

Organisational Dynamics, Social Norms and Information Systems

Ronald Stamper
School of Management Studies
University of Twente
P.O. Box 217, 7500 AE Enschede, NL

Kecheng Liu
School of Computing
Staffordshire University
Beaconside, Stafford ST18 0AD, UK

Abstract

An organisation is defined as a system of social norms. The concept of a norm is the social analogue of the concept of an affordance which Gibson introduced to account for perception of the physical world. Their conjunction is a logic of Norms and Affordances, Norma, which serves as a meta-model for dynamic organisational systems. All elements in a Norma model have bounded periods of existence. All the dynamics are specified by the authorities governing these bounds. A system to implement Norma has been constructed, called a 'Normbase' handle time intrinsically to achieve a huge reduction in the complexity of a specification. The Normbase allows a system to be prototyped dynamically, as the specification evolves. These methods have been used with great success in reducing implementation and maintenance costs.

If we are beginning to study the modelling of organisational dynamics and information systems we may appropriately ask what we mean when we talk of 'an organisation'

This paper defines organisation in terms of social norms and then shows that these norms define the kernel structure of the information system necessary to support that pattern of organisation. The model which this view produces gives us as detailed and precise a picture of the dynamics of an organisation as it is reasonably possible to construct. It covers changes at all levels and the roles that people play in bringing change about. The model also leads to methods for investigating information systems where dynamics are on centre-stage from the beginning. And, finally, we show that these methods lead to techniques for building systems that are very easy to maintain and, therefore able to function well in a dynamic business environment.

This paper concentrates on the conceptual background. Space allows us to deal with details of the

modelling tool and with applications only very briefly.

1: (An) Organisation and norms

A group of people who act in an organised manner certainly display regularity in their behaviour: no regularity, no organisation. The regularities we can characterise as behavioral norms. In certain kinds of circumstances the people involved know how they are expected to behave. These shared expectations help the members of the group to coordinate their behaviour. Thus, at least one key ingredient in any specification of an organisation should be the behavioral norms that allow it to function in an organised way. In addition perceptual, evaluative and cognitive norms [15] are involved in organised behaviour.

Organisation is found without there being an *organisation*, as such. Just look at any busy city street. Hundreds of people, some in powerful cars, some with small children, manage to carry on their daily lives in an orderly way. The prevailing culture provides all the necessary norms, with some support from legal norms. No one is the boss; no one manages this complex scene; no particular organisation is responsible for delivering every-day orderliness in a city street, it is a cultural product which, in many places, has persisted for and evolved over many centuries. By way of contrast, *an organisation* has a definite start and finish to its existence.

Basic to the dynamics of an organisation is a life span. Its organisation depends upon social norms, including those of the ambient culture or cultures. The dynamics of the organisation will be affected by this relationship with the culture of its members. It will not be easy to establish non-cultural norms if they are seriously in conflict with the cultural norms (this is the root of most resistance to change). Perhaps we can characterise an organisation as a structure of social norms which allow a group of people to act together in a coordinated way for some purposes. The common

purpose may be little more than to maintain the existence of the organisation so that it may serve as an arena for the pursuit of diverse individual and sectional interests of its members - a political party springs to mind as a good example of this kind of organisation. At the other extreme are our business and administrative organisations, and the various ad hoc structures within them which have been created to achieve more or less precise goals. The criteria for the existence of an organisation are important aspects of its dynamics.

Note that norms in general include informal norms in a team or in the background culture as well as formal, explicit rules, including background legal norms and legislation.

2: Norms and information requirements

A typical norm has a shape that tells us the essentials about the communications necessary to act upon it:

Whenever <situation> **then** <an agent>
is <deontic attitude> **to** <action>

See the example of a norm in a library application:

Whenever *a book is overdue* **then** *Librarian*
is obliged
to send an overdue notice to borrower.

The obligation of a borrower to return the book before its due date can be defined in a similar way.

The agent (person or group) is necessarily the recipient of messages about situations that may fall under the norm. It may be that someone or some computing device keeps track of events and then signals to the agent only when relevant situations arise. Then the agent will be obliged, permitted, not required, or forbidden to perform some action. Typically many of these actions are not physical acts performed directly by the agent but either semiological acts (such as claiming or informing or selecting . . .) or acts performed by someone else on behalf of the agent. Even when the agent has to perform some physical act, the other people in the organisation will need to know what has happened, and this implies a message should be created. So the action side of the norm indicates the outward flow of messages from agents.

Thus a complete picture of an organisation in terms of its norms will define the essential information flows. However the specification of the norms does not prescribe the exact 'dataflows' in the usual sense. For a given set of norms, there may be an infinity of different patterns of dataflow that would serve the organisation's purposes. Information systems analysis methods commonly require that some particular dataflow solution should be recorded as an essential

part of a system specification. If this is only one of many equally satisfactory dataflow patterns, then all such methods lead to over-constrained, and therefore misleading specifications. A norm model gives us a minimal definition which makes organisational change easier to understand and control.

This minimal definition is especially valuable for complex systems. The dataflow idea was a natural consequence of thinking about the analysis and design of routine administrative systems. Such models describe uniform networks of pipelines with their sequence the only aspect of dynamics represented. In many other situations it makes little sense to work in terms of data flowing in some fixed pattern. For example, in cooperative work, all the norms may affect all the agents, and information has to be diffused over the whole community. Analysis and design methods for CSCW systems cannot be based on the over-simplified dataflow approach. The pattern of message passing can change from moment to moment. The relatively stable, underlying invariant is the system of social norms.

The research reported here began with the observation that a large government department exists to give effect to a relatively brief set of legal norms which are then turned into huge volumes of clerical codes in the dataflow style of expression. In the UK Department of Social Security the ratio is roughly 30-plus slim volumes of norms c.20% of which become 400 thick volumes of clerical codes. Modelling systems using norms reduces documentation in a similar proportion.

3: Perceptual norms

The most basic norms, arguably, are those which determine how we divide up our world. But how do we perceive things, relationships, states of affairs and so on, and how do we determine the boundaries round such percepts?

Certainly, no world is perceived without an agent to do so and perceiving always implies some kind of behaviour, at least of a passive kind. For our purposes, the most relevant approach to perception we have found to be that of James Gibson. He worked on the perceptual problems of pilots during the second world war and created new insights into the nature of perception which may be extended into the organisational domain with advantage.

Gibson gave the agent a central, creative role in perception instead of the passive role of a receiver of sense data which the agent's eyes and brain assemble into a perception of some object, first on the retina and then in the mind. The classical picture of visual

perception shows rays of light radiating from the object, a few of them being intercepted, sensed and interpreted by an observer; this is illustrated (from [4]) in Figure 1.

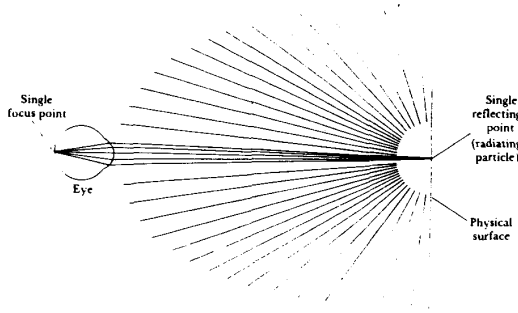


Figure 1: Classical paradigm for perception - ready-made objects being sensed

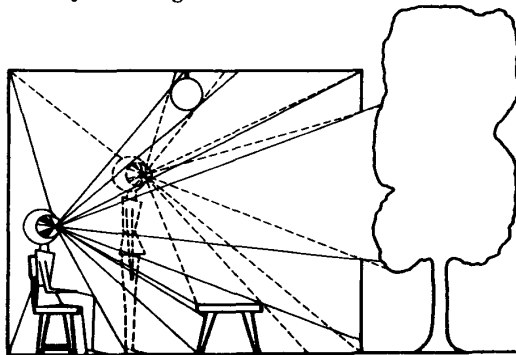


Figure 2: Gibson's paradigm for the perception of invariants in the ambient information

Gibson's theory replaces this classical diagram by a totally different one of an agent bombarded with signals from all directions, an agent swimming in a sea of information, an agent who perceives *invariants* in this information-loaded environment. Figure 2 (from [4]) shows a little of the optical environment sensed by an observer in two different positions. The agent also senses sounds and their echos from the walls of the room, smells, temperatures and, most importantly, his or her own movements. The invariants the agent recognises are those that matter for its survival or well-being. They depend upon the structure of both agent and environment and upon the relationships between them. Gibson called these types of invariants 'affordances'.

Anyone walking but especially anyone driving a car towards a wall or flying a plane towards a runway depends for safety upon an ability to perceive, in the combination of visual field and kinaesthetic sensations, a set of invariants that includes the following invariants

in the optical array: a central point towards which one is moving; an array of closed curves moving out from that central point; at invariant speeds, depending upon their proximity and angle from the centre and speed of approach; with invariance in the rate of change of these speeds for a given acceleration.

These invariants are what we perceive as *towards* something when we are moving, that something being the stationary, invariant point. The computer games programmer produces an illusion of movement by simulating these invariants without the important element supplied by the kinaesthetic signals which the walker feels continually and the driver and pilot feel as they accelerate. Gibson's illustration is reproduced as Figure 3.

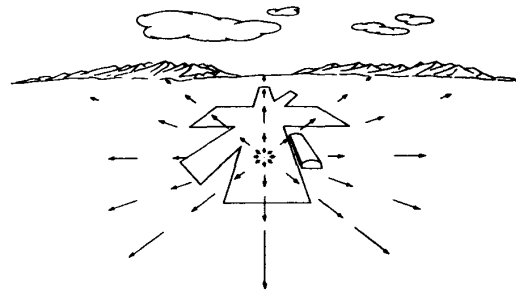


Figure 3: Invariants in the visual field of a pilot landing an aeroplane

This approach to perception introduces a remarkably unified ontology. Objects are no longer thought about as things that stand by themselves which have properties by themselves. A cup becomes an agent's experience of a repertoire of affordances - the ability to hold liquids in certain positions, the noise it makes in hitting various surfaces, the visual/tactile shape it displays, and so on. The entity-relationship-attribute confusion vanishes from this perspective. But there is a greater unification still.

Gibson was concerned with the perception of the physical world but the notion of an affordance was generalised in [16] to include the invariants that we perceive in our social world. If one has a copyright one should be able to expect an invariance in the behaviour of people towards one in respect of that work. A cup has a number of social invariants that are valuable, for example it enables us to drink liquids in a manner acceptable in polite company and it can enter into the invariant of ownership, just as a copyright can. The invariants in our social world are what we call norms.

4: A logic of norms and affordances

These insights point the way to constructing a language, based on the perception of invariants, for specifying organisational behaviour. We call it Norma, a logic of norms and affordances, and it serves as a specification language for social systems. Norma when implemented on a computer gives rise to a kind of CASE tool from which the information requirements of a system can be deduced from the norms.

The well-formed formulas of Norma all have the structure

Agent invariant

where the agent may be a responsible group or an individual. The invariant is a mechanism and, when it is used successfully by the agent on a particular occasion, we call the particular state of affairs a 'realisation'. A simple example is:

John cup

for John perceiving a cup with which he becomes a modified agent (John cup) able to experience other invariants, such as

John cup hold

for John then adding the invariant of holding it

John (cup, wall) towards [, = *while*]

where John perceives both cup and wall and the motion of the cup towards the wall.

The combination of affordances that we see in

John (cup, wall) towards

gives rise to a compound affordance, (cup, wall), which exists while John is perceiving both some wall and some cup. There are other combinations:

John (walk; sing) [; = *while not*]
meaning that John walks while not singing, or

John (talk; sing) [: = *or while*]
meaning that he is talking or singing (but we are indifferent which). These allow us to construct descriptions of complex states of invariance (or states of affairs) in a manner reminiscent of using propositional logic.

There is, however, a big difference in the notion of 'if . . . then . . .' in Norma and the truth-functional, material implication of classical logic.

John (happy → sing) or if John is happy then he sings

John (sing ← happy) or John sings whenever he is happy

are specifications of mechanisms which are invariant so long as that is how John behaves, even when he is neither happy nor singing. The period of existence of an affordance is independent of its activation.

Norma also allows us to construct names for affordances that are always related to any specific

affordance. Thus the affordance *talk* has associated with it the underlying ability to talk

John talk* or John is able to talk

John talk< or John is beginning to talk

John talk> or John is ending his talking

all of which are percepts of importance. Conversational fluency, of course, depends upon our instantly recognising the beginning and ending of turns at talking.

Our behaviour in organisations is largely concerned with using signs to get things done, for which we need signs with a certain meaning. To represent them, the convention in Norma is exactly as in writing, thus:

John "Mary smile" says

uses the sign called "Mary smile" meaning

Mary smile

to perform the act of saying it. This allows us to construct norms to govern behaviour:

Society person(happy → "person sing"#permitted)

for the norm that a person in a particular Society, when happy shall be permitted to sing, or realise the invariant

Society person sing

Note that the permission depends for its existence upon having a sign for whatever is permitted.

5: Invariants and dynamics

The word 'invariant' suggests lack of change, so why introduce a language based on the concept of invariants in a paper on dynamic modelling? Of course these invariants are only temporary and are, implicitly, always bounded by a start and a finish, their period of existence. Thus we want to be able to refer to the start of John singing

John sing+

or to the finish of him holding some cup

John cup hold-

But notice that we distinguish between the beginning and ending processes and the start and finish events. This is clearest in the computer implementation of Norma where the beginning and ending

John sing< and John sing>

generate records for separate percepts, while the start and finish operators

John sing+ and John sing-

are the names of pointers from

John sing

to other invariant states of affairs during which these events occurred.

The means of handling dynamics is increased by the operators for constructing names for the implied invariants of the periods before and after any invariant:

John sing\ and John sing/

but the most important aspect of the dynamic structure in Norma is provided by the operator implied by concatenation in the above examples. Thus

John object hold

tells us that holding is an invariant that can only exist when some object exists to be held. This relationship is one of *ontological dependency*, the object being antecedent and holding the dependent.

6: Evaluative norms

Even deeper than the perceptual norms are the evaluative norms. These determine whether some invariant is significant enough in physical, biological or social terms to warrant being recognised as a distinct percept.

What a particular society has evolved as the set of things its members perceive in the world is a reflection of its evaluative norms. It will often have bitter arguments when the value system is shifting or when sub-cultures with different values come into conflict. For example the boundaries defining the start and finish of a person are currently under hot debate as the 'abortion issue' and the 'life-support issue'. The behavioral and status norms (see below) written into our laws often provide explicit resolutions of these dynamic parameters. Values are brought into play by the control over the dynamics of the system.

When explicit norms are not available, we can place with some agent the responsibility for deciding when things start and finish. We represent that as, for example:

Nation person@

or

Company (person, department) works_in@

which maps to a sub-agent which is a component of the agent in the formula. For example a law court within a particular nation state, in the case of a person's existence; or in the business example it may be the Personnel Director. This linkage between the formal system and the people involved is a way of introducing the value judgements of certain representative members of society without attempting to be explicit about them. Of course, if the model is being used to describe, rather than serve the organisation, some function will be needed to provide values as the model is operated as a simulation. Our intention is to model systems in order to use them as shared knowledge-bases for the agents involved, and in this case the *authority function* shows how the computer interpreting the model should communicate with these agents.

7: Modelling of enterprise dynamics

We now have enough linguistic apparatus to allow us to model the dynamics of organisations. Notice that nothing appears in the specification without a period of existence associated with it. It is not possible to be more thorough in modelling the basic facts about the dynamics of a system.

Note the most important elements. The agents who account for the pro-active dynamics have a period of existence and some agent or norm will be associated with every start and finish by the authority function. Every perceptual ability will have a start and finish - these correspond to the data-types or object classes of conventional models, so that the schema of a Norma model is dynamic also. Particular invariants that are realised have their starts and finishes and these are the instances or data-objects of conventional data-models. The norms such as

Society person(happy → "person sing"permitted) have their starts and finishes which appear as enactment and abrogation in a legal context. Norms determining the making of norms also have their dynamics, for example, contract law:

Society ("(person-1, person-2) act" intend illegal)
→((person-1, person-2) "(person-1,person-2) act"
contract) forbidden)

which says that the joint agent formed by two persons is forbidden to enter into a contract to perform some act which is illegal.

Even generic/specific relations are dynamic in this system. At one time we would have said

Society (fish ← (shark: mackerel: whale))

but today

Society (fish ← (shark: mackerel))

is our understanding of the genera.

In the system which manipulates these Norma models, the manipulation of time is central. The data are held in the form of one unified Norma statement. To build the model, the first task is to construct a picture of all the affordances (types or object classes) in the form of what we call an *ontological schema* or *ontology chart* (if in graphical form). This contains the key dynamic constraints in a form that is very easy to understand.

8: An illustration

A simplified portion of the well-known, standard CRIS case (to specify an information system for organising international conferences - see [12]) will suffice as a first illustration (Figure 4). The whole structure represents the behaviour of a root agent, assumed to be

human society in general, which is the arbiter of commonsense meanings. If we want to be more precise about meanings then we must rely upon more exactly circumscribed agencies. For example, the learned society, itself, depends for its constitution upon the nation-state in which it is legally incorporated. The meaning of 'conference' is operationally determined by the responsible learned society and may change from society to society. Sometimes the commonsense meaning is quite precise enough, as in the case of 'person'. Role names are introduced in parentheses whilst another graphic notation is the box containing a set of specific/generic relationships (eg work/paper) and whole/part marked with a dot (eg learned-society • interest-group). Some of the *affordances* are followed by a # to indicate that they may be instantiated by identifiable individuals, and the # sign precedes one element '#priority' to indicate its status as a determiner, a kind of generalisation of a measurement.

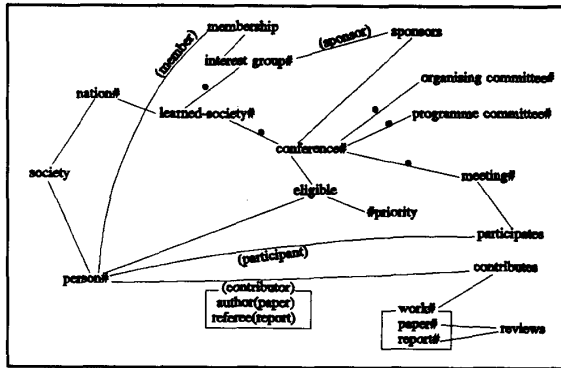


Figure 4: Ontology chart or semantic schema for organising a conference

A static case-study such as this can give no impression of the analytical strength of the method. Superficially Semantic Analysis is like data analysis but, on closer inspection it is fundamentally different. It is based, for example, upon a precise semantic principle - the relationship between language and action - which forces the users to arrive at a canonical solution. Using the conventional methods of data analysis, any number of different solutions are equally acceptable because the analysts are free to make many legitimate, arbitrary choices. The odds are that some of these arbitrary features will not match future organisational needs, and so give rise to systems which are difficult to adapt to change. However, using Semantic Analysis, differences in the schema always signal genuine differences in the underlying conceptions which can be removed by further careful analysis or, if necessary by negotiation. The resulting

semantic structure, free from arbitrary features, is robust. The constraints force the user to think very deeply about the problem domain and guide him to a higher quality of system specification.

Semantic Analysis models a dynamic enterprise by describing agents and their patterns of behaviour. The patterns of behaviour are universals and the instances are particulars. For example, *conference* is a universal which started sometime in the historically fairly recent past as an affordance of a learned-society, and it may have a finish in the distant future. It is an ontological dependent of the learned-society which can experience many particular conferences during its existence. Employing starts and finishes as inseparable properties of the agents and their behaviour enables us to store a series of conceptual models of a business organisation which change over time. The schema is treated in the same manner as the data. The particulars, as they start and finish, depict the changing world in the framework of the changing universals which comprise the world model.

There are constraints in the ontology at both universal and particular levels. At the universal level, the existence of each affordance must be constrained to the co-existence of its antecedents. The left-right positioning of two linked concepts in the ontology chart indicates this kind of antecedent-dependent relationship. In our example, learned-society is the antecedent of conference. At the particular level, the existences of instances cannot go beyond the existences of their universals. These two kinds of constraints are strictly followed in models and applications of databases using Semantic Analysis.

As a small example of this process of organisational problem-solving, consider the element *work* which is given here as ontologically dependent on the conference. In a sense, what constitutes a work for this organisation is what it chooses to be a work. The ontological structure tells us that, if the learned society decides to cancel the conference, then these works will cease to exist. However, a lawyer who knows the method, glancing at the chart, will no doubt point out that the authors will not want their works to be regarded as dependent for their existence on the conference, it will be important to take account of copyright issues when dealing with the papers submitted. This view would lead us to place a work as an ontological dependent of a nation which is the agent determining the existence of a literary work for the purposes of copyright. So we would recommend altering the chart displayed here, otherwise any administration concerning the editing and publishing of the proceedings might not be served correctly. This illustrates the power of one of the tests we apply to

draft semantic schemas - ask what the consequences are for the existence of each item when each of its antecedents ceases to exist. Dynamics underlie meanings. Until you have tried this you will not believe that such a simple analytical criterion could be so stringent.

Semantic Analysis has been applied to a very wide range of problems with marked success. The value of investing in a thorough analysis of the business problem has been demonstrated in practice. As a replacement for conventional data analysis, it has demonstrated its power to reduce the implementation costs but, more especially, the maintenance costs of systems.

9: The semantic temporal database

If you treat the structure in Figure 4 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 (if you like thinking in relational database terms) a relation. Each of these relations has exactly the same attributes that include the antecedents, a start, finish and authorities for the start and the finish. 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 language, Legol, which is like Norma but simpler to use because it exploits the knowledge in the schema to remove potential ambiguities. 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 usable, after a minimum of training, by non-technical people.

Legol handles the dynamic aspect as a integral part of each operation. Every operation on the semantic temporal database consists of two parts: a standard relational operation and a calculation on the temporality. The three Legol operators **while**, **orwhile** and **whilenot**, for example, involve temporal operations of intersection, conflation and exclusion respectively. Let us see some examples.

- (1) *eligible (person, conference#CRIS)*
will list persons who are eligible for the CRIS

conference. The start and finish times of the eligibility of each person are also listed.

- (2) *eligible (person, conference) whilenot contributes (person, work (conference#CRIS))*
Contributors of conference works (i.e. papers and report) are certainly eligible for the conference, but this lists all persons eligible on grounds other than their contributing a work to the conference. The listing will give all the historical data too. In order to avoid this one would have to add to the expression 'while now' where 'now' is a reference to the current time known by the system.

The semantic temporal database adopts a non-destructive update technique. Actions and events are recorded in the database with temporal stamps. New states of actions and events are appended with different times. No overwriting of data is necessary from a technical point of view. This allows one to maintain a continuous dynamic scene of an enterprise. In many business application areas, this non-destructive update safeguards essential information resource [1].

10: The Normbase

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

The normbase accepts the semantic model presented in the ontology chart as a schema. At the presentation level, an ontology chart contains the dependency relationships between the agents and actions, authorities for actions, and other information such as generic-specific and whole-part relations. All the agents and actions have a canonical data structure containing the same attributes. Therefore as soon as the ontology chart is defined and entered into the normbase, a semantic temporal database is generated automatically. The ontological dependencies and the business semantics serve as constraints during the database generation.

The major functional components in the normbase are the semantic temporal database, the norm table, the Legol interpreter and the triggering mechanism. The former two components maintain the business-specific knowledge while the latter two are concerned with applications of the knowledge. Norms are specified both in natural language and Legol. Because all the data elements in the semantic temporal database have start and finish times as their properties, their states can be constantly calculated in relation to time. The triggering mechanism calculates the states of events and actions described in the database against the triggering norms. When an event or an action meets the

conditions of a certain triggering norm, the action prescribed by the norm will be performed by the normbase.

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 inextricable mixture of knowledge about the business and about the technology in the form of programming statements formulated in order 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 or possibly of the cognitive model we are using.

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:

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. To avoid assigning a priority value to the members of committees of quite different conferences, we preface the rule with the *context statement* "for each conference" saving 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:

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:contributor' as in the previous example which could also have made use of the schema's ability to link 'person' to the persons selected by the condition formula.

The normbase assists management to solve problems by constantly monitoring changes of the business world and executing prespecified norms:

for conference#CRIS

6 month before meeting while

(contributes(person, paper) whilenot reviews(report, paper))

→ obligation(chair(organising committee), 'check refereeing of <paper>')

This norms says that 6 months before the meeting if there is any paper having not been reviewed then a message will be produced to inform the chairman of the organising committee that he is obliged to check this matter.

As far as the managers and policy-makers are concerned, they will have defined the functionality of their organisation fully when they have allocated responsibilities and defined the essential norms. From this a prototype information system, automated where necessary, can be generated by the Normbase to support the people working on this problem. 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. The Normbase works as an engineering environment for producing systems implementations quickly and dynamically. The systems specifications in the form of the ontology chart and associated norms are time-dependent. The conceptual elements appear in a specification model while the particular instances are stored in the database but categorised in accordance with the conceptual elements. The canonical structure for both the conceptual elements and the particular instances not only facilitates the construction of dynamic views of business world portrayed by the temporal data, but also allows one to have dynamic models of conceptual representation. By using the start time and finish time one can easily locate a conceptual model or a version of it and implement the model quickly as a Normbase prototype for an application. This offers the problem-owners and users an opportunity to understand the real meaning of the representation model, therefore justifications and adjustments can be made by the users.

In developing several versions of the normbase and applying this semantic approach to modelling and systems development, it has been observed that the

volume of specification documentation is moderate, normally not more than a few pages of ontology charts and explanations [2, 7]. Users can often understand the documents quickly after some brief instructions, although to teach them to be able to produce an acceptable ontology chart would require serious effort.

11: Re-engineering

Complexity in organisational information systems tends to grow as structures are patched to make them suitable for changed circumstances. The original version of the system always seems clean-cut and well designed. The implementation, however, is usually full of expedients and clever solutions to problems based on local optimisation. Even the design will contain many hidden, arbitrary devices waiting to be exposed by changes in business requirements for which they would have been clearly unsuitable [17], if only the changes had been anticipated. Re-engineering is necessary, from time to time, to sort out the accumulating mess.

A basic strategy for coping with complexity is to find good criteria for partitioning the problem into components that can remain, as far as possible, invariant as the other components change. The meta-problem is then to find such criteria.

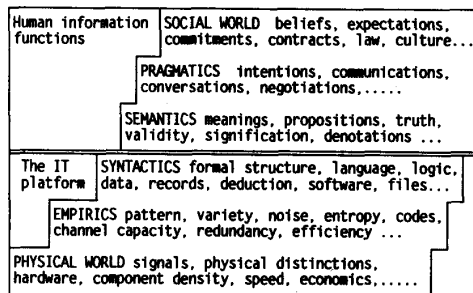


Figure 5: Layers of Semiological Structure

The methods described above are based upon semiotics, the doctrine of signs, a well-established framework for partitioning problems about information [8, 14, 10, 15]. Information is a vague term which we prefer to replace by the simpler primitive notion of a sign, something which stands for something else for a person in a particular culture. This is a very complex relationship, in fact. As shown in Figure 5, it is based upon physical tokens or signals which are then organised according to several, relatively independent layers of structure, until, at the top level, they have an impact on the social structure.

Information systems, if they are of any value, function correctly on all semiotic levels. However, the legacy of old systems that must be re-engineered have been constructed primarily with regard for the IT problems at the lower levels. The human and business functions of the upper three levels have been embedded inextricably in the technical structures. But from an organisational point of view they are what matter.

The norm-oriented methods described here adopt the opposite approach. Norma makes no concession whatever to the structuring needed at the technical levels, it deals only with semantic, pragmatic and social structures. Moreover, the implementation of Norma and Legol in the Normbase is built uncompromisingly on these semantic, pragmatic and social structures which dominate over the technical ones.

Re-engineering using these methods would depend upon building a thorough specification of what must be preserved for conducting the business. This certainly does not include the outdated bureaucracy probably embedded in the old systems. All the data-elements in the existing systems can be given precise semantics in terms of the ontological schema. Data from diverse sources can be assembled under the control of these semantics for processing in a semantic temporal database. The complex functionality of the existing programs will be replaced by a functionality defined in terms of the social norms governing the substantive business activity. The reconstructed systems will have a very stable core of semantic structures upon which the more volatile norms are carried.

From the point where the reconstructed systems are in the form of semantic and norm structures, changes in design are no more than changes in the data at the level of universals. The emphasis on dynamic modelling is justified by making the meta-system dynamic. Change is then smooth and cheap rather than the dominant chore and financial drain in systems management.

This illustrates that one has to be concerned with IT systems re-engineering as well business and corporate re-engineering which are appropriately handled by the norm-oriented methods.

12: Summary and conclusions

Dynamics lies at the heart of this method of conceptualising and modelling organisations. The dynamic elements of an organisation are principally the norms governing the behaviour of its members. The Norma language allows these norms to be expressed in a computer-interpretable form. This interpreter, called a Normbase, holds all information as well as the design of schema and functionality in a dynamic form where

every element has a bounded period of existence. No information is overwritten. All that happens is that the finish of the existence period is recorded and archiving frees space when necessary. The dynamic history of the system is fully accessible, even as it runs under a dynamic design.

This radical new approach has many benefits. *Dynamic enterprise modelling* has been demonstrated with the well-known CRIS case. *Diagramming techniques to capture time-related aspects* have been exemplified by the ontology chart for the CRIS case and by the explanation of the underlying formalism. The discipline of constructing the ontology chart has given rise to Semantic Analysis, a contribution to improving the *methodological aspects of modelling dynamics* which makes it easy to capture the logically necessary, dynamic constraints of a system. This methodology usually leads not only to *organisational problem-solving* but to the prior and more difficult identification of organisational problems. The meaning of 'work' in the CRIS case demonstrated this. The Semantic Temporal Database and the more sophisticated Normbase provide us with *support tools*, a means of *transferring models for design into actual implementations*, as well as a means for *transferring models for design into training instruments*. The Normbase produces an instant prototype allowing us to improve what we experience as the *dynamics in information systems development*. This advantage is felt most strongly in the role of these methods and tools for *process, business and corporate re-engineering*.

The method as a whole, incorporating Semantic Analysis and Norm Analysis, also has a set of Problem Articulation Methods and Tools, and it is called MEASUR [19].

13: Acknowledgements

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

14: References

1. Achueler, B.-M. (1977), *Update Reconsidered. Architecture and Models in Data Base Management Systems*, Nijssen, G.M. (editor), North-Holland Publishing Company, Amsterdam.
2. Ades, Y.M. (1989), *NAMAT System User Manual, SIS project*. University of Qatar.
3. Gibson James J. (1968), *The senses considered as perceptual systems*, Allea & Urwin, London.
4. Gibson, James J. (1979), *The Ecological Approach to*

5. Jones, K.P. (ed) (1988), *Meaning: the Frontier of Informatics*. ASLIB, London.
6. Kitcher, Philip (1984), *The Nature of Mathematical Knowledge*, Oxford University Press.
7. Liu, K., E. Boekkooi-Timminga and L. Sun (1990), *Systems Analysis of a Computerized Test Construction System (CONTEST)*. University of Twente, Enschede.
8. Locke, John (1690/1959), *Essay Concerning Human Understanding*, unabridged edition 1959, Dover, NY.
9. Michaels, C.F. and C.Carello (1981), *Direct Perception*, Englewood Cliffs, Prentice-Hall.
10. Nauta, Doede (1972), *The Meaning of Information*, Mouton, The Hague.
11. Newcomb, T.M., R.H. Turner and P.E. Converse (1965), *Social Psychology*. Holt, Rinehart Winston, London.
12. Olle, T.W., H.G. Sol, and A. Verrijn-Stuart (1982), *Information Systems Design Methodologies: a Comparative Review*, North Holland, Amsterdam
13. Olle, T.W., A.A. Verrijn-Stuart, and I. Bhabuta (1988), *Computerized assistance during the information systems life cycle*, North Holland, Amsterdam.
14. Pierce C. S. (1931-35), *Collected Papers*, (6 volumes), Hartshorne C. & P. Weiss (eds.), Cambridge, Mass. Harvard U.P.
15. Stamper, R. (1973), *Information in Business and Administrative Systems*, Batsford, London & Wiley, New York (2nd edition forthcoming from Blackwell, Oxford).
16. Stamper, R.K. (1985), *A Logic of Social Norms for the Semantics of Business Information*, in Steel and Meersmann
17. Stamper, R.K., J. Backhouse, S. Marche, K. Althaus (1988), *Semantic Normal Form?* In K.P. Jones.
18. Stamper, R., K. Liu (1992), *Semantic analysis and some fundamental concepts*, University of Twente, Enschede
19. Stamper, R.K. (1993), *Social Norms in Requirements Analysis - an outline of MEASUR*. in *Social and Technical Aspects of Requirements Engineering*. Edited by Bickerton, M., J. Goguen and M. Jirotko Academic Press (forthcoming).
20. Steel, T. and R. Meersmann (eds) (1985), *Database Semantics*, Amsterdam, North Holland.
21. Tarski, A. (1935), *Concept of truth in formalized languages*, in *Logic, Semantics and Mathematics*, pp. 152-278, ed by Tarski, Oxford University Press.