

ASPECTS OF DATA SEMANTICS: NAMES, SPECIES
AND COMPLEX PHYSICAL OBJECTS

Ronald Stamper
London School of Economics
Houghton Street,
London WC2

ABSTRACT

Investigating the definition of administrative information systems in the LEGOL Project using legislation as experimental material, a new approach to data semantics is found necessary. The emphasis is upon the operations linking data to reality. The operations are performed in the Discourse System by people according to language norms, thus the social reality of natural language links the structures in the Formal System to the Object System. Data representing things in the real world may be regarded as subroutines in programs which people can interpret. This principle is being used to explore the possibility of a canonical data model based on an operational semantics. This is contrasted with other data models. To illustrate the principle, it is used to examine names and species, an aspect of the problem of universals and particulars. The results enable structures to be defined for the complex physical objects: collectives and systems. The use of an 'alias' function to express the operational identity of entities with different representations is introduced.

(The work is supported by UK Science Research Council with additional support from IBM)

ASPECTS OF DATA SEMANTICS: NAMES, SPECIES AND COMPLEX PHYSICAL OBJECTS

Introduction

This paper is another in the series explaining the concepts underlying the semantic model of the LEGOL language*. It is a sequel to a paper on the identifiers of physical objects (ref 1) which skated over the problem of universals and particulars. The treatment here is made self-contained by treating some conclusions of earlier papers without further argument.

The LEGOL Project is exploring the relationships between formal information systems and the world they represent and endeavour to control. As experimental material, the project uses legislation because, for example, a body of tax statutes in effect defines a large and complex formal information system. By trying to devise a formal language which can express whatever is in the statute, we are forced to explore the fundamental problems of systems definition at the highest level: i.e. saying what should be done without saying much about the procedures of how to do it. By testing each version of the language, as it evolves, against samples of legislation, well-founded progress is being made. These samples provide tests far more severe than do the synthetic examples usually conjured up to test theories of data modelling.

Program, data and human performance

The LEGOL language is interpreted by a computer**. Superficially it resembles a computer language but this is misleading. Only a subset of a body of legislative rules is sufficiently cut-and-dried to be interpreted without human judgement; LEGOL has to encompass both the mechanical rules and the exercise of judgement. LEGOL is therefore a language for specifying a formal organisation, not merely the computer programs used within it. It is a language for the systems designer from which programs (very inefficient ones) are derived automatically.

* The LEGOL Project is supported by the UK Science Research Council with additional support from IBM UKSC, Peterlee. Reports are obtainable from the author at the London School of Economics, Houghton Street, London WC2.

** Development of the first prototype system is complete and work has begun on the second.

Judgements are exercised by people whenever they report observations or perform specified actions. Whereas input to a computer system is achieved by the electrical or mechanical transduction of characters from one representation to another, the prime input to an organisational information system depends upon a person choosing a symbolic representation for what he observes or feels. Similarly, output from a computer system in the form of character strings only becomes an output from an information system when a command is obeyed, and this entails human judgement. Therefore a formal information system to help run an organisation depends both on the computer and upon human performance.

It is useful to think of the data which the computer holds as a means of 'programming' human behaviour. An entry on the list called DISTANCE is like a subroutine call with a precise operational meaning which will be modified by the context in which it is used. The entry may be in a typical LEGOL example:

20 miles, Lightship X, Portsmouth Point, 1952-

If this forms a part of the data base which is assumed to contain our relevant knowledge for a task, then it can supply an assertion to an enquirer in response to his request for information. This he may employ instead of making an independent observation. If he is sceptical he may treat it as an hypothesis to be tested by navigating between and making observations upon 'Lightship X' 'Portsmouth Point' and possibly some third location. In the LEGOL semantic analysis we construe the meaning of the data in terms of these operational procedures. The data make use of calls to various 'subroutines' in the sense of programs directing human behaviour. Thus 'distance' indicates a procedure for assigning a value jointly to 'Lightship X' and 'Portsmouth Point'. These two identifiers of the objects mentioned are also subroutines which permit a person to find two particular objects which are distinct from each other but which can be relocated repeatedly. The enquirer, to be able to make a precise operational interpretation of the data, has to be given the necessary 'code' for the subroutines and other bases for his actions. These include:

- knowledge of the language
- understanding of the problem context and
- motivation to respond

each of them being provided by the society in which he forms a part. Living in a community and solving the practical problems of day-to-day living, he acquires a set of norms of perception, evaluation, cognition and behaviour. Language does not enter into all these social norms but into a high proportion of them. Such are the language norms to which we shall refer in the rest of this paper.

Language norms should be a primary concern of any information system designer. If he creates a system which does not make appropriate use of the ways in which people interacting with the system have been 'programmed' linguistically, then that system is likely to function incorrectly. Unfortunately, the literature of computer science almost totally ignores the social foundation of language (e.g. refs 2-5). This is true even in the field of data modelling where the current search for a well-founded theory of conceptual schemas could profit most from this simple observation.

Canonical forms for data models

To understand the significance of the kind of semantic model being evolved in the LEGOL Project, it is worth considering briefly the quest for canonical data models. We should not expect there to be a single canonical form like a holy grail for a mathematical crusade: there should be several to serve different purposes. Each contains the minimal information for some purpose, subject to relevant constraints that make its expression irreducibly simple in that context. Some purposes are:

- (a) to specify access paths among data elements
- (b) to specify an information retrieval interface
- (c) to describe data movements
- (d) to characterise operational meanings.

The last of these is the purpose of a canonical data model for the LEGOL language. Let us examine each of them in slightly more detail.

Access structures will be equivalent in a fundamental way if they refer to the same data elements and if they embody the same pathways among pairs of elements. A binary relational structure is natural for this task and authors such as Senko (q.v.) have adopted this approach. A totally formal definition of meaning can be based upon transformations which preserve the structure. It may be possible to find certain constraints under which these transformations will result in a standard representation. It does not matter that this binary model requires the decomposition by artificial devices of relations that are 'irreducible' by criteria relevant to another purpose.

A retrieval interface should impose more structure than is needed to characterise access pathways. Codd's third normal form aims to give the user a view of his data which is irreducibly simple in respect of functional dependencies among attributes. This model moves away from the totally formal definition of meaning to one which is implicit and dependent upon intuition unless a purely empirical treatment of the functional dependencies is adopted. For example, that an employee has only one manager may be observed or it may be thought to be likely or it may even be prescribed; the model does not distinguish. There is an arbitrariness about the translation from real world properties to data representations but once chosen the representation can be transformed into this canonical form.

Data movements relate to messages and Langefors has developed a notion of 'elementary messages' to show how complex messages can be decomposed (refs 9, 10 and 11). The notion of meaning employed in this analysis is also intuitive. Langefors introduces 'elementary concepts' to construct his model. This explicit treatment of meaning (e.g. refs 9 p.229, 11 p.50), however, is conceptualist. The result is a data model appropriate if one is concentrating upon the flows of data in a particular organisation. The associated canonical form preserves the essential features of these data flows.

Operational meanings which underlie the LEGOL language are the patterns of behaviour which enable the data to 'program' the human user in an organisation. They embody distinctions which are organisationally significant (e.g. is 'to each man, one manager' a descriptive or a prescriptive constraint?) but computationally or procedurally irrelevant (see ref 12 for details). Given enough constraints of this kind, it should be possible to eliminate most of the arbitrariness of the information analysts' views of the data permitted by the approaches of Codd and Langefors. The goal we refer to as a 'semantic normal form'. The data in this model do not correspond to messages but to hypotheses and there is no conceptual treatment of meaning but an explicit operational one. This paper applies these principles to the cases of some complex objects.

Operational ontology

To understand the problem of the meanings of the data in a formal information system we need far more than intuition. Being concerned with organisational behaviour rather than data manipulation, the LEGOL Project forces upon the researchers a regard for the ways in which operationally (rather than intuitively) the data are linked to the real world. The use of legal material, by itself, makes one conscious of problems of evidence and judgement which are the key

aspects of an operational semantics. The data model which underlies the LEGOL language must express the implications of the data in terms of human performance. Ideally, given any datum in the system and the question 'what does it mean?' then there should be a route, through rules and decisions, back to the observations and value judgements and another route forwards to the actions that anchor that datum's meaning in reality. A standard way of doing this which enables one to check equivalence of meaning or to characterise precisely differences of meaning should lead to a semantic normal form. This is not a substitute for any of the other three types of data model described above. Each one entails concepts or constraints that are irrelevant to the others although they may overlap in other respects.

Universals and particulars

For the other data models described above, this ancient problem of universals (e.g. woman, beauty) and particulars (e.g. Grace, Jennifer) is not important. It may have some relevance for the message-orientated model which may use syntactic categories bearing these names, as a feature of its intuitive treatment of semantics. In the LEGOL data model, and any other concerned with operational semantics, the problem is fundamental. Obviously the operational processes of referring are different for universals and particulars. We have to say precisely how. This is easier if we confine our attention to what we say about physical objects. Abstract objects can be dealt with at a later stage.

As a background to the issue, let the main philosophical views on the problem be presented in an outline so brief it may seem like a parody*. The theories are generally about the relationships between mental concepts and external reality. The Realists assert that, just as concepts of particulars correspond to physical objects (Grace and Jennifer), so do universals (woman and beauty) correspond to ideal objects. For Platonic Realists these are the transcendental ideals (Woman and Beauty) which physical reality dimly reflects. For Aristotolean Realists the form of woman and the form of beauty are only immanent in the matter (Grace and Jennifer) with which they co-exist. Nominalists, on the other hand, reject the existence of real universals saying that these are only names which refer to particulars in a general manner in propositions. Objective Nominalists accept external particulars which cause the corresponding internal concepts. Subjective nominalists take only the concepts as real, the external world being a projection of them.

* For a still brief but more substantial summary see Lyons' useful book on semantics (ref 13).

These traditional views are deeply embedded in Western thought and one or other of them tends to inform any intuitive view of meaning (e.g. Platonic Realism in mathematics). They are all conceptualist in the sense that they treat concepts as one of the kinds of things being studied. Concepts, however, are not much use to an empirical scientist, neither are they much use to anyone designing an information system for running an organisation. As designers of systems to handle and share data in the public domain we should avoid a theory of data which needs access to the processes inside someone's head. This objection leads the LEGOL model towards what might be called an Operationalist theory of semantics which differs in its ontology from all those described above. This is a significant departure from the conceptualism which is implicit in most data base work today.

Operational view of universals and particulars

As pointed out earlier, the character strings in our data bases are linked to the external reality by people who use these data items according to linguistic norms. These norms depend upon the purpose for which the data are being used. (This was argued at length with copious illustrations in ref 1.) Linguistic behaviour is not mechanically uniform so that meanings, in the sense of signs signifying objects or states of affairs, will be rough and ready. If these variations hinder the performance of some practical task, or the resolution of a definite problem, they are too rough and will be adjusted by those involved, otherwise further precision is superfluous. The majority of data processing specialists seem to treat all data as though they had some definite meaning in terms of an external reality. (In this naive confidence in words and numbers lies the most serious social threat posed by the computer and its technologists!)

To illustrate the operational view of universals and particulars, attention is confined, at first, to physical objects. This is relatively safe ground where we can feel some confidence when asked to explain the meanings of our data*. The problem in this context we may narrow down to the use of names for individuals and for species. For the sake of brevity, let these be called 'names' and 'species'.

* However, even in the case of individual physical objects, the reality we know is partly a product of our use of language. As was argued in ref 1, the partitioning of the world into components bearing different labels is partly under the control of those labels and their use in a given problem context. (e.g. my son's new bicycle is a parcel as I carry it home but a vehicle when he rides to a friend and therefore subject to different laws on its two journeys).

Names and species

The words that we are talking about are used by people according to language norms which they acquire in a natural way whilst solving problems and performing tasks. The Formal System that we are interested in designing employs these words (or codes for them or other equivalent signs) to represent things in the external Object System. The linkage between the Formal and Object Systems is dependent upon the informal use of natural language in the Discourse System. To be effective, the signs in the Formal System must imply precisely (to the degree warranted by the practical problem) how a person in the Discourse System should locate an object in the Object System.

This is straight-forward in the case of individual physical objects. If the problem we are dealing with requires the use of an object with certain properties but does not require us to relocate precisely the same object, we can ask for it by using a species or common name:

a prawn a pipe section

Though generally a gourmet may not wish to know that the prawn he is eating was called 'Fred' nor the engineer that the pipe section being laid is '1234', these particulars are important to others. The prawn was called 'Fred' by the keeper of an aquarium from which it was stolen; his keeper was enamoured of Fred's quite distinct personality among the other prawns and fishes. The inspector, who originally failed the pipe section for use under high pressure after an X-ray check, was careful to report its exact code name. The way a species name is used will be learned, and there will be pressures both to retain and to alter the language norm: a chef who confuses 'prawn' and 'crayfish' will offend his client but a frozen food manufacturer who ennobles the shrimp into a prawn with enough breadcrumbs may make himself a great deal of money.

A species name enables a person 'programmed' by the appropriate language norm, to locate an individual which meets criteria appropriate for some practical task. Provided that the individual remains literally in the grasp of the finder, and it is treated in a way that prevents confusion with others of the same species, it may be given a local name, perhaps merely

the pipe section

and if there are several of the same kind, they may be distinguished by adjectives or pronouns:

my pipe section	your pipe section
mine	yours

all of which serve as local names of individual pipe sections.

A name enables a person to 'navigate' within the Object System and repeatedly return to the same individual object. For completeness, the notion of sameness can be defined operationally in terms of the continuity of a person's grasp upon the object, starting with simple cases (e.g. a book) and gradually generalising to more difficult ones (e.g. airport, star), whilst individuality is operationally defined in terms of the separation of the grasped object from others of the same kind. These operational definitions are learned and generalised during the acquisition of language. We may revert to the same type of instruction whilst training people in specialised skills relevant to an information system which we are designing*. A local name is easily constructed and it serves symbolically the same function of control as the grasping of the object. The problem of naming becomes severe when we put down the object among many of its kind. The local name is then useless. A name must be given in such a way that it is equivalent to a description of the individual, sufficiently precise to distinguish it from all others. In this process of name-giving, the local name will be employed. Legislation governing the registration of vehicles or of births includes rules for assigning names. A local name 'the vehicle' can be given during the registration procedure when an alias 'UNP 313F' is uniquely associated with that machine, at least in the wider context of the UK. When several individuals are distinguished locally, as in multiparous births, the physical 'grasp' on the individuals must ensure that individuality is established and maintained: in the delivery room and the nursery, careful management of the babies and their labels is necessary to maintain even the local names of Stamper 1, Stamper 2, Stamper 3.

The above account is exaggerated if it seems to suggest that species names are always established informally by norms and that proper names are assigned by formal procedures for which local names are the natural starting point. A few particular objects of sufficient importance may be named by natural norm formation among a group of people. Conversely, species names established by informal norms are not precise enough for many tasks and refinements have to be introduced by formal procedures. In all cases where formal procedures are used to establish proper or species names, recourse must ultimately be made to unsupported language norms.

The method of treating universals and particulars adopted as a semantic principle in LEGOL avoids the usual philosophical approaches sketched earlier. There is no need to talk of concepts if a strictly operational treatment of the question is

* A pertinent example is the design of an information system for computer-aided design of D.P. systems. The analyst must learn what is meant by the individual D.P. system and its sameness.

used, in keeping with the engineering approach of the project. It is possible to account for the paradox that language creates, to some extent, the reality that it represents, by fixing attention upon the ostensible phenomena of language as a social instrument for problem solving.

There are three ostensible phenomena. It is convenient to distinguish between them by using three types of brackets, if there is a danger of confusion. The two obvious ones are recognised by all authors dealing with the conceptual schema:

the sign used in the Formal System: e.g. <book>
and the physical book in the Object System: [book]

No mental object or concept need detain us. The link between these two is established by the phenomenon neglected in computer science:

the language norm sustained in the Discourse System: (book)

which is a pattern of behaviour and is itself ostensible. If we are pressed hard to explain the meaning of <book> we can avoid any meta-language other than demonstration by pointing at <book>, [book] and (book). Generally we shall find a common-sense use of natural language an aid to our explanation. However we must take care to recognise that the explanations have not a philosopher's purpose (some ultimate unravelling of metaphysical mysteries?) but the pedestrian object of building better information systems. Not until our explanations palpably fail in that problem context have we any need to revise them. This point has to be made lest it be thought that this is the beginning of a vicious infinite regress through layers of semantic analyses.

Elaborations upon the idea of physical objects

A semantic model to represent individual physical objects and their species must be elaborated to deal with complex physical objects such as collectives and systems and to extend to abstract objects. The same kind of semantic analysis must be applied. That is, each new data-construct should be treated as a 'program' interpretable by members of the social group upon whose language norms our meanings depend.

To direct a person, who has learnt the appropriate language subroutines, towards any individual of a certain kind we need give him only a value for

<species>

where the value is from a domain limited to the species names of objects. To direct him towards an individual he needs

<species>, <name>

where the name is a value domain defined within the context of the named species. A more specific species included within the species governing the name domain can be employed:

 <Swedish citizen> , <national ref. no>
 <citizen of Lund> , <national ref. no>

Also, we should note that the individual has associated with it, as a matter of ontological necessity, a period during which it is said to exist. That is, during which the 'programme' given to the person to locate the object is operationally effective. Thus

 <species> , <name> , <period>

constitute the parts of a data-construct which can represent a physical object. This structure may be subdivided into

 <identifier> :: = <species> , <name>
 <necessary attribute> :: = <period>

where the identifier provides a set of data values which correspond to the unique individual. When applied to the task of searching the real world, we refer more appropriately to the identifier of the individual whilst 'key' is reserved for locating a record about it. The period of existence is not required to find an extant individual but it is an attribute inseparable from any such entity and tells us when it would be fruitless to search for anything but reports of the individual.

Collectives

Often we wish to talk about rather amorphous objects such as a box of index cards, a stock of nails, a consignment of sugar or a roll of paper. These have many of the attributes associated with physical individuals but may lack coherent spatial features (e.g. a flock of sheep).

A collective is not merely a set of individuals nor is it so much as that. A set is represented in LEGOL by the values of the individuals:

```
( <species> , <name> , <period> )
  PERSON      TOM      1934-72
  PERSON      DICK      1918-69
  PERSON      HARRY     1940-78
```

If the members of this set change then the set changes. A collective, however, e.g. The Dairylea Herd of Pedigree Ayreshires, continues its existence whilst the membership changes. The important features of a collective are that its membership, though varying, remains of the same kind and that it has a focus, a collector or 'shepherd', as it were. The collective as a whole has a species which is not

the kind of its members. Any collector may have associated with it more than one collective of a given kind of individual, in which case it will need a name local to the collector and kind. Thus a representation for a collective may be:

```

CATTLE
( <species> , <name> , <collector> , <kind> , <period> )
  HERD      A      DAIRYLEA  AYRESHIRE  JAN 56 -
  HERD      B      DAIRYLEA  AYRESHIRE  MAY 62 -
  HERD      -      DAIRYLEA  FRESIAN    APR 48 -
  DELIVERY  123     DAIRYLEA  FRESIAN    JAN 78 -
  SHIPMENT  99      DAIRYLEA  AYRESHIRE  JAN 78 -

```

where 'herd' is the collective species word for a stock of cattle; we may imagine two herds of Ayreshires are run on different lines of management for experimental reasons whilst there is a single herd of Fresians (for which no local name is required); the delivery and shipment are other species of collectives which have distinct roles within the management of the live-stock; the existence of a delivery will cease once it has been merged into a herd, and of a shipment when it has been accepted by the customer, these rules being explicitly treated in the LEGOL formulation.

The identifier of the collective is given by:-

COLLECTIVE

```
( <species> , <name> , <collector> , <kind> )
```

in which the collector has a value drawn from a domain of identifiers of individuals. This identifier enables the collective to be referred to as though it were a simple individual object. The attributes <collector> and <kind> are also identifiers. As an aside, it may be of interest that, in the LEGOL language, the identifier attributes may be referenced by symbols in brackets following the entity name, e.g. HERD(FARM,ANIMAL), the <name> and <period> attributes remaining implicit. An unusual feature of the collective is that it draws one identifier, <kind>, from a set of universals not particulars. No other entity yet encountered has required this device.

One problem remains. What happens if the Dairylea Fresians are sold to the Archers' Farm? Does not the collector change and the herd cease to exist only to become another one? In practice the Archers would probably want the herd to continue to be regarded as the same as the old Dairylea herd because of its reputation and accompanying good-will in the dairy industry. The way this is handled is to continue using the value 'Dairylea' as the name of the herd. (Note that this has no implication for ownership, in a legal sense. Ownership would have to be represented as a relation quite independently.) Operationally, the collector is the link through which access to the members of the herd can be established. This

is now "Archers'" To record the continuity of the herd we can express the equivalence of names which have different periods of use:

IDENTIFIER. APR 49-MAY 78: (HERD, —, DAIRYLEA FRESIAN)

ALIAS. May 78 - : (HERD, DAIRYLEA, ARCHERS' FRESIAN)

Thus may continuity of an entity's existence and individuality be traced through changes of name or during the use of a multiplicity of names.

Systems

Systems are like collectives but they have additional structure: as a minimum, components with designated roles and probably individually named components. The components may be simple individual objects or other, complex ones.

The construction of a particular system may be arrived at, either analytically if one starts with an individual and then decomposes it into components, or synthetically if the individual components are identified before the system is constructed.

We also need to talk about systems generically especially in such applications as design or product specification. To define a system as a universal we must at least state the components' roles. For this we need

ROLE NAMES

<role> , <system>

FATHER FAMILY

MOTHER FAMILY

CHILDREN FAMILY

Whether the role were associated with an individual, a collective or another system would have to be indicated as additional structural information about the system. Using collectives or systems as components we are able to make it a requirement of a system definition that at any one time there should be assigned to each role no more than a single object. This will ensure that names can be assigned unambiguously via systems and their role names, as will now be illustrated.

To specify a particular family, the names of the individual members must be given. In the Family Allowances Act 1968 the family system is synthesised from specified individuals who satisfy certain conditions and relationships. The legislation does not deal with the vital administrative questions of how to name a family and when a family begins and ends its existence. To resolve disputes the legislation can rely upon the use of local names for dealing with each specific case. A semantic model for the administration of Family Allowances over extended time periods must handle names less informal. This naming problem concerns a synthesised

system which cannot be localised and perceived independently of its components (e.g. each member of a family may be in a different town yet still constitute a family). To solve the naming problem we must refer to the system via some integrally perceivable object. Normally this will be one of the components; for a machine assembly it would probably be the chassis but for a family in the context of the Family Allowances Act 1968 it is the mother, unless the family has only a male parent. The data structure for a specific system such as a family will include the values:

ROLE ALLOCATION

<species>	,	<type>	,	<compenent>	,	<whole>	,	<period>
FAMILY	,	FATHER	,	JOHN SMITH	,	MARY SMITH*	,	1960 -
FAMILY	,	MOTHER	,	MARY SMITH	,	MARY SMITH*	,	1960 - 1978
FAMILY	,	CHILDREN	,	CH(MARY SMITH*)	,	MARY SMITH*	,	1960 -

where probably the name of the family will eventually be a code number but it may initially be given as MARY SMITH*, using the name of a key number to refer to the whole system. The naming of the children as a collective (if we are only interested in the number of them and not individually) or as a system (if their individual identities are relevant) is also based on the system called MARY SMITH* as the collector or reference individual. Similarly for an analytically defined system the components would be named by taking the whole system as the reference individual.

In the table for the Smith family, Mary Smith ceased playing her role as mother in 1978 when in fact she died. The ontological problem posed by this change is whether the family continues or ceases to exist, becoming a new family based on John Smith as the reference object. The family, whilst having many physical features is also to some extent a legal fiction, being dependent on the rules defining it. Operationally there is no incontrovertable answer to questions of its continued existence. Rules must therefore prescribe when the family begins and ends. The rules in the Family Allowances Act establish that there is one family existing until 1978 called MARY SMITH* and another from that date called JOHN SMITH*. They might be treated administratively as distinct so that a new Family Allowances Pass Book would be issued to mark the change. However, the continuity of a group may be regarded as sufficient to preserve the family identity as a system despite the change of reference object. This, as in the case of a change of collector for a collective, could be established by a definition of an alias relation between the names:

```
IDENTIFIER.1960-1978:(FAMILY,MARY SMITH*)
ALIAS      .1978      :(FAMILY,JOHN SMITH*)
```

This again illustrates the interplay between reality and our use of language.

Conclusions

A radically new approach to data modelling has been introduced. It was used to show how the signification of a data value or a suitable group of values can be explained precisely in terms of the operations a person would follow to link the sign for an object, $\langle \text{object} \rangle$ and the real, tangible object, $[\text{object}]$. The linkage depends upon the relevant language norms which we may call (object). The method of analysis was extended from simple individual objects to complex objects such as collectives and systems.

Throughout the analysis it was necessary to draw upon the distinction between universals, such as species of objects, and particulars, such as individual objects. It was shown that the conceptualist, philosophical problems of universals and particulars are avoided by an operationalist view of meaning. In the LEGOL semantic model, particulars are data-structures linked to physical reality within certain operational or formal constraints. The same formal constraints are obeyed by abstract particulars. Universals are data-structures that establish domains of values and their formal interrelations linked to the social reality of language norms.

The practical applications of this analysis will become evident as computer-based systems in organisations become larger and more complex. The explicit and precise analysis of operational meanings will then be necessary to prevent automation from wreaking organisational havoc. The work dovetails with analyses of semantics at a different level which is exemplified by a paper of Biller and Neuholdt (ref 14). They examine the problem of demonstrating formal equivalence of data schemata but do so on the basis of a definition of meaning that they merely indicate. We feel that the treatment of meaning which is evolving from the LEGOL Project will provide the underpinning which data-base management scientists require.

Acknowledgements

I wish to thank my colleagues on the LEGOL Project - Susan Jones, Peter Mason and Paddy Mudarth for their help. In particular, Susan Jones deserves my thanks for many sessions of critical discussion on collectives!

References

1. Stamper, R.K. 'Physical Objects, Human Discourse and Formal Systems' in reference 5.
2. Klimbie, J.W. and Koffeman, K.L. (eds) Data Base Management, North-Holland, Amsterdam, 1974.
3. Nijssen, G.M. (ed) Data Base Description, North-Holland, 1975
4. Nijssen, G.M. (ed) Modelling in Data Base Management Systems, North-Holland, Amsterdam, 1976
5. Nijssen, G.M. (ed) Architecture and Models in Data Base Management, North-Holland, Amsterdam, 1977
6. Senko, M.E. 'The DDL in the Context of a Multilevel Structured Description: DIAM II with FORAL' in ref. 3.
7. Senko, M.E. et al 'Data Structures and Accessing in Data Base Systems', IBM Systems Journal 1973, 12, 30-93
8. Codd, E.F. Normalized Data Base Structure: a Brief Tutorial, IBM RJ935, November 1971.
9. Langefors, B. Theoretical Analysis of Information Systems, Auerback 1973 (4th Edition), Studentlitteratur, Sweden.
10. Langefors, B. & Sundgren, B. Information Systems Architecture, Petrocelli/Charter, New York, 1975.
11. Langefors, B. & Samuelson, K. Information and Data in Systems, Petrocelli/Charter, New York, 1976
12. Stamper, R.K. 'Towards a Semantic Model for the Analysis of Legislation', Informatica e Diritto (in press)
13. Lyons, J. Semantics Vol. I, Cambridge University Press, 1977
14. Biller, H. & Neuhold, E.J. 'Semantics of Data Bases: the Semantics of Data Models' Information Systems Vol. 3, No.1 pp11-30, 1978