From Database to Normbase

R. STAMPER, K. LIU, M. KOLKMAN,
P. KLARENBERG, F. VAN SLOOTEN, Y. ADES
AND C. VAN SLOOTEN

After the database concept, we are ready for the normbase concept. The object is to decouple organizational and technical knowledge that are now mixed inextricably together in the application programs we write today. The underlying principle is to find a way of specifying a social system as a system of norms. Our existing languages, developed to handle machine-like structures, do not enable us to embed our technical systems comfortably into the far more subtle human systems they should be serving.

The present approach to the design of formal systems incorporates a philosophical position that obstructs one's thinking about the relationship between social systems and technical systems. The existing specification languages embody a view of the world as an objective reality. But social systems are constructed by their participants, so we propose a language that treats the world as essentially subjective. Our way of doing so introduces two fundamental postulates: (1) there is no reality without an agent, and (2) the agent only knows the world through actions. Based on these postulates, a formalism has been created that enables us to represent systems of social norms.

In this system, meaning is regarded as a relationship between sign and behaviour (more strictly, invariants in the flux of behaviour). Semantic analysis is an essential prelude to norm analysis. A new prototype implementation of this normbase is under construction. The goal envisaged is explained in terms of an illustration of how the normbase would be used to develop a system. At every stage of design, the specification is turned immediately into a default version of a working system. The application programmer can then concentrate on tuning this system or developing another that can perform the business functions in the most efficient technical manner. Experiments have already shown the viability of the concepts and methods incorporated in this normbase.

The authors are with the School of Management Studies, University of Twente, PO BOX 217, 7500 AE, Enschede, The Netherlands.

This research is now based at the University of Twente, having moved from the London School of Economics. In the past, funding has been received from IBM, the UK Science and Engineering Research Council, the UK Economic and Social Research Council and the UK Computer Board, but the work is now supported by DEC and the University of Twente.

Introduction

Since modern information technology was launched in the middle of this century, IT has been on an upward trajectory. The sky has been the limit of our expectations for so long that we are perhaps reluctant to acknowledge that the trajectory shows signs of turning down. We may soon come back to earth with a bump, at least where computers are used in organizations. Why?

1. A high proportion of systems put into development eventually never run productively.
2. Those that do run tend to absorb the lion's share of the information systems budget simply to be maintained as useful systems.
3. Systems frequently need to be surrounded by expensive bureaucratic procedures linking the formal IT system to the informal system of human operations.
4. Accompanying every system there is an expensive, poorly conceived and badly written body of documentation which also needs costly maintenance if it is to remain useful.
5. Increasing proportions of IT resources are having to be used to manage the complexity of the IT resources themselves.
6. Those responsible for procuring complex systems are finding the

demands for extra resources increasingly difficult to understand and are beginning to suspect that they are the subjects of a kind of technological blackmail.

The list could be extended but these points will suffice. What is the cause of such problems?

Our diagnosis is that the trajectory of IT is still largely determined by a technological imperative. We can agree that each new technology is wonderful. We can agree that there will be some use for it in organizations. What we do not accept is that, therefore, it *should* be used in organizations (the principle of the technological imperative). The basis of our objection is that IT, as many managers are learning to appreciate, can cause at least as many problems as it solves. Our suggested cure is to resist the technological pressure to use IT and instead make sure that IT is only pulled into the organization by genuine business needs.

We are not recommending a narrow cost-reduction approach. We are recommending the imperative of social or business value-added, especially as stated in organizational terms.¹ The normbase concept introduced in this paper is one way of making it easier to articulate our needs for IT, and so return IT to an upward trajectory.

The simplest history of our field would recognize the following major epochs:

1. The 1950s, dominated by *stand-alone systems*. These gave rise to the problems of expensive, repetitive data capture, solved in
2. The 1960s by *integrated systems*, which, in their turn, led in
3. The 1970s to *database systems*, which allowed the unexpectedly difficult task of data structuring to be solved and managed separately from applications programming.
4. The 1980s brought communications technology and the opportunity to correct the tendency of the DBMS to centralize, and so inaugurated *distributed structures*.

Our contention is that we are moving into a new epoch.

The 1990s may see us creating *normbase structures* that will complete the separation of data management from applications programming, extend the distributed systems concept, and also divide applications programming into parts that focus exclusively on knowledge of the organization (expressed by the norms of teams, firms, social groups or nation states) or upon knowledge of how to exploit the technology. The normbase concept should alleviate, if not cure, the problems listed above.

Inextricable mixing of business and technical knowledge

Very simply, a normbase stores, maintains and makes available, to both people and application programs, the 'laws' governing the pattern of behaviour in a group. The object of the norms and the computer system should be to support the functioning of the group. The business knowledge, which today is confused with technical knowledge within the program, would be a separate statement of what the organization requires. Behaviour, in so far as it is organized, displays regularity — this is virtually tautological. The regularities may be just a part of the

¹STRASSMANN, P.A. (1985). *Information payoff*. New York: Free Press. STRASSMAN, P.A. (1990). *The business value of information technology*. Canaan, Conn: Strassmann.

cultural system or they may be capable of being expressed explicitly as simple, written, company standards or elaborately drafted legal rules of behaviour with all degrees of complexity in between. The requirements to be imposed upon a formal bureaucratic or automated system consist of the explicit norms and a supporting semantic structure. The normbase we are developing would store such norms and semantic structures in order to provide, automatically, a default implementation of the necessary application programs needed to support the people in their work within the organization. The application programmer, if he were needed to intervene, would have the job of tuning the technical system to perform a fully specified business activity, with the desired degree of efficiency in the 20 per cent of the system where the transaction volumes are high. Through the normbase, the business specialists would create the complex, low activity 80 per cent of their application programs without involving application programmers.

More will be said later about the prototype normbase now under test and further development.

To illustrate the present situation, consider a company that grants to its young staff support for sports training. There are different age limits for receipt of these privileges in each company division. The government also has a scheme for day-release for study for similar age-groups. The government's and the company's own insurance and pension provisions have points at which young employees have to exercise a variety of options. One can easily imagine a constellation of different applications, clustered round a common database, dealing with notifications, requests for information from employees, production of statistics, and so on, that make use of conditions of the form

'if age less than x then . . . ' and
'if years_of_full-time_education greater than y then . . . '

These different conditions will be included in the programs for quite different business purposes, which may be revealed by the form of 'x' or 'y', if they are parameterized in a clear manner that fully represents their semantics. But *x* and *y* are typically known in the programme by such expressions as 'max-age' and 'min-fte', invented by a programmer. All too often *x* and *y* may just be numerical values inserted directly in the program in some cases, during hasty maintenance operations. Moreover the comparisons will be organized in sequences that take account of what is known of the efficiency of the resulting operations. If the government or the insurer or the company change their rather complex rules (not just a parameter) then the maintenance analysts and programmers will have to burrow through the applications programs, once they know what the business changes are. Their object will be to incorporate, at each relevant point in the program, the correct consequences of the new rule (note: not the rule itself) without seriously reducing the performance of the applications. It is easy to see why it is very difficult to unscramble the business knowledge and the technical knowledge in normal application programs.

There are many current trends in our thinking that seem to be leading towards the notion of a normbase. Let us consider them.

Trends towards the decoupling of organizational and technical knowledge

Databases are an essential step in this direction. They vindicate our

belief in the value of the principle of divide and conquer as a problem-solving strategy. By removing responsibility for the data and data-structures from the applications programmer, the database concept allows us to produce high quality records for the organization and, at the same time, simplify the task of programming. A by-product of the separation of data from application programs is the recognition by companies that the data constitute a key business resource that is independent of the technology. Unfortunately, the databases we develop do not separate the data fully from the technology.

So-called 'conceptual', and now 'semantic', schemas are a natural development from the idea of a database. The search for principles to guide the organization of data to suit many different users leads immediately to the notion of data as a representation of a world inhabited by those users. (For example see several papers in Falkenberg and Lindgreen².) The idea of having, in one place, a representation of a shared conceptual framework, related to the organization and quite distinct from knowledge of a technical kind, helps us to aim more confidently at the goal of a normbase that completes this separation between the organizational and technical knowledge involved. Unfortunately the problems of semantics raised by this train of thought are far from simple.

Knowledge-bases are relatively recent and they also build upon the principle of problem partitioning. Research in this field will certainly inspire prototype normbases, especially generic knowledge-bases not programmed with a specific problem domain and specific computing resources in mind. Knowledge representation — the key problem — cannot avoid raising the essential philosophical issues that must be faced to arrive at a normbase. Until recently, computer scientists were rather impatient of those who discussed epistemology, ontology and so on.³ But the problem of what is valid knowledge is a key practical issue for anyone using a database to run a business. It is one of our objectives to build the normbase on sound epistemological principles that provide greater assurance that the knowledge it provides has an assured level of quality. Present day knowledge engineering methods take for granted that the problems of epistemology are solved by the good intentions of the system designers. Unfortunately, there is a long way to go before we have a consensus on these vital but difficult issues.

A rule-base is a kind of knowledge-base and highly relevant to the study of norms. Norms are the implicit, sometimes explicit rules that govern the behaviour of groups of people. The methods of logic have speeded up our progress in the development of rule-base systems.⁴ Unfortunately, our logics have several defects from the point of view of anyone trying to account for social behaviour, rather than machine behaviour, and, of course, a normbase is preeminently an account of social behaviour.

The reservations associated with each of these trends in our thinking will be examined in more detail later. Fortunately, there is a way of eliminating them.

A change of philosophical attitude

The most important and most difficult step to take, in our opinion, is to make a change of philosophical position. Without this paradigm shift researchers will approach the analysis and design of organizational

²FALKENBERG, E.D AND LINDGREEN, P. (1989). *Information system concepts: An in-depth analysis*. Amsterdam: North-Holland.

³STAMPER, R.K. (1985). *Management epistemology: Garbage in, garbage out*. To appear in *Knowledge Representation and Decision Support Systems* (L. Methlie, ed.). Proc IFIP WG8.3 Conference, Durham, July 1984. Amsterdam: North-Holland.

⁴SERGOT, M.J., SADRI, F., KOWALSKI, R., KRIWACZEK, F., HAMMOND, P. AND CORY, H.T. (1986). The British Nationality Act as a logic programme. *Communications of the ACM*, 29 (No. 4).

systems with mechanical models in mind and continue the sterile tradition which regards an organization as just another machine, though larger and more unruly than the computer.

The technological imperative tends to force upon organizational information systems a structure that suits machines but not people. Of course, given enough coercion, people can be induced to behave as machines. That is how the 19th century treated most workers. That is the direction followed in centralized, authoritarian organizations. Up to a certain point and in special circumstances, it works. But when the situation ceases to be stable, so that Taylorian methods of work study cannot be used, when machines are an economic alternative to men for the routine tasks that remain, and when the driving forces behind the organization are not mass production at minimum cost, but instead, the drive to innovate and respond flexibly, then an authoritarian, mechanical or bureaucratic hierarchy is an obstacle.⁵ Intrinsic to working on technology is a way of thinking that sets one's mind in a philosophical mould where the irrationalities of people are unwelcome.

The post-industrial economy is already with us, generating products and services with high information content, even in the manufacturing sector. The days of simple mass production are behind us and success in the new economy now depends upon just those characteristics of innovative people — dynamism, continual innovation and flexibility — that were formerly suppressed to make repetitive mass production possible. Readers of this paper will, almost certainly, be information workers who pride themselves on having exactly those characteristics. As we begin to build systems to support our own organizational tasks — as opposed to supporting other people's technical work — we shall recognize the inappropriateness of treating people mechanically. So, if the philosophical arguments do not immediately register on some readers, sooner or later the practical demands of the information economy will restate the argument forcefully in the context of their own jobs, but perhaps too late for the understanding to be of use for them.

Bureaucracy is a typical organizational form associated with authoritarian, mechanistic structures. It uses people as machines. The work of the organization is analysed into small fragments and individual officers are given the job of administering small sets of rules that are applied to data which are entered on forms and kept in files of documents. Bureaucracy is a method of sharing a complex task among different people so that the level of skill required can be quite low. The individual officer is intended to need no critical understanding of the meanings of the words employed on the forms or in the rules. Provided that the formal processes are correctly carried out, he is in the clear. The essentially mechanical character of the work is exploited by some officers who learn the power of altering interpretations and regulating the flow of documents in exchange for favours, but these forms of corruption are essentially human behaviour that the system was trying to exclude!

If we want to design information systems to help human beings cooperate in essentially human ways, not as puppets on a bureaucratic stage, then these systems must facilitate the critical understanding of meanings and they must make use of, rather than suppress, human capacity to make interpretations. The formal systems are an aid to cooperation where they serve as an explicit framework of agreement supporting the informal system and supported by it. A normbase should

⁵ZUBOFF, S. (1988). *In the age of the smart machine, the future of work and power*. Oxford: Heinemann.

serve as a repository for the formal system but it should be designed in such a way that it is under the control of the appropriate human agents at each point where interpretation is needed. Our conception of a normbase is not as the ultimate knowledge-base containing organizational forms to be imposed on unruly people, but as a visible record of temporary consensus which the owners can continually criticize and change.

Cooperation in a team, but more so in a larger organization, depends upon having an agreed, explicit framework in common. A formally agreed set of terms, agreed standard procedures and a body of commonly agreed facts, expectations and plans are all important to allow individuals to act in a way that suits the achievement of shared objectives. Nevertheless, the formal structure must be within the control of the community. The meanings of the terms, the interpretation of the procedures, the confidence attaching to the facts and expectations and the priorities given to the plans can never be made fully explicit nor completely stable. Unless the members of the community can supply these finer features and constantly revise them at the level of their informal communications, the formal structure will become gradually more of a hindrance than an aid to team or company effectiveness. Without the continual, frequent, rapid exchanges of enquiring, critical commentary on the formal structure that test, confirm and sometimes alter its meanings and intentions, the users lose contact with and confidence in the mechanistic part of the information system.

A normbase should have the necessary features to support the interaction of the formal and informal parts of the information system.

The old position

When we learn natural science and mathematics at school and engineering or economics at university, we imbibe an old philosophy. These subjects are dominant in the education of most information technology specialists. We are led to believe in a quest for truth about a world that is essentially the same world for all of us: beyond the nose of every one of us (we are assured) lies the same objective reality. Our first academic task (we are taught) is to learn the established laws that describe that objective reality, and later to add to or improve upon that objective knowledge. In many respects, the tradition is a noble quest for 'the truth', and it has the advantage (perhaps doubtful) of suppressing a range of disturbing questions.

A glance at the literature of our subject will confirm that this view dominates. In the particular fields of databases, knowledge representation and logic, which are important to understanding the trends mentioned earlier, this philosophy encourages the unquestioning acceptance of a certain view about the nature of reality and our knowledge of the real world.

It is generally accepted that the world is comprised of objects. These objects can be assembled into sets. Ordered n -tuples of objects can be constructed and these can also be assembled into sets. Even more elaborate constructions can be imagined, and indeed that branch of mathematics, often taken to be the foundation-stone for the whole discipline, called 'set-theory', is concerned with devising and studying such structures.

Database theory accepts this position. The objects of interest here are

character strings which can be organized into tuples and those tuples arranged in sets, or relations. Indeed, in this context, the model of a world composed of discrete, identifiable objects is entirely accurate. But it is a totally artificial world, part of a machine. If you ask the philosophical question, the ontological question: what do you have to presume exists in order to construct a theory for database management? You can answer that we can confine ourselves to a study of characters, storage locations and a simple computing device. The subject is intrinsically machine-oriented, saying nothing about the world to which the character strings refer.

The theory of conceptual schemas and semantic databases takes the step of considering, or appearing to consider the difficult question of the meanings of the data. However, on close inspection, many of the data modelling methods used to construct schemas involve no explicit theory of meaning. They rely on the informal human system to answer all questions concerning the links between data and reality. When an explicit theory of meaning is introduced, it is almost invariably a naive interpretation of set theory. This assumes a world, exactly analogous to a database, composed of objects (which are the meanings of names) grouped into sets (the meanings of predicates) ordered in n -tuples which are grouped into sets (the meanings of relations). The structures generated by the designers of schemas, typically use concepts of entity, attribute, and relation corresponding to these structures but what constitutes an entity, attribute or relation is arbitrary.⁶ From this arbitrariness we can only conclude that the distinction is not one of importance in the world referred to. To overcome the problems of maintaining one arbitrary form of schema against the competition of others, the database structure has to be imposed by the authority of a data administrator and policed by his staff. Only an arbitrary outside authority can maintain these structures because there exists no overriding logic governing what place each item in the schema may occupy. Thus a semantic theory based on the idea that data items somehow point to individuals and sets of them in an objective reality will force us to embed it in an authoritarian social structure if its integrity is to be maintained.

Many theories of knowledge representation may be criticized in the same way but the growing unification of the database and knowledge-base fields makes it likely that these issues will receive increasing attention. What we need are principles that determine how the elements of a schema or knowledge structure should be related to represent reality, rather than merely to act as a convenient, navigable storage structure. The logical principles that can make the job of the data administrator redundant will be presented later.

Rule-based systems generally employ first-order predicate logic as their theoretical foundation.⁷ These are also in the same philosophical tradition that presumes a common objective reality as the basis of our knowledge. For handling social norms predicate logic is not sufficiently unsophisticated. A large number of logical rules without additional structure are impossible to understand and no basis for cooperation among a group of people. Moreover, the meanings of the predicate expressions appearing in the rules are surrounded by uncertainty. The predicates have to be constructed by the problem analysts. Different predicates may have the same meaning, e.g.,

x shall be eligible for y
the entitlement of x shall be y .

⁶STAMPER, R.K., BACKHOUSE, J., MARCHE, S. AND ALTHAUS, K. (1988). *Semantic normal form?* In: *Meaning: The frontier of informatics* (K.P. Jones, ed.). London: Aslib. THÖNISSEN, K. (1990). *Semantic analysis, a study of comparison*. Graduation report, University of Twente.

⁷KOWALSKI, R. (1979). *Logic of problem solving*. Amsterdam: North-Holland.

But who is to know that they have the same meaning? They may have been assigned by different analysts working on different parts of a large system. The inference engine itself will not make the connection. And it will be difficult to devise a program to uncover such potential similarities of meaning. The equality of meaning has to be registered by an informal human system that provides the essential semantic support. This example reveals another potential confusion: suppose we are concerned with eligibility for several different benefits under, say, several different insurance policies. The predicates will have to be long and complex sentential formulae, or they will have to be elaborated with many other variables to avoid semantic confusion, e.g.,

From date t a person x shall be eligible at a rate y for benefit b
under the policy p of insurer i as negotiated by company c .

This difficulty has been referred to as the ‘Humpty Dumpty Syndrome’ of analyst/programmer language construction⁸ — it is not a problem confined to predicate logic — it afflicts programmers in most languages. Rule-based systems, as presently conceived, can help us to automate bureaucracy but they do not enhance human comprehension of the tasks of a whole organization, which should be the goal of a normbased system.

A new philosophy

Our research, described here, is based on the premise that any group which is not simply a random set of individuals performs in a coherent way because its members tend to adhere to some system of norms. Some of those norms will belong to society at large or some other superordinate group, but some norms may be proper to only that group. When the norms have been made explicit, one may think of them as the ‘laws’ of that group. If one were able to state fully the norms of a group, then one would have a complete specification of that group as a social system.

Of course, no such complete statement of a system of norms is possible. Social systems are infinitely complex. For example, if a norm were to be disobeyed, how should the group respond? According to another norm, of course! And thus we start an endless chain of analysis. Similarly we cannot define all the terms used in writing the norms without embarking on an infinity of definitions. The analytical processes can stop when we reach a point at which we are confident of handing over the formal system we are devising by our analysis to the informal system, as represented by *responsible* individuals and subgroups. Complete analysis is then unnecessary.

The problem of meaning is most important if we are to have the norms correctly interpreted. As noted above, in the case of predicate logic, meanings are dependent on the interpretations given to individual names and predicate names by the person using them. (The truth-functional semantics within the logic does not help us to solve the practical issues of intentions because the extensional meanings of terms are unavailable in real situations. For example, what possible *operational* meaning is given to the word ‘red’ by defining its extensional meaning as the set of all red things?) In our system, the issue of meaning is handled by the assumption that the symbols within our formal system have to be translated (or translatable) into action by responsible agents,

⁸STAMPER, R.K. (1989). *The role of semantics in legal expert systems and legal reasoning*. Proc. Conf. on Legal Expert Systems, University of Bologna.

if they are to have any meaning. Thus, ultimately we have to indicate the agent who determines what that connection should be. The same expression can have, for different agents, different meanings. The meaning structure, as far as possible, has to be made as explicit.

Our language for representing the social knowledge of a group as a system of norms and the underlying semantic structures, called NORMA, is based upon two assumptions:

1. There is no reality without an agent.
2. The agent only knows the world through actions.

The agent may be an individual or a group. The world we know certainly depends on our actions, but in the flux of events and our own behaviour we find certain states to be of importance to us biologically or socially, and to some of these we have attached words. You will find every word capable of interpretation as the name of some invariant in behaviour.

The combined form of the agent and environment make possible ('affords'⁹) certain behaviours. (A certain larynx, sharpness of hearing, coordination of the two through years of skilful tuition, and the atmosphere on our planet, all contribute to affording Pavarotti top Cs that are the envy of other tenors.) The agent who realizes in his actual behaviour some invariant state becomes a modified agent, thus opening up the possibility of other behaviour. (A top C or other note can then have a crescendo, for example.) From these simple observations and the two assumptions, there arises an intrinsic 'logical' structure to behaviour that can be captured by analogy in a language, NORMA.

These two assumptions contain the kernel of a new philosophical position. The notion of an objective reality has gone, and in its place is a commitment to a subjective reality.

In this new paradigm, every view of the world is associated with some agent, usually a collective agent, a community that has evolved a shared vision of the world. Different cultures, different language groups, different professions, different organizations, even different teams within the same organization will use words with different meanings. Their worlds are indeed intrinsically different. 'Indeed' indeed, because their meanings depend upon *deeds* not other words.

A system developed on the basis of this position of radical, socially-based subjectivism allows for semantic diversity. Thus it respects one important aspect of the informal human information system. Can it do this whilst also providing a formal way of handling the many possible different meanings of the same expressions?

Semantic diversity can be tolerated if there is a theory of meaning to handle it. This is contained in the above assumptions. Meanings are not the possession of the words themselves, they have to be provided by identifiable agents. Unlike classical logical languages, you cannot introduce names and predicate expressions arbitrarily, to be understood by the analyst and whoever has understood him. Every word in the system has to be linked to a responsible agent (individual, group or role) and the meaning has to be explicable in terms of action, not just defined using other words. Different agents are entitled to their own interpretations. (Consider what 'customer' means to different people in a company — the market specialist, the salesman, the accountant, the company lawyer, the production engineer, etc.) Also, actions depend upon other actions and this gives rise to a structure that makes it easy to

⁹GIBSON, J.J. (1968). *The senses considered as perceptual systems*. London: Allen & Unwin.

disambiguate an expression used in many ways. A technique of semantic analysis is the product of this theory.

It must be pointed out that the details of NORMA are not fully worked out. As mentioned earlier, the philosophical issues arising in the design of a normbase are not trivial and it is they which are making a final solution difficult to find. The cause of the difficulty is that this language refers to reality not just to the artificial world of data elements, storage elements and operations on them. The difficulties are ones that have troubled philosophers for 3000 years, so we know that we shall have to be content with a rough approximation.

The normbase system under development

Taking these theoretical ideas which have evolved over a long period of working with concrete examples, we are implementing the fifth prototype of our system. It consists of three main components:

1. NORMA, a formalism for representing knowledge of group functioning, which allows meanings, norms and allocations of responsibility to be expressed and easily maintained.
2. LEGOL, a formalism for manipulating the NORMA knowledge-base, having the advantage of being able to use NORMA to disambiguate its expressions, and so having an unusual succinctness of expression.
3. MEASUR, a methodology plus methods for eliciting, analysing and specifying user requirements, in three main sets concerned with (a) articulating vague needs, (b) semantic analysis, and (c) norm analysis.¹⁰

They are intended to comprise a multi-user environment, in which participants can both develop the systems they want in an organic fashion, and conduct their formal relationships through the systems they have already in place.

This paper is not intended to report the work done, so much as to indicate a direction in which our community should be heading. Let us try to give you a picture of this process of organic system development that a normbase structure should make feasible.

From the users' point of view, they are being given a set of methods, MEASUR, for exploring their organizational behaviour. They will be seeking to establish, where necessary, a consensus about what they mean and how they are to behave. They will not attempt to provide an exhaustive solution (as indicated above, that would be futile) but they will put their decisions into their specification where they deem it necessary to be explicit and formal. They can assume that their common culture will fill many gaps but where that is unsure they may choose to be explicit. They could, in principle, do all this work using paper and pencil in order to arrive at a better organization. Most of their results will apply to parts of the organization that are not particularly orientated towards IT applications. But they can also use the results to define their business information requirement prior to a software engineering activity or the design of a bureaucratic administration. The specifications will be expressed in NORMA, for the semantic schema, or LEGOL, for the norms.

It is also significant to note that these methods support the dismantling of formal explicit systems. Other methodologies have a built-in

¹⁰STAMPER, R.K., ALTHAUS, K. AND BACKHOUSE, J. (1988). MEASUR: Method for eliciting, analyzing and specifying user requirements. In: *Computerized assistance during the information systems life cycle*. (T.W. Olle *et al.*, eds). Amsterdam: North-Holland.

ratchet effect that almost compels one to solve problems by introducing more formality as a precondition for more automation. Pushing back the tide of bureaucracy and formality can be accomplished, at least within the organization, by appropriate training, team formation and personnel management.

The normbase will support this activity, acting as a kind of CASE tool, except that software engineering is not necessarily involved. *Moreover, every step they take towards the definition of their organization will result in a default implementation of an IT system to support their activities.* This link is the one needed to make the organic growth of the information system possible. As they work, the size of their system can be estimated and thus the necessary resources assessed.

The goal envisaged

The group members specify how they want to operate as a group, regardless of any considerations of technology. They do so by establishing the semantics of their language and by defining the social norms they should obey.

Let us try to illustrate the process in the well-known case of a society that organizes international conferences (the CRIS case-study¹¹). The team includes members from headquarters, and representatives and chairmen of relevant committees.

Stage one: specification of terminology

Logical analysis: The team members work independently to specify the terminology of their task in a semantic structure, called an 'ontology chart' because it expresses the dependency of one structure upon another. This, the lowest degree of formalization will clarify the meanings in the language they use. In the course of this activity they will expose misunderstandings which they will have to resolve by negotiation and discussion.

(Example: Using the semantic analysis tool independently of one another, the lawyer from headquarters and a program committee chairman disagree about the meaning of *work* and its specific forms, *abstract*, *paper* and *report*. The PC chair thinks that his committee should determine the existence of a work but the lawyer, taking copyright law into account, advises that a work depends on the existence of a nation state, as recognized in its law of copyright, not on the existence of a programme committee. The solution shown in Figure 1 is the PC chairman's view but we could accommodate both meanings of 'work'. If experience and argument lead to a consensus in favour of the lawyer, the accumulated wisdom could be incorporated into a new standard, recommended specification for this kind of situation.)

Activity analysis: Some analysis of the dynamics of even this rudimentary system is possible. Each of the terms recognized will correspond (in conventional DB jargon) to an entity, relationship or attribute. But in this ontology chart, every entry (except 'IFIP') is a universal having numerous particulars, and every universal or particular has a start and finish to its existence. Focusing on particulars, we can record the expected numbers per item linked to its left, 'called its 'antecedent'. The duration of each of them can be estimated to provide a guide to the volumes of information to be handled.

¹¹OLLE, T.W., VERRIJN-STUART, A.A. AND BHABUTA, I. (1988). *Computerized assistance during the information systems life cycle*. Amsterdam: North-Holland.

Key

WG	Working group
WC	Working conference
PC	Program committee
OC	Organization committee
TC	Technical committee
(m)	member
Nat. rep.	National representative
IFIP	International federation of information processing

Figure 1. Semantics for a system to support the running of international conferences

Implementation: Even this rudimentary specification can be implemented immediately as a dictionary but it can also be converted to a schema for data that are available for storage, retrieval, processing, where that is appropriate. Although the handling of time in the normbase is not the focus of this paper, it should be mentioned that it functions as a temporal database in which every semantic element, either universal or particular, incorporates an existence period which is involved in the evaluation of all the operators, as indicated below under 'final stages'. (See Jones *et al.*¹² for an account of the time handling functions in the early Legol-2.0.)

Stage two: specification of responsibilities

Logical analysis: The next step is to add the structure within the group, specifying the teams, committees as well as the responsible individuals they will appoint. To all these agents they will allocate responsibility for the entities, relationships and parameters identified in step one. The locations of the agents may be added advantageously to the schema.

(*Example: In this case, the governing body of IFIP will be responsible for the existence of a new working conference, whilst the sponsorship will be under the joint authority of the working group and the working conference, and so on.*)

Activity analysis: Once again the dynamics can be examined in order to estimate how the work loads fall upon the various agents identified;

¹²JONES, S., MASON, P.J. AND STAMPER, R.K. (1979). LEGOL-2.0: A relational specification language for complex rules. *Information Systems*, 4 (No. 4). JONES, S. AND MASON, P.J. (1980). Programming the law. In: *Information systems environment*. (B.C. Lucas *et al.*, eds). Amsterdam: North-Holland.

minimum numbers of new staff required, or frequencies of committee meetings might be estimated. The messages arising at different locations can be estimated now.

Implementation: The required network can be added and exploited directly by the system if the addresses of the agents are available for electronic communications. Otherwise printed messages can be despatched through the mail.

Stage three: proto-norm definition

Logical analysis: Whereas the definition of who is responsible for any part of the system, provided by step two, is the weakest form of norm analysis, the third step creates what we call 'proto-norms'. These go one stage further but they do not lay down how the agents should behave except to indicate what information they should consider in deciding upon some parameter. These lists may be quite long and correspond to the inputs to a decision taken by some individual or collectivity in the fully evolved system.

(*Example:* The decision to start a new working conference will be taken by the governing body in the light of the sponsorship obtained, previous performance of conferences run by the relevant technical committees, the other conferences planned for the same year and the proposed location of the meeting. How these facts are used is never analysed; it will probably vary from decision to decision, and so, it is essentially informal.)

Activity analysis: The dynamics of the system will be revealed now, in rather more detail, at the level of the message traffic from the agents responsible for the inputs to those responsible for the decision parameters.

Implementation: The default implementation will now begin to support the agents by prompting them to find their required information, by indicating where to find it, and in some cases by furnishing the information from the database.

Stage four: specification of triggers

Logical analysis: The timing of the group's interrelated tasks can now be given greater definition by including the minimal conditions triggering the various decisions and actions.

(*Example:* Approval for a conference has to be considered at the next council meeting after the technical committee has submitted a proposal for the conference. Papers must be selected at least four months before the conference meeting, might be another trigger.)

Activity analysis: The dynamic analysis now begins to reveal a schedule of business activities and any problems inherent within it, reporting the critical areas of activity so that adjustments can be made by the team to their proposed organization.

Implementation: The specification at this level will be interpreted by the normbase as a DSS that will prompt decision-makers, supply them when required the information that it holds, request information in time for it to reach its users, or indicate sources of information not held in the database which the decision-maker must approach.

Stage five: specification of conversational rules

Logical analysis: The functioning of the system depends on communications among the responsible agents. Their communication acts are governed by culturally established norms. For example, Figure 2 contains a simple form of network showing the sequence of communication acts initiated by a request. These norms are a part of the common subroutines available to all systems. Supplemented by the local conditions governing when conversations are initiated, they provide a major increment to the logical analysis. Notice that these *formal* conversational patterns are *chosen* by the users to support their work, they are not *imposed* upon communications that would otherwise be regarded as belonging to the informal domain. (Problems in this domain are discussed in the context of the 'the coordinator' by Carasik and Grantham.¹³)

(*Example:* The national representatives of the sponsoring technical committees will be *requested* by the programme committee to *suggest* people from whom contributions should be *solicited*. In this illustration this conversational subroutine would be used twice: to request names and to request contributions.)

Activity analysis: The dynamics of the same 'conversational' routine will depend upon the context: making a contract in the currency market will be totally different from making a contract for the supply of a complex manufactured good, or agreeing to provide a paper for a conference. The activity analysis at this point reflects the known complexity and accepted behaviour in the problem area. More realistic estimates of staff requirements can be made.

Implementation: The interpreter of the specification will now evolve to include facilities to coordinate patterns of communication. This is particularly appropriate in office activities. Most of the routines would be based on the generally accepted social and commercial norms governing communication acts.

¹³CARASIK, R.P. AND GRANTHAM, C.E. (1988). A case of CSCW in a dispersed organisation. *CHI Proceedings: Human Factors in Computing Systems*. Special issue of *SIGCHI Bulletin*, Association for Computing Machinery.

Final stages: defining norms allocated to the computer

Logical analysis: Finally, the responsible agents can subdivide their tasks into those which they perform without computer assistance, those assisted by analysis of data, and those tasks (perhaps only fragments of

Figure 2. A simplified conversational protocol for making a request

them) which can be fully automated so that the IT system will evaluate the conditions *and* perform the actions. These are the kinds of norms that we embed in computer programs.

(*Example:* In the conference organization, it may be decided to compute priority ratings and the eligibility to participate in the meeting. Table 1 shows two such rules together with some definitions of roles and a generic/specific structure that actually belongs in the formation of the ontology chart.)

Activity analysis: We begin to define the computational requirement to support the group, over and above the operations of storage, retrieval and communications. The dynamics and the workloads associated with all this routine computing activity will be deduced. The need for a more efficient implementation than any that the default implementation can provide will be evident now.

Implementation: The implementation of the system through the norm-base ends here. Assuming a Pareto distribution, 80 per cent of the system will handle the most complex 20 per cent of the group's transactions. The 80 per cent of routine transactions will be handled by systems developed by more orthodox methods to meet the requirements specified by the semantic structures and the norms arrived at during the evolutionary process described above. Ideally, the normbase should be driven by the users working on their problem from a business point of view. The technical input will only be required at the 'hot spots' where the work generates too great a load for the technical system if not tuned by technical experts.

Current position

NORMA is not yet fully developed. Not surprisingly it throws up severe problems of a philosophical kind, bearing in mind the fact that it incorporates an attempt to build a logical system based on subjective

Table 1. *Norms for conference organization*

-
1. Priority #1 (member *or* contributor) *whenever*
 member (organizing committee *or while* programme
 committee) *or*
 contributor (selected paper)
 2. Contributor (paper) *then* author
 3. Contributor (report) *then* referee
 4. *For each* conference
 member (interest group *while* sponsor) *or* author *or* referee *while not*
 priority #1 (person), *then* priority #2 (person)
 5. Interest group
 whenever technical committee *or* working group
- Note: the operators all take account of time, as we have a temporal database;
or and *or while* are the same operator;
the *for each* operator determines the context in which the following expression is evaluated;
for each is equivalent to the *while* operator;
then and *whenever* are the converse of one another and replace the implication operator.
-

principles. For example, we normally think of a world consisting of individuals, but we have a world consisting of behaviour in which individual objects are just a special kind of repertoire of behaviours. The part/whole relationship presents difficulties, because in a behavioural sense there is no sharp (or perhaps any) boundary between an object and its environment. Time is a behaviour also, a behaviour that always depends upon our use of signs. A NORMA expression represents only what is here and now, and the past, the future and everything over the horizon have to be constructed here and now, using signs. Such problems illustrate the remaining difficulties in working out the theoretical side of NORMA. However, from a practical point of view, there is enough fully working now to demonstrate the effectiveness of the approach in a wide range of practical case studies.

It may appear that these philosophical questions cannot possibly have practical consequences. That is not so. The system currently under development has some interesting features that are their direct consequence. In particular, the meta-schema, schema, data and norms are all held in the same structure. (For speed of prototyping this will not be entirely so in the version under development.) Every record has exactly the same structure, so the database can be regarded as a single relation. This feature, of course, makes it much easier to create a good default implementation than would be possible if one had to support all kinds of unpredictable structures. A part of the standard structure is a period of existence bounded by start and finish times. This enables us to operate with no overwriting of data; only archiving is permitted. If you wish to delete items, then you are forced to recognize that the loss may have serious organizational consequences. The norms and the schema entries are also time bounded. Changes to the system do not affect the implementation, they only appear in the specification. The specification of the organizational system is all the 'program' needed. Hence the documentation of the application 'program' is greatly reduced and made easy to understand because it contains nothing but knowledge of the organization. These features go a long way towards answering the problems introduced in the opening of this paper.

Use of the approach in practice

The power of the methods incorporated in this system has been demonstrated in a series of experiments. Each of the various components has been tested separately. The whole suite of specification, processing and analysis modules has yet to be assembled.

One of the most interesting applications of the general principles of design outlined above has been made in the implementation of a significant part of a large administrative system on Wang equipment, in the University of Qatar.¹⁴ Initially, semantic analysis was applied and the resulting ontological structure translated into the form of an orthodox database schema, before building an orthodox database system to support the applications. As confidence in the method grew, it was decided to make no translation but to implement the ontological structure as directly as possible.

The new student registration system was implemented smoothly in this way in 1989. Unlike the normbase system under development now, the Qatar system has a relational database with two, rather than only one, relation. Most of the data, however, are held in the same form,

¹⁴ADES, Y.M. *et al* (1989). *Namat user manual*. Doha, Qatar (mainly in Arabic).

each tuple being a time-bounded 'propositional attitude' (in philosophical jargon). The change of a prescription into a fact, or 'fact' into an error, or an hypothesis into a plan, and so on, are all handled without recourse to overwriting data: archiving is all that is needed. The necessary functional modules have been implemented in a 4GL, not without difficulty. The results are interesting.

Development, maintenance, acceptability and control have all been improved. The carefully designed repertoire of functional modules has made it possible to implement without difficulty all the clerical procedures, even the most complex ones that apply only to very small quantities of data. Maintenance is simplified, because the high quality of semantic analysis allows the system to be evolved without disrupting the parts of the system already running. This also benefits development, as it is also possible to proceed incrementally in the early stages where change can more appropriately be called development, before the same work is taken over under the rubric 'maintenance'. Rigorous semantic analysis, which is expressed in a form that is easy to understand in business terms, has made the system highly acceptable to management, who were having difficulty in establishing a consensus on the managerial concepts that were needed in this institution experiencing explosive growth. For the systems professionals, the most obvious benefit is the shrinkage of documentation to the ontological structure plus a statement of the norms governing student registration and associated disciplinary statutes: applications program documentation is virtually redundant. Contrary to expectations, the fact that every clerical error is treated as a non-erasable record, turned out to be acceptable to the clerical staff. Errors can have social consequences, and it is easier to put the record straight if the problems are clearly linked to their causes. The impossibility of overwriting data ensures that there is a foolproof audit trail, and the powerful manipulation language makes it easy to apply any tests which the auditors require, especially as the system mandatorily establishes a responsible agent for all data. This experiment has demonstrated that the concepts are viable even in conventional IT environments.

Conceptual tools

Information systems are not IT systems — IT systems only handle data. Information systems entail human activity, at least to interpret the data and create whatever value they have. Designing and developing information systems can take place without using computers but using the same methods of analysis. What we are trying to create are ways of representing shared knowledge, opinion and judgements, in a form that is clearly visible to all in an unbiased way, supportive to all, open to critical appraisal, reinforcing the informal system and not a replacement of it. The normbase is essentially an implementation of a number of formalisms for sharing ideas relating to the users' problems or tasks. We should like these conceptual tools to be regarded as just as important as any software tool.

It is equally important to be able to use the concepts on the back of an envelope and in conversations about problems, as to be able to manipulate windows behind the screens of a software system. Among the many tests of the concepts have been experiments on the use of the analytical tools, divorced from technology.

Semantic analysis is a powerful problem-solving aid. We have used it in the drafting of complex rules, and one of the lawyers charged with drafting the principal UK legislation has judged it an effective means of quickly turning a novice into a competent assistant in the drafting process. Semantic analysis has been applied with success to disentangling basic concepts in difficult scientific problem domains, where it functions as a supporting tool for 'knowledge elicitation'. Those who have used semantic analysis notice that the structures they create tend to be portable into analogous situations, so we are beginning to see the possibility of accumulating, more effectively than before, problem solutions in consultancy domains. The key idea in semantic analysis is that of ontological dependency, which, although difficult to grasp at first, lends itself to 'back of the envelope' use in supporting conversations between problem-solvers.

Norm analysis (hardly mentioned in this paper) has proved effective at the informal level, when advice was sought on the problems of a large bureaucratic system. The problem owners were reluctant to reveal their highly political problems for discussion, although they wanted advice. The resulting catch-22 was evaded by a back-of-several-envelopes analysis of the organization's statutes, using the structures that are arrived at by the process of norm analysis. The result was a picture from which it was evident that political and industrial relations problems had been incorporated into the statutes themselves. These norm structures include among other things features which reveal the complexity that results from lack of trust and lack of informal contact and access to information. Understanding these, even talking round a flip-chart, can help a group to roll back the tide of bureaucracy.

Creating information systems should be about increasing the mutual understanding throughout a group, not just with automating their record-keeping and message-passing. Since the computer was introduced into organizations the tendency has been to impose the formal structures of technically orientated systems upon the informal system of human interactions. The normbase concept should turn the tables by allowing the inhabitants of the informal system to impose their chosen formality on the technical domain. There must be both formal and informal information systems and the link between them must be the interfacing language which ought to serve a double role. The interface for the Normbase does not pretend to be a natural language (no computer language can be and it is always misleading to suggest otherwise) but it carefully respects the differential semantics of the user groups and it provides a much friendlier data manipulation language than current 4GLs because it can exploit the semantics. We are confident that groups of non-technical people can use these facilities to grow, organically, their own systems.