same entry in the semantic dictionary. If what we call in English 'earth' is distinguished, pointed to, held, manipulated, made use of etc in the same way by people calling it variously 'aard' in Dutch, 'Erde' in German, 'terra' in Italian, 'tierra' in Spanish, then, regardless of language, they share the same concept. This behavioral invariance at the root of meaning may be expressed by an arbitrary 'surrogate code' which can be translated into any language. When we implement a semantic schema in a Normbase, the whole structure consists of these surrogates, so that it is intrinsically language-neutral, stripped of all words - it is a surrogate for realised behavioral affordances. Words are supplied by a layer of language dictionaries that translate to and from surrogates.

Meanings do vary, of course, from culture to culture, even within the same linguistic group but these variations can be expressed by indicating the criterial governing the start and the finish of an instance. It is important, for example, to log all the changes in meanings that take place in a legal context. Legislation, on close inspection, is very largely consists of rules giving precise meanings to words by defining when things start and finish their existence. For example, one finds many exact, different, explicit meanings for even such common ideas as 'child' in legislation (there were at least seven different meanings used in the old UK Family Allowances Act, each with a different rule for determining the start and finish time). We have a rather similar dependence on precise meanings when developing computer systems in business, the problem is that we have poor methods for defining precise meanings except by making them implicit in computer programs where the knowledge of the semantics is lost among knowledge of how to make best use of the particular computer. All these variations taking account of different periods of existence can be handled by extending the semantic dictionary for an application. Hence, to the canonical form limited to the ontological structure, we can add norms (also independent of any information technology) that tend to be a relatively variable as we pass from organisation to organisation and from culture to culture.

You may ask: "What is new about all this - are not the familiar Entity-Relationship or NIAM data models also the basis for semantic dictionaries?" The answer lies in the fact that, though superficially alike, semantic analysis is quite different from data modelling. This fact is made evident by the practitioners using such Classical methods who seem to think that a canonical model for data is neither possible nor desirable. The idea of aiming for a 'semantic normal form' (Stamper 197?? and 198??) is roundly criticised by Veryard (1992, p.167) saying "Uniform modelling could remove the creativity and serendipity of this activity, and replace it with a bureaucratic sameness." He goes on to illustrate and applaud the arbitrary choice which classical methods permit by showing that "parenthood can be modelled in three ways:" by the entity 'person' or an attribute of a person, or as a relationship between two persons. This arbitrariness more or less guarantees that computer systems will not be compatible unless some data-administrator imposes uniformity. Veryard (p,169) prefers not to "mask the personal style and flair of the analyst". He succeeds in eloquently confirming my view that classical data modelling deals with the syntactics of character-strings within the computer (where arbitrary solutions are acceptable) rather than with the semantics of the language used by a group of people (where precise agreement is essential). When people try to coordinate their actions by communicating with one another they are unlikely to succeed unless they use words with sufficient agreement about their meanings. A semantic model should express as fully as possible these meanings that are shared by the community of users in order to be as sure as practically possible that they (and their computers if using EDI, for example) understand one another. The "creative element" and the "analytical flair" may be fun for the analyst but they are obstacles to understanding within the user community and a barrier between computer systems.

Currently we are developing a semantic dictionary containing the kinds of information illustrated in the previous subsection. Using this dictionary, it is possible to generate draft semantic schemas automatically, ready for critical scrutiny by the analysts. Every time a dictionary element is used and subjected to close examination in context, there is a chance that errors will be found. These findings if fed back, could be used to improve the dictionary. So, in principle, we could accumulate in the semantic dictionary knowledge about meanings which would be global in its validity. This would make it possible, in the long-run to achieve consistency at the semantic level among independently developed systems through the self-discipline of logic rather than as a result of administrative control. Meanwhile, it will be easier to establish agreed standards for EDI and Product Data Interchange, PDI. Without such tools, the progress of global data-interchange will be delayed.

The Analysis and Specification of Social Norms

Earlier, I suggested that we can usefully think of 'information' as a process of 'giving form' or 'informing' in the original sense, as in "The potter informs the clay on his wheel". But what is being formed by the use of signs? The answer is: whatever constitutes our social reality.

If the physics, empirics and semantics of a sign are in order, we have an *utterance*. For example, you may own a parrot which can say, "He's a dirty old man." the parrot produces an utterance with, as far as the parrot is concerned, no meaning and no intention behind it. There will be no social consequence. However, if you follow the parrot's words by saying, "And he's quite right, you know." you will have used his utterance to communicate *your* meaning and *your* intention with potentially explosive social consequences. Computer science, as normally practised, is concerned solely with the processing of utterances.

I am advocating an extension to computer science or 'informatics', to include the making and changing of social reality through the use of signs. Signs change the perceptions, value judgements, beliefs and commitments of people, in particular by allowing people to share them as a basis for cooperative, coordinated action. If we are to make this extension to informatics in the tradition of the technical branch of the subject, then we need a formal and rigorous way to represent social reality.

The ontological structures introduced earlier take a step in that direction. They capture the perceptual norms relevant to the domain we are concerned with. What we perceive depends to a large extent on what it is valuable for us to recognise for biological or social reasons, so the ontological structures also express much about our value systems at their most basic level. To extend the ontological structure we need to say something about beliefs and commitments. Cognitive norms, which reach their most refined form in scientific laws, express the links between circumstances and what is to be believed or expected. Behavioral norms, which were introduced earlier, express when an agent (person or group) is permitted / forbidden / obliged to do something. The cognitive and behavioral norms can be added to the ontological structure as authorities for the start or the finish of instances of its elements.

I now want to explain more of Norma, the formalism for representing norms and affordances of which I introduced little more than the concept of ontological dependency in order to introduce semantic analysis.

A Formalism for Norms and Affordances

The story, so far, is that we, human beings, collectively understand reality in terms of the patterns of behaviour that the physical and social environment affords us, given our own nature. Reality comprises our own biological structures, our social structures and the world beyond. Before continuing the story, let us recapitulate the key points about the representation introduced so far. (The numbering is the same as above for the same concept.)

1.	**A**	**John**	**agent-in-his-environment,**
2.	**Ax**	**John stands**	**agent in some temporary invariant state**
3.	**Ax***	**John stands *able***	**agent in the state of having the ability to discriminate that state**

We can refer to the authority responsible for the existence of the invariant by the expression

4.	**Ax@**	**panel of judges**	**responsible agent**

which may be replaced by separate agents judging the start and the finish of the invariant as in the case of a person's sickness for insurance purposes.

4'.	**Ax@+**	**doctor**	**authority for start of sickness**
4".	**Ax@-**	**John himself**	**authority for finish of sickness**

An agent in some invariant state acquires a different repertoire of behaviour, thus:

5.	**Axy**	**John water swim**	**agent in one state attains another**

The start or finish of an invariant coincides with some other invariant, usually taken as the state of a chronometer or calendar plus clock. We can represent these:

6.	**Ax+**	**John stand start**	**another invariant used to represent time**
	Ax-	**John stand finish**	
7.	**A.x.y**	**John has heart has valve**	

The part relation may be useful to represent but it is not intrinsically different from other affordances. This notation may be helpful informally.

There are joint affordances which depend upon the coexistence of invariants:

9.	**A(x,y)z**	**John (cup, saucer) on**
10.	**A(x,y)z**	**Society (person, person) marriage**
	A(x,y)z	**Society (person, nation) citizenship**

where (,) = **while.**

Determiners generalising the concept of measurement are indicated thus:

15.	**Ax# #d**	**Society coat# #size**

where, identity, the commonest determiner, is indicated using the hash sign alone:

11.	**Ax#**	**Society person#**

Where two individuals are involved in a joint affordance, they must be individuated, usually by giving them role names, thus:

14. **Society (person# #husband, person# #wife) marriage**

Generic/specific hierarchies are defined using norms:

16. **A((a:b:c:d:e) → f)** **A((person:apple:bridge:box) → physical object)**
where (:) = **orwhile**. This can be represented in a graphical form either by an arrow from specific to generic, or by a box containing a list of specifics with the generic as a heading.

Now let us look at some more features of this formalism.

Composite Realisations

When an agent is doing one thing, **x**, he might also be doing another, **y**. (These are not the above **x** and the **y** which is ontologically dependent on it, these are ontologically independent, so that one or other, both or neither can be realised at any time.) Doing **x** while doing **y** is symbolised:

17. **A(x,y)** **John (stands *while* speaking)**

This is called a 'restriction' of the affordance **stands** by the affordance **speaking**. Our agent, A, may also find it significant to do **x** or **y** or both indifferently:

18. **A(x:y)** **John (speaks *orwhile* singing)**

This is called a 'conflation' of the affordances. Finally, the agent can recognise the kind of behaviour that maintains the invariant **x**, while **y** is *not* experienced, perhaps, but not necessarily, deliberately avoided. This is symbolised:

19. **A(x;y)** **John (sits *whilenot* singing)**

This is called an 'exclusion' from the affordance **sits** of the affordance **singing**.

These operators correspond roughly to the conjunction, disjunction and negation of classical logic. However, notice also that we are not joining propositions together but behaviour patterns. Actions are more basic than words so we must recognise some important differences. One of the most significant of these is that we cannot have negative actions, so that it would not make sense to have a unary negation operator. A second key difference between this part of NORMA and propositional logic, to which it bears some resemblance, is the role played by time. Whereas we have been accustomed to assuming that propositions are timeless, here we deal only with invariants in the flux of activity which have a finite existence. Thirdly, note that the periods of existence of these combined realisations are functions of the existences of their components, but that these are not truth-functional operators because the notion of truth does not enter into our considerations until we introduce propositions as a special kind of semiological behaviour. At this point we are concerned with the kinds of behaviour patterns and their combinations which can be known to the simplest of organisms, even ones that have no power of language. Here we are concerned only with *direct knowledge* of the world obtained by our actions and NORMA is an attempt to find a way of representing the internal structures of the action patterns that the agents themselves can recognise.

The restriction, **A(x,y)**, as the co-occurrence of two affordances may jointly afford the actions, relationships or experiences that would otherwise be impossible. No more than two antecedents to a joint affordance are necessary, nor permitted. The theoretical argument in favour of this constraint is that, given a triadic or more complex relationship, we can always associate every pair and so construct the complex relationship from binary ones. More important, however, is the empirical test of the constraint: far from causing difficulties of analysis, it forces us to look more closely and accurately at the semantics of the problem.

Semiological Affordances

So far, we have only accounted for the kind of knowledge that an agent unsophisticated as an earthworm might possess about its world of the here-and-now. As each change takes place in its world, such a primitive agent would probably lose all awareness of the previous state of affairs. This limited form of consciousness can be extended if the agent can use signs. But evolving organisms evolved the use of signs from the simple kinds of behaviour introduced so far. They evolve naturally using the mechanisms of metonymy and metaphor. In constructing signs by metonymy we use some part of a situation (available here and now) to characterise and represent the whole which may be in the past, the future or out of sight (eg:

the footprint stands for the earlier presence of the creature with its foot on the soft clay). In the use of metaphor, something present to us can stand for something at a distant time or place because it is similar to it in some respect (eg: a portrait stands for a person). Let us (quite reasonably I think) assume that our powers of language have their historical roots in simple mechanisms of these kinds without enquiring further or attempting a fuller explanation. Let us take it for granted that the kind of agent of interest in the information systems domain does have this capacity to use signs in a sophisticated manner.

NORMA must have a means of representing signs which encompass far more of our human knowledge that the direct knowledge of the here-and-now. We can represent a sign meaning **Ax** by the affordance called "**Ax**". The earlier discussion in this paper on semiotics makes it clear that signs are the product of social norms. Thus a language group is the kind of agent that realises the sign "**Ax**". An agent by uttering or receiving an instance of it, that is by making a sign begins the process of communicating. A full explanation of the representation of signs would require more space than would be justified in this paper, so the treatment here is intentionally very short and informal. As a first approximation we could write:

20. **L(B,"Ax")u** **L(Mary, "John happy") says**

where **B** performs the act of uttering a sign meaning "**Ax**", perhaps that John is happy. **L** is the language group, the collective agent needed for linguistic abilities to exist. "Utter" is more familiarly represented by "says" (NB: not by "says that".)

Signs are of two kinds: tokens and types. If you write the same sentence ten times each instance is a different sign-token but they constitute only a single sign-type. The expression "**Ax**" signifies a sign-type, among its invariants being the meaning of the sign. It does not matter by what means Mary expresses herself - speech, writing or gesture - we are dealing with the same invariant.

A person who can utter signs can use them to perform communication acts. The ontology chart for pragmatic concepts has already introduced the two kinds of communication acts: those we perform when we formulate a message we can send to someone (called illocutionary acts) and those we perform when we act upon a communication (called perlocutionary acts). Any affordance with an utterance as its antecedent will be an illocution. Every illocution (such as 'assert that', 'say that', 'request that', 'suppose that', 'predict that', 'decide that' and so on) presupposes that an utterance is made so we need only express the illocution, **i**, by writing simply **i**:

21. **L(B,"Ax")i** **L(Mary, "John happy") assert**

Evidently we shall have to make much use of these structures when specifying the communication processes in an information system.

We can now introduce the concept of truth. Truth we regard as a complex semiological notion concerned with the relationship between direct knowledge of the world and propositions spoken about the world. When a community reaches a consensus about the correspondence between a proposition and the direct experience of the aspects of the world which the proposition represents, then, as a shorthand, we call the proposition true.

22. **Community ("Ax", Ax) correspond**

is an invariant that holds while "**Ax**" is true *for that community* and that community must take responsibility for what it regards as true. We have discarded the notion that there is anything absolute about truth that seems to be suggested by Tarski's presentation (1935) of a correspondence theory of truth. Of course, the community cannot recognise the simple correspondence of a proposition with the direct experience of the world when the proposition deals with anything but the here-and-now. In these more complicated situations, we can be explicit about the behaviour that can, to some measure or other, justify the community accepting the proposition. How to do that in various circumstances we handle in the law of evidence or the epistemology of science or of historiography and so on. A community can select its corpus of truth on the basis of simple faith. Taking responsibility is the key issue and the importance of a process of justification is its role in persuading individual members of the community to share fully in the consensus and the implied collective responsibility.

Related Affordances

Having learned how to achieve and recognise the behavioral invariant **x**, the agent may learn to recognise two related patterns of behaviour, the beginning and the ending of **x**. They are symbolised thus:

23. **Ax<** **John sleep beginning**
Ax> **John sleep ending**

Notice that < and > are operators providing the names of putatively related affordances. Agents must include these in their perceptions of the world if they are to be able to act purposively to begin or end some state of affairs. Our language, sometimes but not always, has explicit terms for beginnings and endings. A wedding begins a marriage and a divorce can end it. A company begins with its legal formation and ends with its dissolution. A person begins with birth and ends with death. The finish of a successful beginning or ending of **Ax** will be, respectively the start and finish of **Ax**, but a beginning or ending process can proceed without eventually being successful.

The ability to relate certain kinds of behavioral invariants with the beginning and ending of others is the basis of our ability to act purposively or to explain events in terms of cause and effect. The key forms of purposive act, being to begin, to end, to sustain or to prevent some state of affairs **Ax**, are represented respectively by **Ax<, Ax>, Ax>>, A<>**. We can get into deep philosophical water dealing with the issues of purpose and causation, so we shall skate round these problems. Suffice it to point out that we humans can use language to describe animal behaviour in terms of their apparent purposes when they may only have a series of instinctive mechanisms to drive them. Perhaps purpose depends upon our being able to represent our behaviour in symbolic form, if so, the formalism developed so far does not deal with this capacity.

All we can do in the parts of NORMA explained so far is to represent the behaviour of the agent in the here-and-now. The special behaviour that distinguishes mankind from other species is our ability to use one thing to stand for another. Much of our world is semiologically constructed, as noted above. NORMA must be able to represent semiological behavioral invariants.

Time

Until we introduced the notion of time, the formalism was only capable of representing the here-and-now world of the agent. The two basic axioms (1. No knowledge without a knower. 2. No knowledge without the knower acting.) force us to account for all reality, including the past, future, distant or possible worlds, by showing how they are constructed out of parts of the agent's present reality. To know about any reality that is not present, here and now, we are dependent on our ability to use signs, whether the internal neural signs in the brain of the agent or externalised spoken or written signs.

Our capacity to use the notion of time is one essential, basic part of our semiological reality. The classical logical specification languages for information systems lead one to assume that time is a part of an objective reality. NORMA, on the other hand, allows one to describe only the here-and-now, forcing one to recognise that we have access to nothing more. Knowledge of worlds that are not present we achieve by using signs to represent aspects of these remoter worlds.

Time depends upon our ability to use signs. Because we have access to nothing more than the here-and-now, the start and the finish of every invariant state in the flux of our experience are always either in the past or the future. They are known to us only in recollection or in prospect. The start and finish events only exist because we can make signs to represent them. Consequently,

24. **Ax+** **John sleep *start***
Ax- **John sleep *finish***

the start and the finish of the realisation, Ax (John sleep), must be sign-types. From an epistemological point-of-view, sign-types have the marvellous property that they can, but need not, coexist with what they represent. This allows society to bind events over time with no ordinary cause-and-effect chain to provide the link. Mary might establish the truth of an utterance by the correspondence test

25. **B("Ax",Ax)c** **Mary ("John sleep", John sleep) correspond**

but, also, when Ax is no longer the case, the sign continues to exist. Thus the sign is our means of binding together events at different times, without which we would not be able to build our social or business reality. In the physical world, causal chains involve a sequence of here-and-now interactions of material objects. In the social world (eg: tax liabilities today for last year's income) semiological linkages are basic. Time is a key semiological construct. Not only do we use the signs **Ax+** and **Ax-**, we also construct other useful semiological affordances derived from them, for example ***before*** and ***after*** the existence of a realisation:

26. **Ax** **John sleep *before***
Ax/ **John sleep *after***

neither of which is a directly experienced state of affairs, but known only through our use of signs.

Norms, mechanisms and responsibility

We have now assembled a repertoire of expressions that enable us to write the norms of any organised social system that we may wish to specify. The mechanisms relating one affordance to another are specified using the notion of an affordance that ensures that a condition **x** will always give rise to some disposition to act, **y**, which we can express either as **(y *whenever* x)**, or **(y *then* x)**:

27. **$A(x \leftarrow y)$** **John (should stand *whenever* sing)**
$A(y \rightarrow x)$ **John (sing *then* should stand)**

which appear to be like saying that **x** is implied by **y** in a classical logic, but notice some important differences. Norms are not true or false. They can be in some sense valid in their application within a community but they may or may not be obeyed, in other words, the consequence of the norm applied to the factual condition is contingent upon the behaviour of the agent(s) to whom the norm applies. A deontic norm of this kind proposes some intended line of conduct. The use of 'should' in the forms on the right of ?? signals an obligation but norms may also establish a permission or a prohibition, prescribing different patterns of beginning, ending, sustaining and preventing actions.

In some cases the consequence can be categorical as, for example, when the condition defines a state of affairs named by the consequent, such as the start of a copyright, in this case nothing changes, the agent is the community responsible for the norm and it uses the norm to assign a new significance to the complex state of affairs in the condition and, incidentally, assigning a convenient name to it.

Also note that the norm is quite unlike material implication in classical logic. There is no counterpart to the truth-functionality of an implication because the existence of a social norm is not related to the existences of its components. The relationship between standing and singing may just be a temporary norm, introduced by a conductor for a particular concert. Moreover the *modus ponens* rule (x, x implies y, therefore y) has only a weak counterpart in that a norm can yield only an expectation that the responsible agent will act as prescribed. There is yet another precaution we must take before applying a norm. Whereas in classical logic a truth is, independently of anything else, a truth, and every postulate or fact is potentially relevant to every other one, in NORMA the facts are the responsibility of an agent and there is no automatic assumption that every norm can always be applied to every fact. In the use of social norms, we use our judgement before applying them in particular cases, legal norms, for example, are selected for their relevance by the judge before whom the case is considered. The last thing we want to do in the real world of practical affairs is to treat laws with the mechanical certainty of an unconstrained deductive system, on the assumption of a closed world governed by predicate logic. We must have a place for the agents who take responsibility for formulating the facts about an infinitely open reality and deciding which norms are relevant to them.

One may think of a norm as existing as a kind of internal mechanism within the members of the community to which it applies - such as a nation-state, a company or a learned society. Written laws and rules are signs representing norms: **"A(x → y)"** rather than **A(x → y)**. The development of NORMA is motivated aesthetically by the desire to provide a way of representing social norms (exemplified in all their rich complexity by legal norms) simply and clearly, taking the forms evolved in natural language as the guide.

I think that mathematics is a misleading source of inspiration for a logic to support the modelling of social behaviour of the kind that we find in organisations (= information systems). I hope that the above discussion of the roles played by time and responsibility throughout the representation of facts, and the making and applying norms suggests clearly the distinction of our domain from that of mathematical reality where time and responsibility have no necessary place at all.

In a mathematical model one produces a system that is free-standing in the sense that you supply the appropriate parameters and apply the formulas to them to generate a model of the processes being studied. In a mathematical model there is nothing within it that links it to the world outside it, but a specification in NORMA is quite different in assuming that some *actual, responsible agent* must determine every start and end event unless a norm is used to share the responsibility over a group of agents with different relevant knowledge. A system of this kind can be automated (as will be explained below) but it cannot run autonomously, it must continually exchange messages with the agents who belong in the social system. This is important because the specified agents alone are the arbiters of meaning, they belong in the informal social system and provide the knowledge of those informal norms that cannot be expressed in symbols. The formal system can only be an over-simplified version of any real social system and NORMA makes the link explicit. One can turn the NORMA model into a mathematical model by providing mechanical surrogates for the agents, for example, by generating decisions using appropriate probability functions for their choices. However, if we are interested in designing information systems, then we shall use computers to handle the parts which can be formalised but we shall need to embed these automated components into the informal social system. A specification in NORMA, by requiring the responsible agents to be identified, forces one to be explicit about how that embedding should be done.

Software Built on Semiological Principles

The specification formalism, NORMA, introduced above, allows us to describe organised social behaviour with whatever degree of precision is appropriate to a problem. More importantly, it enables us to concentrate on the social issues without having to introduce any technical issue. The social or business requirements are the basic invariants (boundary conditions) which every technical solution must satisfy. Norma provides the counterpart to a data model in the form of a semantic schema and the counterpart of a model of system functionality in the form of the norms. It appears that we have simply a very high-level programming language.

Two systems have been built to 'implement' this formalism. One, the Semantic Temporal Database, handles the semantic schema and provides modules for the functions. The other, the Normbase, interprets the norms. The concept of a Normbase takes us in the same direction as the Database concept but a great distance beyond.

An interesting detail to note is that these implementations are, incidentally, CASE tool generators. If you think of a system which can store and manipulate data about any of the aspects of systems shown in Figures 10 to 15, then we have a software tool to support systems analysis at each of the six levels of the semiotic framework. In other words, NORMA is good meta-modelling language in which the Classical methods and models can be expressed.

The Semantic Temporal Database

If you treat the structure in Figure 16 as a schema, then you can associate with every element a population of particular instances - of people, learned societies, conferences, works and so on. Within the schema, you can associate with each element a relation - if you like thinking in relational database terms. Each of these relations has exactly the same attributes. The attribute list includes the following items:

$	surrogate	an internally generated code expressing the unique identity of the real world element
	. . .	
a1	antecedent-1	surrogates of first and
a2	antecedent-2	second ontological antecedents
	. . .	
+	start event	times or surrogates of events marking the start
-	finish event	and finish
	. . .	
@+	start authority	surrogates of agents, norms or communications
@-	finish authority	which determined the start and finish events
μ+	start mood	the epistemic, deontic and axiological attitudes
μ-	finish mood	associated with the start and finish of a surrogate

Every element has the same structure. The implementation of this is a Semantic Temporal Database in which it is almost impossible to record meaningless information because of the ontological constraints and the temporal constraints implicit in the ontological relationships.

To manipulate the semantic temporal database for retrieval purposes we use a 'legally oriented language', LEGOL, which is like NORMA but simpler to use because one may use potentially ambiguous expressions which may be detected by exploiting the knowledge in the schema to remove them. The operators in LEGOL are those in NORMA, all of which take account of time. Examples are **while, orwhile, whilenot, before, after, current, past, future, ever, start, finish** and so on. This makes the handling of time very simple in the expression of queries, far simpler than in any form of SQL yet encountered, and LEGOL is hence quite usable, after a minimum of training, by non-technical people.

The Normbase

The semantic temporal database is a platform from which we can take a major step beyond the conventional database management approach.

The conventional DBMS allows us to strip from the application programs a common core of data which can then be managed as a corporate resource, separated from the technological issues of computer applications. However, the residue in the application programs consists of a confused and indissoluble mixture of knowledge about the business and about the technology in the form of programming statements constructed to perform the required business functions whilst taking account of the transaction load on the given technical configuration. What we can do is to strip away all the business-specific knowledge and hold it, together with the data and the semantics, in the Normbase as a business resource.

To understand how this is done, consider the CRIS case, above. The ontological schema provides a framework for all the norms needed to run this organisation. Every aspect of the system's dynamics belongs as an authority for the start or the finish of an instance of either a universal or a particular in the problem domain. These norms have nothing to do with any information technology, they are parts of the problem-solving policy or of the underlying social structure.

Many of the authorities will be agents exercising their discretion. For example, a
selected(work)
will start at the discretion of the programme committee;
membership(person, committee)

may be left to the choice of the chair, in the first instance, and to the discretion of the committee subsequently. Cases of rigid rules will be the exception but

#priority

may well be determined this way. For example:

28. **for each conference**

member(organising committee:programme committee):contributor(selected(paper)
→ priority #1(member:contributor)

which says that a member of one of the committees or a contributor of a selected paper shall have priority value 1. The symbol ':' means 'orwhile', a kind of 'or' that takes account of time. To avoid the ambiguity that would assign a priority value to the members of committee of quite different conferences, we could preface the rule with the context statement "for each conference" saves us from having to embed a reference to the conference in every term of the formula. The same context definition would suit the next rule:

29. **for each conference**

(member(interest group,sponsor(conference)):author:referee);priority#1(person)
→ priority #2(person)

where we have used 'person' in the consequent instead of the list 'member:author:referee' as in 28 which could also have made use of the schema's ability to link 'person' to the persons selected by the condition formula. Two more operators are used; ',' meaning 'while' and ';' meaning 'whilenot' which are rather like 'and' and 'not' taking account of time.

The managers and policy-makers can define the functionality of their organisation when they have allocated responsibilities and defined the few essential norms. There is always the default solution of allocating all responsibilities to the root agent and relying on tacitly understood norms, so a fully specified system is easy to attain. From this a prototype supporting information system, automated where necessary, can be generated by the Normbase. If the transaction load is not heavy, the prototype might be the production system. This is accomplished entirely without having to think in terms of messages or data-elements, the managers think only of the substantive elements of the problem.

Arithmetic functions operate under semantic constraints. Numerical determinants cannot be manipulated without regard for their meanings: measurements are numbers plus semantics. For example, when two weights are added, the objects concerned must coexist, thus adding Plato's weight to Napoleon's would not be acceptable. However, the values of their weights may be stripped away from their semantics, regarded simply as numbers and summed. Operations on pure numbers are meaningless. The Normbase supplies the semantics which can be stripped away before submitting the parameters to the mathematical package and, when the results of a calculation are returned, the Normbase can assimilate them into the semantic structure to reflect the way the results are to be used in relation to actions. For example, a tax law might use the most outlandish formula taking account of the number of children a person has had, the tax he paid two years ago, the current index of inflation and so on, in order to calculate an allowance - there is nothing to stop a government in power from making the craziest of laws - we simply strip away the semantics of the input parameters, do the arithmetic as a mathematical exercise, and treat the result as the value of a determiner of a tax liability. Thus mathematical functionality of any complexity can be added without problem.

Architecture of the Normbase

Semiotics is embedded in the architecture of the Normbase. Figure 17 shows the layers of the software which correspond to the layers in the semiotic framework.

- The central core contains all the surrogate structures that are the analogues of social norms and the behaviour of the people involved.
- Outside that is a layer which received intentional messages and responds to them by changing the core.
- Before that, incoming messages have their meanings supplied by the semantic schema in

the third layer.

- The fourth layer contains the dictionaries which translate between the surrogate codes and the vocabularies of different languages. This translation layer might, one day, be able to put messages into the correct syntax for a natural language but presently this is not our interest. Translations into formal languages will also take place here so that the Normbase can communicate with other computer systems 'speaking' COBOL, for example or databases with arbitrary schemas.
- In the fifth layer, outgoing messages are put into and incoming messages are stripped from their interface structures. This is where redundancy is added or exploited, where the message is matched to the statistical properties of the outside channels at the empiric level. These channels may be human, so form designs, screen designs, choice of font or graphic elements would be exercised here. Technical channels would need the messages to be embedded in the right 'envelopes' for transmission - the OSI standard would be placed here.
- Finally the last and sixth layer switches the messages to the correct physical channel, making any optimising decisions on economic grounds if necessary.

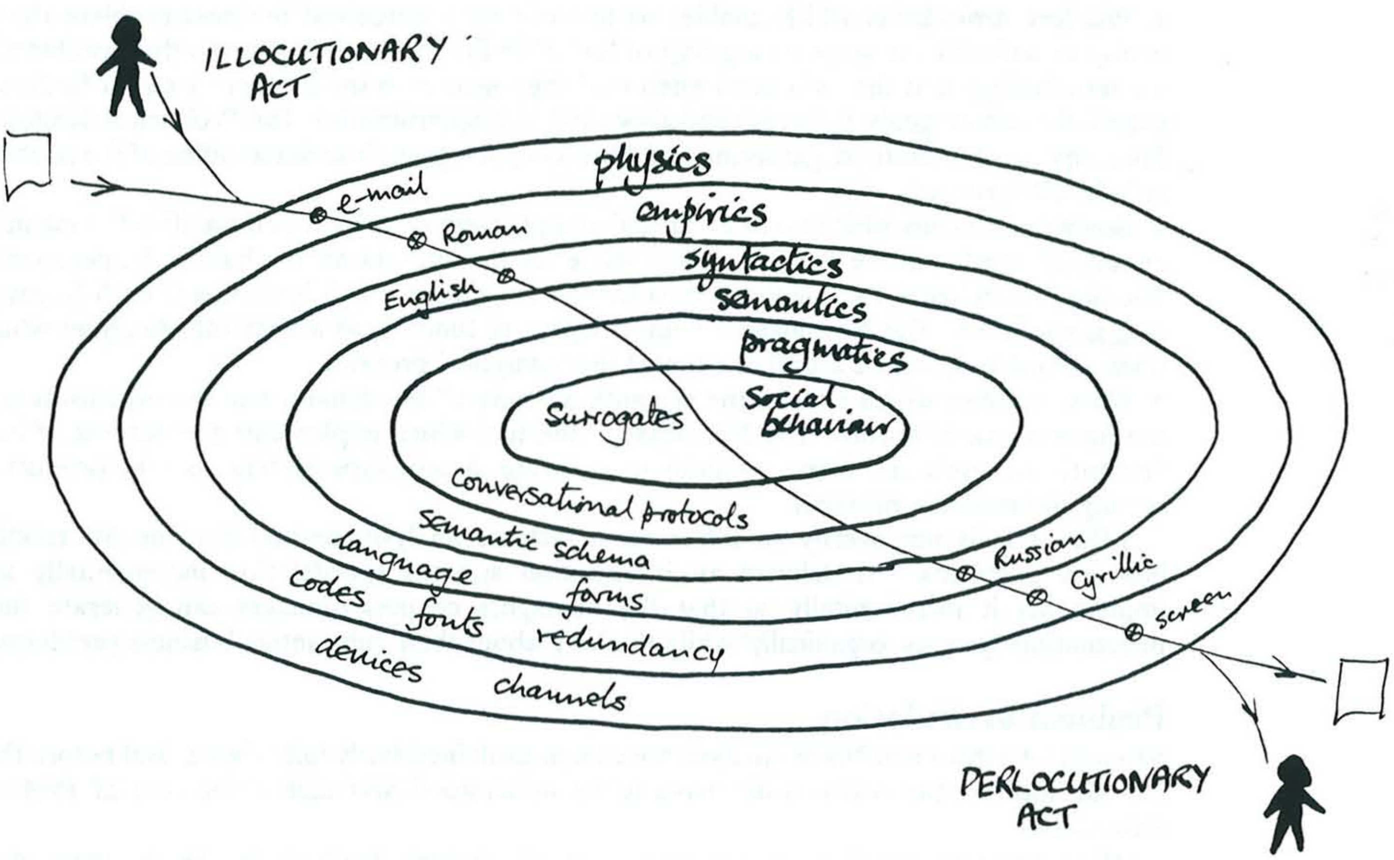

Figure 17: The architecture of the Normbase

Currently (September 1993) we have a Semantic Temporal Database which has been used for nearly four years to implement a large administrative system, and a new, portable LAN version of the STDB is nearing completion. Whereas the STDB does not interpret complex expressions such as the norms illustrated above, the Normbase does. Currently, a Normbase is running the core of the system on a PC, under Windows, and it is being developed further. The Normbase has a simple interface using the ontology chart whenever possible both to support analysis and to store and retrieve particular facts. Using the chart, it is easy to browse through a huge schema and examine particular data where required. LEGOL expressions, like the conditions in 28 and 29, above, allow data-sets to be selected quite easily. It is planned to use the STDB-LAN as a server for Normbase work-stations with the added goal of making it possible for users to evolve their own systems in an 'organic' process supported by the analytical tools and prototyping capability of the Normbase.

User-Oriented Analysis and Specification Methods

The third major area of applications of semiotics in the *processes* of problem-solving, analysis, specification and organisational design. Having a specification formalism does not necessarily provide any help with the process of making a specification: predicate logic is a fine formalism which has PROLOG as an implementation but the user has little guidance from the logic itself in how to develop a specification. NORMA, on the other hand, provides a great deal of help in the analytical process and further use of semiotic theory supports this feature. The result is a set of 'Methods to Elicit, Analyze and Specify User Requirements', or MEASUR.

MEASUR

The methodology we have developed on this semiological foundation is called 'MEASUR'. It has three main sets of techniques with supporting tools:

♦ *Problem Articulation* which enables us to work on a perceived business problem that is poorly articulated. It supports a group of 'users' in finding ways to express their problem in the terminology that they will need when working together in the problem area. It facilitates negotiation about goals, priorities and allocation of responsibilities. The Problem Articulation Tool supports the method, gathering and organising the group's understanding of the problem in indexed hypertext.

♦ *Semantic Analysis* which takes an articulated problem statement and clarifies the meaning of each concept with the precision appropriate for the operations that have to be performed. The method delivers a semantic schema for the Normbase which implements each fragment as it is specified. The Normbase is being adapted to function as a Semantic Analyzer which, when completed, will be a tool to support this analytical process.

♦ *Norm Analysis* which adds to the semantic schema all the dynamics of the organisation in the form of social norms. The Normbase is the tool which implements the outputs of both Semantic Analysis and Norm Analysis to generate, a prototype system, as a by-product of solving the business problem.

MEASUR is not overtly an information systems analysis method but one for tackling business problems. It delivers an information systems specification incrementally and implements it incrementally so that the managers or policy-makers can generate their information systems 'organically' while thinking about their substantive business problems.

Problem articulation

Semantic Analysis and Norm Analysis have been explained fairly fully above. But before they can be applied, the wider issues have to be understood and that is the role of Problem Articulation.

Many business problem arise as rather vaguely defined needs to be met or issues to be considered or crises to be faced. These ill-defined problems must be articulated in reasonably clear language if they are to be solved by a group of people. This is accomplished by undertaking what is often called "soft systems analysis". A range of valuable methods are available for this phase of analysis, many of them being introduced in Rosenhead (1987). Most soft systems analysis methods need to be applied with the assistance of a professional facilitator. The special advantage of Problem Articulation Methods, PAM, is richer structure they provide to help the users. A facilitator is not required because the methods can be taught to all the people likely to be engaged in innovative work. Equipped with PAM, a team of problem-solvers has been shown to improve their performance in terms of inventiveness, thoroughness, self-critical analysis, clarity of purpose, avoidance of unnecessary conflict and speed of working (Bruijn 1993). The structures of PAM also enables the organisation to accumulate and preserve their knowledge of how to innovative in various domains and so avoid having to re-invent the wheel each time.

The techniques of Problem Articulation are rooted in semiotics. Each of the techniques is briefly described.

♦ *Semiotic Analysis* treats the organisation as an information system and examines its functioning on each of the six levels of the semiotic framework. The result is a wide-ranging audit of strategic strengths, weaknesses, opportunities and threats. (See Liu 1993)

♦ *Valuation Framing* identifies the stakeholders and analyses the cultural norms according to which they value, positively or negatively, the present position and any future solution. It is the starting point for a thorough cost-benefit analysis but its most important role is to prevent the problem-solvers from overlooking the so-called 'intangible' aspects which often prove to be the downfall of innovative projects.

♦ *Collateral Analysis* is almost the opposite of the familiar top-down, structured analysis because it looks outwards from the situation to be achieved and asks about the surrounding 'collateral systems' that will put the solution into operation and keep it functioning: including the cycles for operations construction, maintenance, back-up, exchange and the life-cycle, of course. These systems have a structure which is a direct application of the syntax of Norma. Each collateral system is a unit system.

♦ *Unit System Analysis* treats every component of the problem and its solution as a repertoire of behaviour, that is, as an affordance of the community making the innovation. Each unit system may be thought of as a little system of laws governing the actions of an organisational unit. The analysis looks for the responsible agents, time, resources, criteria of performance and so on. Combining this information with the results of Collateral Analysis leads directly to a project plan, schedule and a management tool.

♦ *System Morphology* looks at the internal structure of each unit system, using the main divisions shown in Figure 5, above: the informal, formal and technical norms. The unique feature of this technique is that it draws attention to the interface between the informal culture and the formal bureaucracy. Most methods emphasise rationality and they tend to generate solutions which increase the amount of bureaucracy. This technique emphasises the neglected interface between the informal and formal and so enables one to remove elements of bureaucracy, replacing them by improving the norms in the corporate culture. System Morphology also looks at the hierarchy of norms governing the flow of information and the exercise of control.

♦ *Contention Management* helps in resolving conflicts between different world-views revealed by the other techniques (Kolkman 1993).

Problem Articulation makes sure that we consider the whole problem and put it into the context of the existing informal systems. Many innovations, information systems included, fail because of lack of complete analysis and because of disregard for the socio-technical and cultural environment.

Complexity Reduction

The weakest link in the development of computer-based information systems lies in the analysis and specification of requirements. Once these are in good shape - complete, precise and formal - we can usually deliver a serviceable technical product to meet them. Sadly, a large percentage of IT investments fail for lack of an adequate statement of requirements.

What can we do about it? One answer is suggested by a cartoon in Jones (1981) of an Egyptian scribe carving, on the wall of a tomb, a tally of soldiers, one hieroglyph for each. His companion asks "Isn't there a simpler way of writing 'The Pharaoh has 10,000 soldiers?" Methods of communicating complex ideas precisely have always depended upon our notations. The ancient Egyptians were hindered by a form of writing that was only able to create a rather ambiguous aide-mémoire. To learn it was such a huge investment that writing remained the preserve of a small community of priests, other people had to take their work on trust. Much the same may be said of requirement specification notations today. Technically oriented systems analysts have replaced the priests and managers and other computer users are now as puzzled as were their ancient predecessor by the papyri written by the priests. In ancient times the breakthrough came with the alphabet, "the noblest invention" (Mavor c1785), which made the link between sound and graphic signs simple and reliable. With the coming of the alphabet the priestly monopoly was broken and the methods of communicating in graphic form became widely available, to such an extent that quite new ways of organising society, of understanding the world and even of thinking (abstract concepts and logic, for example) became possible.

Experience with MEASUR suggests that much can be done to improve our notations. Where do we look for improvements? I think that the notion of an invariant, which has been often underlined in the discussions above, provides a clue. Each semiological layer must be understood and specified. Mixing the levels together results in needless complexity. This is the fault of the Classical methods. They allow us only to describe a world populated with data-objects and these plus the functions performed with them are all that their notations allow us to represent. They assume that words have intrinsic meanings so that there is no need to account for them explicitly, it is sufficient that the analysts use them. Moreover they give meanings to character-strings that they invent purely for their documentation (opening textbooks at random, I find the following examples: ENEW = customer opens account with deposit, HASPRE = has prerequisite). Real meanings cannot be given, they have to be 'grown' as norms in the minds of people who translate the words into actions. Classical methods embed the norms governing actions within the computer functions they specify. And there is no explicit place for the concepts of human responsibility and intention which are basic to running an organisation. MEASUR makes use of the invariance of meanings, intentions and norms, leaving the technical expression of them to be varied in any way that improves the efficiency of the symbol manipulation by computing machinery.

Semiotics leads us to discover features and structures that are invariant across many problems. Problem Articulation, for example, we have applied to such a diverse range of issues (a new concurrent engineering department, a new automotive product, a system for the automatic assessment of insurance claims, catering for patients in a group of hospitals, integration of subsidiaries into the parent company, running a conference, setting up a new company . . .) that we are confident it can be used without serious restriction. Compare this with one of the classical top-down structured methods for information systems projects.

Collateral Analysis	***A well known IS project-planning method***
1 project =	1 project =
7 cycles	6 sub-projects
17 tasks	51 tasks
	206 sub-tasks
24 items in total	263 items in total
which sub-divide	all with different names
on the same basis	in seemingly random relationships

conceptual complexity for IS projects alone reduced in ratio 11 to 1

Note the additional benefit for a company having n different kinds of projects to analyze!

We have also experienced huge reductions in documentation when a project using Classical methods converted to Semantic Analysis as its main technique. Instead of dozens of pages of diagrams of the dataflow type, plus pages of E-R diagrams incomprehensible to management, and a data dictionary defining a new vocabulary invented by the analysts, the user documentation was reduced, virtually to a couple of pages which the professionals could understand.

The reason for this is that every item in an Ontology Chart has the same invariant structure, as explained above. Here is the comparison between the two notations.

Ontology Charts	**Conventional methods**
Each element in a chart determines a code, two events, two authorities or processes, possibly eight moods.	The conventional equivalent would require a separate element fore each of these or between 7 and 15 items.
Each line in represents a number of semantic constraints.	The equivalents would be expressed separately as two time constraints, two constraints on the authorities and at least 1 and typically 3 links in a data model.

If each element in an Ontology Chart has two links, 1 element needs 7 to 15 conventional elements and the 2 links need 12 conventional elements

conceptual complexity is reduced roughly in the ratio: 8 to 1

In addition, the links in the Ontology Chart also express temporal constraints which would have to be expressed, conventionally in rather complex logical formulas.

Finally, the Legol notation which can be used for manipulating the Semantic Temporal Database is very easy to understand, containing expressions such as **a while b** which translate into a whole page of SQL if the data were to be manipulated in a relational database. The complexity reduction is as follows:

Legol expression	SQL version
a while b	[one page of incomprehensible code]
1 operator	35 operators
2 operands	64 operands
3 concepts in total	99 concepts in total

complexity of functional specification reduced in ratio : 33 to 1

If we can make the specification of information systems so much simpler, we shall obtain the necessary input from the user community, the present lack of which accounts for many abject and expensive project failures.

Gordon Scarrott (199???) pointed out that the available or usable entropy of a source should be partitioned into a communicable component and an organising part. It is only in the artificial examples we find in textbooks on signalling theory that all the available entropy serves to carry messages. In life, natural sources of messages are hugely complex and most of the possible combinations of signals are irrelevant to the parties trying to communicate, even if they are meaningful or true. The knowledge of what is relevant, meaningful and valid is shared by the parties and kept in repair by them as a part of their common stock of knowledge. If the entropy to organise the message stream swallows up a high proportion of the total available entropy, little will be communicated with great effort and high risk of error. Egyptian hieroglyphs are typical of sources that consume a high proportion of the available entropy, needing walls and walls of characters to say a modest amount. Classical system specifications are similar.

Scarrott has used his law of communicable entropy to explain the disappointing results that have seemed to flow from a major improvement in the performance of electronic devices: a ten-fold performance increase at a component level will be partially absorbed at the assembly level by the structures necessary to organise the components so that they may perform more complex and useful functions. This is also significant in relation to messages. When the scale of the system we are have to specify increases by an order of magnitude (as when we go from internal company applications on a small scale to huge national and internation systems) we shall expend disproportionately larger overhead of organising entropy to be able to handle these messages. To lower the cost of the overhead we must find organising principles that we had not perceived before, as did the alphabet inventors. The semiotic principles, because of their layers of invariance, make it possible to organise notations and systems more economically, can bring within our competence the specification and design of systems too large and complex for our current methods.

The cosy little applications each within the walls of one organisation make up a diminishing proportion of our work. Most systems are, potentially at least, a part of a global network. We must think on a larger scale than has been our custom and we must organise better our ways of thinking about information systems.

Conclusion

Any information to have a value must have a social effect. That is, perhaps, the most important proposition implied by the semiotic framework. A sign to be valuable must be well-formed and used effectively on each semiological level. A communication must be physically received, readable and grammatical, its meaning must be clear and so must its intention, and finally the receiver must respond appropriately.

Information technology deals with the first three levels but there is little it can do about meanings, intentions and interpretations in most cases. Information systems that do not function on all levels will not serve society or any of its organisations. Signs are social instruments and useful information technology must always a part of the social fabric and needs to be understood, designed, built and used as such.

The conclusion I wish to draw is that *computing and telematics should be treated as branches of the social sciences.* Although they draw upon resources from electrical engineering and mathematics, to treat them as extensions of those disciplines is to risk a misdirection of effort.

I have tried to demonstrate that a social perspective, especially if harnessed by semiotics to the understanding of information, can bring useful new ideas to extend and enrich computer science. Conversely, I also believe that, as a formal branch of the social sciences, an enlarged computer science could open up new and fruitful lines of enquiry concerning the functioning of social and provide new ways of modelling social behaviour.

Moreover, such a partnership will be likely to harness information technology far more effectively for our global economic benefit.

Acknowledgements

I should like to acknowledge the work of the members of the research team, especially the current members, Yasser Ades, Lena Chen, Martin Kolkman, Kecheng Liu, Karel Thönissen and Joost Zuurbier.

References

Ades, Yasser, *et al* (1989), *Namat User Manual*, Internal Report, University of Qatar, Doha

Andersen, P. Bogh. (1990), A Theory of Computer Semiotics. *Semiotic Approaches to Construction and Assessment of Computer Systems*, Cambridge University Press

Bruijn, Jan (1993), *Groups Empowered with Problem Articulation,* MBA Thesis, Twente School of Management, University of Twente, Enschede

Gibson James J., (1968), *The senses considered as perceptual systems*, Allea & Urwin, London.

Gibson, James, J, (1979), *The Ecological Approach to Visual Perception*, Boston, Houghton Mifflin Company.

Holt, A, H.R. Ramsey & J.D. Grimes, (1983) "Coordination system technology as the basis for a programming environment" *Electrical Communications* vol 57, pp. 307-314.

Jirotka, Marina, Joseph Goguen and Matthew Bickerton (1993), *Social and Technological Issues in Requirements Engineering*, Academic Press, London and New York.

Jones, Cliff B. (1980) *Software Development: A Rigorous Approach*, Prentice Hall, Englewood Cliffs

Kolkman, Martin (1993), *Problem Articulation Methods*, PhD Thesis, University of Twente, Enschede

Kitcher, Philip (1984), *The Nature of Mathematical Knowledge*, Oxford University Press.

Lehtinen, E. and K. Lyytinen (1986), "An action based model of information systems" *Information Systems*, vol 13 no 4 pp. 299-318

Liu, Kecheng (1993), *Semiotics Applied to Information Systems Development*, PhD Thesis, University of Twente

Locke, John, 1690/1959 , *Essay Concerning Human Understanding*, unabridged edtion 1959, Dover, New York.

Michaels, C.F. and C.Carello, (1981), *Direct Perception*, Englewood Cliffs, Prentice-Hall.

Nauta, Doede, (1972), *The Meaning of Information*, Mouton, The Hague.

Olle, T.W., H.G. Sol, and A. Verrijn-Stuart (1982), *Information Systems Design Methodologies: a Comparative Review*, North Holland, Amsterdam

Olle, T.W., H.G. Sol, and C.J. Tully, (1983), *Information Systems Design Methodologies: A Feature Analysis*, North Holland, Amsterdam.

Olle, T.W., A.A. Verrijn-Stuart, and I. Bhabuta (1988), *Computerized assistance during the information systems life cycle*, North Holland, Amsterdam.

Pierce C. S., (1931-35), *Collected Papers*, (6 volumes), Hartshorne C. & P. Weiss (eds.), Cambridge, Mass. Harvard U.P.

Russell, Bertrand (1956), "The philosophy of logical atomism" in his *Logic and Knowledge*, R.C. Marsh (ed), London 1956.

de Saussure, Ferdinand (1949) *Cours de Linguistique Général*, Paris

Scarrott, Gordon (1982), "Some consequences of recursion in human affairs", IEE Proc. Vol 129, Pt A, No 1, Jan 1982

Searle, John R. 1969, *Speech Acts - An Essay in the Philosophy of Language*, Cambridge, Cambridge University Press.

Searle John R. and Daniel Vanderveken (1985), *Foundations of Illocutionary Logic*, Cambridge University Press, Cambridge

Stamper, R. (1973) *Information in Business and Administrative Systems*, Batsford, London & Wiley, New York.

Stamper, R.K. (1979), "Towards a Semantic Normal Form", in G. Bracci and G.M. Nijssen (eds) *Database Architecture,* North Holland, Amsterdam.

Stamper, R.K. 1985c, "A Logic of Social Norms for the Semantics of Business Information", in Steel and Meersmann

Stamper, R.K., J. Backhouse, S. Marche, and K. Althaus, (1988) "Semantic Normal Form?", in K.P. Jones (ed), *Meaning, the Frontier of Informatics,* ASLIB, London

Stamper, R.K. (1993), "Social Norms in Requirments Analysis - an Outline of MEASUR", in Jirotka *et al*

Stamper, R., K. Liu (1993) "Organisational Dynamics, Social Norms and Information Systems" University of Twente, Enschede

Steel,T. and R.Meersmann (eds) 1985. Database Semantics, Amsterdam, North Holland.

Tarski, A. (1935), "Concept of truth in formalized languages" in *Logic, Semantics and Mathematics*, pp. 152-278, ed by Tarski, Oxford University Press.

Veryard, Richard (1992), *Information Modelling: Practical Guidance,* Prentice Hall, New York, London.

Wittgenstein, Ludwig, (1953), *Philosophical Investigations*, Trans. G.E.M. Anscombe, Oxford, Basil Blackwell

DISCUSSION

Rapporteurs: Andrew Blyth and John Dobson

Lecture Two

Professor Randell remarked that automation often leads to organisations rethinking their formal procedures. Professor Stamper replied that technology can play a role in organised and organisational behaviour, and that this role can be the catalyst for change. Of course the role will be multi faceted and may have positive or negative consequences. Professor van Rijsbergen remarked that some aspects of an informal system are not specifyable and therefore may not be modelled using normalisation. Professor Stamper replied that of course some knowledge within an organisation will be tacit knowledge and that it may be impossible or undesirable to make such knowledge explicit. However norms of behaviour must be modelled in context and it is this that makes the link between the informal and formal system possible. Professor Randell then asked if there are any examples of where real organisations have applied this methodology and what their conclusions where. Professor Stamper answered by stating that a middle eastern university had applied this methodology to the development of an administration system with the result that software maintenance costs had been cut by about one fifth. Professor Capriz then asked what the relationship is between Professor Stamper's methodology and classical methodologies such as SADT. Professor Stamper answered by saying that classical methodologies are based upon the information plumbing paradigm in that they are only concerned with how information flows. Thus they do not address the semantic level of why does information flow within an organisation.