



Information processing systems — Concepts and terminology for the conceptual schema and the information base

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work.

The main task of ISO technical committees is to prepare International Standards. In exceptional circumstances a technical committee may propose the publication of a technical report of one of the following types :

- type 1, when the necessary support within the technical committee cannot be obtained for the publication of an International Standard, despite repeated efforts;
- type 2, when the subject is still under technical development requiring wider exposure;
- type 3, when a technical committee has collected data of a different kind from that which is normally published as an International Standard ("state of the art", for example).

Technical reports are accepted for publication directly by ISO Council. Technical reports types 1 and 2 are subject to review within three years of publication, to decide if they can be transformed into International Standards. Technical reports type 3 do not necessarily have to be reviewed until the data they provide is considered no longer valid or useful.

ISO/TR 9007 was prepared by Technical Committee ISO/TC 97, *Information processing systems*.

The reasons which led to the decision to publish this document in the form of a technical report type 3 are explained in the Preface.

PREFACE.

=====

0.1. PURPOSE OF THE REPORT.

It is expected that future data base management systems will include a component for handling conceptual schemata. This Report explains the roles and concepts for a conceptual schema with the intention of providing a framework for discussion and for the design of conceptual schema languages. The rules described in the conceptual schema control to a large extent what may or may not happen in an information system and a data base. Therefore this Report is not limiting its attention to the conceptual schema alone, but also considers basic concepts for the mechanisms involved in manipulating a conceptual schema and a data base.

This Report is aimed at designers of information systems and data bases as well as suppliers of conceptual schema facilities. The provided framework will prepare the way for eventual standardization in the area of data base management. It does not, however, describe any particular method for using such facilities. In the meantime, the general principles in this Report can be used to evaluate emerging DBMS facilities.

The approaches and associated languages described in appendices to the Report are intended to be explanatory only and are not ipso facto candidates for a standard conceptual schema language.

0.2. STRUCTURE OF THE REPORT.

The main body of the Report (chapters one through four) contains the fundamental concepts and terminology for the conceptual schema, the information base, and the mechanisms involved in manipulating them.

Chapter one gives an introduction to the subject, mentions the origins of some ideas developed in the Report, and discusses some major topics. In particular, it explains what a conceptual schema is used for, its roles, and requirements for a conceptual schema facility.

Chapter two explains fundamental concepts, provides definitions of the concepts and terms, and develops some of the consequences of those concepts and definitions. Both static and dynamic aspects of the information system are considered and explained. Some readers may wish to skip this chapter on the first reading.

Chapter three discusses some aspects of implementation. In particular, principles are formulated for the contents and scope of a conceptual schema, and an information system architecture based on three levels is outlined.

Chapter four reviews some approaches to information modelling and manipulation for data bases. The approaches selected for illustration are outlined in more detail in appendices to the Report.

Several appendices have been added to the Report as follows:

Appendix A gives a glossary of the terms and definitions.

Appendix B provides an example situation to be described in information modelling approaches.

Appendix C gives a syntax notation to be used for defining grammars of example conceptual schema languages.

Appendix D outlines Entity-Attribute-Relationship approaches.

Appendix E demonstrates Binary and Elementary N-ary Relationship approaches.

Appendix F discusses Interpreted Predicate Logic approaches.

Appendix G elaborates on expressing dynamic rules and constraints in conceptual schemata.

Appendix H presents some thought on interacting with information systems and examples of permissible action descriptions.

0.3. STATUS OF THE REPORT.

This Report is an ISO Technical Report of type 3. It is the Working Group's first response to item 1 of its Program of Work. As such it is a statement of the Working Group's current view on concepts for conceptual schemata and information bases. Considering the rapid development in data base technology and applications possible, also taking into account the requirements for distributed data base systems and related data communication facilities, periodic revisions of the Report are to be expected.

0.4. REFERENCES.

- [1] MURRAY, J.A.H. et al.

'The Oxford English Dictionary' with supplements,
Clarendon Press, 1933 - 1977.

TABLE OF CONTENTS.

Chapter 1. INTRODUCTION TO THE CONCEPTUAL SCHEMA AND THE INFORMATION BASE.	9
1.1. The ANSI/SPARC framework.	9
1.2. The universe of discourse.	10
1.3. Describing the universe of discourse.	11
1.4. Static and dynamic aspects of a conceptual schema and information base.	13
1.5. Interaction between the real world and an information system.	13
1.6. The roles of users and information processors.	14
1.7. Guidelines for the description of a universe of discourse.	15
1.8. Guidelines for the contents of a conceptual schema.	16
1.9. Roles for a conceptual schema.	17
1.10. Requirements for a conceptual schema facility.	18
1.11. References.	19
 Chapter 2. FUNDAMENTALS FOR A CONCEPTUAL SCHEMA AND AN INFORMATION BASE.	21
2.1. General concepts and definitions.	21
2.2. Basic concepts and definitions for actions on the conceptual schema and information base.	28
2.3. The behaviour of an information processor.	31
2.4. Inserting a conceptual schema - the minimal conceptual schema.	33
2.5. Behaviour rules for the environment.	34
2.6. Static and dynamic rules and constraints.	34
2.7. Expressing rules and constraints.	36
2.8. Co-ordination of permissible actions.	38
2.9. References.	44
 Chapter 3. SOME CONCEPTS AND PRINCIPLES FOR IMPLEMENTATION.	45
3.1. Principles for the contents and scope of a conceptual schema.	45
3.2. Principles for the description of a universe of discourse.	47
3.3. Abstract syntax for a conceptual schema and information base.	50
3.4. Semantics of a conceptual schema and information base.	51
3.5. Principles for the composition of conceptual schemata.	51
3.6. The Three Level Architecture.	53
3.7. Information Resource Dictionary Systems (IRDS) Model.	58
3.8. The conceptual schema in the context of current DBMS implementation.	59
3.9. Correspondence of the Three Level Architecture for information systems and the Reference Model for Open Systems Interconnection.	60
3.10. References.	61
 Chapter 4. OVERVIEW OF SOME MODELLING APPROACHES.	63
4.1. Introduction.	63
4.2. Review of some approaches.	64
4.2.1. Entity attribute relationship approaches.	66
4.2.2. Binary and elementary n-ary relationship approaches.	66
4.2.3. Interpreted predicate logic approaches.	67
4.3. Translation of approaches to current data base technology.	68
4.4. References.	70

APPENDICES.

Appendix A. GLOSSARY OF TERMINOLOGY AND DEFINITIONS.	71
Appendix B. EXAMPLE UNIVERSE OF DISCOURSE.	77
B.1. Introduction.	77
B.2. Rules, etc. for the universe of discourse.	77
B.3. Some things and happenings in the relevant entity world.	79
Appendix C. THE PASCAL SYNTAX NOTATION.	81
Appendix D. THE ENTITY - ATTRIBUTE - RELATIONSHIP APPROACHES.	83
D.1. Emphasis of the approaches.	83
D.2. Primitive concepts of the approaches.	84
D.2.1. The basic concepts.	84
D.2.2. Abstraction concepts.	84
D.2.3. Characteristics of relationships.	86
D.3. Grammar and semantics.	91
D.4. Graphic formalism.	93
D.5. Modelling.	94
D.5.1. Some pragmatic modelling rules.	94
D.5.2. Formal rules for modelling.	95
D.6. Example conceptual schema.	96
D.6.1. Graphic representation.	96
D.6.2. Language example.	97
D.7. Check list for the conceptual schema.	99
D.8. Mapping of an EAR conceptual schema to a network data base schema and a relational data base schema.	102
D.9. References.	104
Appendix E. THE BINARY RELATIONSHIP APPROACHES.	105
E.1. Emphasis of the approaches.	105
E.2. Primitive concepts of the approaches.	107
E.3. Grammar and semantics.	111
E.3.1. The language and its relation to the universe of discourse.	111
E.3.2. Formal syntax.	113
E.3.3. Semantics.	115
E.4. Graphic formalism.	117
E.4.1. Linguistic object types.	117
E.4.2. Binary relationship types.	118
E.4.3. Constraints having a diagrammatic representation.	118
E.4.4. Some examples of the graphic formalism symbols.	119
E.5. Modelling.	121
E.6. Example conceptual schema.	122
E.6.1. Graphic representation.	122
E.6.2. Language example.	123
E.7. Check list for the conceptual schema.	131
E.8. References.	134

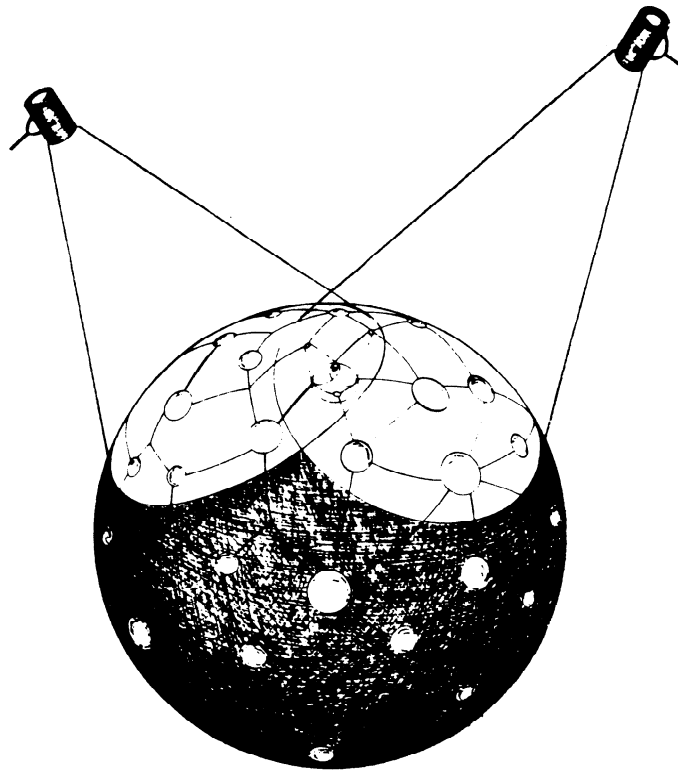
Appendix F. THE INTERPRETED PREDICATE LOGIC APPROACHES.	135
F.1. Emphasis of the approaches.	135
F.2. Primitive concepts of the approaches.	136
F.3. Grammar and semantics.	137
F.3.1. Abstract syntax.	137
F.3.2. Concrete syntax.	139
F.3.3. Semantics.	142
F.4. Graphic formalism.	142
F.5. Modelling.	143
F.5.1. Classification of axioms.	143
F.5.2. Constructs.	146
F.6. Example conceptual schema.	154
F.6.1. Graphic representation.	154
F.6.2. Language example.	155
F.7. Check list for the conceptual schema.	166
F.8. References.	169
Appendix G. EXAMPLES OF DYNAMIC RULE DESCRIPTION.	171
G.1. Introduction.	171
G.2. State-oriented descriptions.	171
G.3. State-oriented description of rules in the example conceptual schema.	171
G.4. State independent rules in action-oriented descriptions.	172
G.5. State dependent rules in action-oriented descriptions.	173
G.6. Action-oriented description of rules in the example conceptual schema.	173
Appendix H. EXAMPLES OF CO-ORDINATING PERMISSIBLE ACTIONS.	177
H.1. Interaction between environment and information system.	177
H.2. Some implementation considerations for permissible actions.	178
H.3. Describing permissible actions in the example conceptual schema.	179
H.4. References.	184

THE HELSINKI PRINCIPLE.

These utterances are to be interpreted (recursively) as international English utterances [1]:

Any meaningful exchange of utterances depends upon the prior existence of an agreed set of semantic and syntactic rules. The recipients of the utterances must use only these rules to interpret the received utterances, if it is to mean the same as that which was meant by the utterer.

(ISO TC97/SC5/WG3 - Helsinki 1978)



"THE METAPHOR OF THE SEARCHLIGHTS" on universes of discourse.

This page intentionally left blank

CHAPTER 1. INTRODUCTION TO THE CONCEPTUAL SCHEMA AND THE INFORMATION BASE.
=====1.1. THE ANSI/SPARC FRAMEWORK.

The reports of the ANSI/X3/SPARC DBSG [1, 2] identified the need for a conceptual schema in the context of a three-schema framework for data base management systems.

Subsequent papers [3, 4, 5, 6, 7] have emphasized the importance of a conceptual schema to users and designers of data base systems. In this context, a conceptual schema comprises a unique central description of the various information contents that may be in a data base. This includes the description of what actions, such as changes and retrievals, are permissible on the information content. The data base itself may be implemented in any one of a number of possible ways. Users and application programs may view the data in a variety of ways, each described by an external schema. Each external schema is therefore derived from the common conceptual schema. The physical storage structure that may be in use at any given time is described by an internal schema that is also derived from the conceptual schema.

The conceptual view, as meant by ANSI/SPARC, concentrates on the meaning of the information. It is the conceptual schema that describes this view. The external views concentrate on the forms - the data - that represent the information to the outside. These are described in the external schemata. The internal view concentrates on the internal physical representation of the data inside the computer system and is described in the internal schema.

Such a three-schema framework is widely, but not yet universally, accepted. It is assumed in this report. Furthermore, it may be noted that the conceptual schema concept is valuable in other environments than a three-schema framework.

It is widely acknowledged that the conceptual schema also plays a key role in systems analysis and data base design. One may therefore ask whether it should be biased to one or the other. Should the conceptual schema be primarily an enterprise model, resulting from the systems analysis, or should it serve as a focal point between user views and the physical data base design? We believe that it should play both roles in the next generation of DBMS.

We believe the data base user will benefit from the clear separation of the information meaning from the external data representation and the internal physical data storage layout. A clear methodology for producing a conceptual schema would help the implementor of an information system to improve his systems analysis, even if a manual step of translating it into data base design in terms of an existing DBMS were then required.

The ANSI reports introduced the conceptual schema in broad terms. Besides, the term "conceptual schema" is sometimes used for data base aspects which are not at all conceptual. Therefore, elaboration of the conceptual schema's objectives, roles, form, and content is needed. What a conceptual schema must include, which appropriate modelling concepts are to be used in it, and the exact role it plays in data bases, are the major subjects of this Report.

1.2. THE UNIVERSE OF DISCOURSE.

In the past, data processing systems were often designed so as to provide all users with the same set of capabilities or functions. However, this uniform functional view is not adequate to construct today's data base systems. A single data base may support quite different functional requirements concurrently, or at different times, during its existence.

The prime characteristic of the data base environment is that common data is shared between many users of a single system. By sharing common data, these users establish a dialogue with each other through the system. Clearly, if this communication is to be useful and reliable there must be some common understanding of the information represented by the data. Since it may happen that two users never meet, this common understanding must refer to something external to both of them. This common understanding must be recorded and in order to establish a dialogue a common predefined established grammar is needed.

We will call those things and happenings to which the common understanding of the represented information refers the universe of discourse. Universes of discourse may be concrete like an inventory, or abstract like the organizational structure of an enterprise. They even may be hypothetical like Wonderland which was visited by Alice.

In this Report we will take an (informal) naive realism approach to universes of discourse.

The typical universe of discourse is perceived as containing real and abstract objects, which we will call entities. It can be perceived as also containing classes of entities, e.g. persons, departments, and dates. This classification is based on similarity and takes into account characteristics common to several entities. The selection of characteristics for grouping the entities into classes is arbitrary; the choices will be made pragmatically, based on the purpose of the universe of discourse.

Some general properties to which entities adhere, that classify entities, that associate entities, etc., in the universe of discourse are also perceived (e.g. persons are not departments, a person may be assigned to no more than one department). These may be informally described as "classifications", "rules", "laws" or "constraints" about the state of affairs and behaviour of entities in the universe of discourse.

In general, what is considered to be part of the universe of discourse will be time-dependent, that is, the selected things and happenings may change with time. This will be equally true for the classifications, rules, laws, etc; however, it is likely that the rate of change of these will be relatively slow compared with that of the former.

1.3. DESCRIBING THE UNIVERSE OF DISCOURSE.

There are in fact two systems of interest: the universe of discourse and a data processing system which contains a linguistic representation of that universe of discourse. Following common usage we say that information about the universe of discourse "describes" or "models" that universe of discourse. We want, however, to emphasize that the description process may be in fact a very complex task calling for creative analysis and iterative refinement.

Without prejudging its physical representation we consider that the information contained in the data processing system describes the universe of discourse. A concrete physical representation of this information will be called a data base. We will use the term data base system for a data processing system dealing with a data base. It is possible for the data base system itself to be one of the subjects being described, in which case the data base system would be included in the universe of discourse. However, to simplify the discussion, we will generally assume that the data base system is disjoint from the universe of discourse, although this is not necessarily the case.

It is the classifications, rules, etc., that are of primary interest to a systems designer designing a data base system. In analysing the universe of discourse, it is these things he will want to identify, discuss with users and describe. In recording them he will actually create a "skeleton" description of the universe of discourse, the conceptual schema. In this way the conceptual schema describes which entities can possibly exist in the universe of discourse, that is, which entities exist, have existed, or might ever exist. In the same sense it describes what facts and happenings are possible for those entities or, if relevant, are required for them. We assume it will be held in a formal representation within the data base system.

We also want to record all other relevant information which describes the entities that are considered to be of interest and their actual state of affairs at a specified instant or period of time (usually "now"). We call this further information the information base.

Although each description necessarily will have a representation form to make the description communicable, it is the interpretation of this representation (the meaning of the description) which interests us in the first place. The representation form, although not irrelevant, is considered to be of secondary importance. We will use the term "information" when we want to emphasize our interest in this interpretation. We will use the term "data" when we want to concentrate on the representation forms of the information.

At this point it may be useful to consider the information describing a universe of discourse within the context of an ANSI/SPARC framework: We consider both the conceptual schema and the information base to be at the conceptual level, providing a conceptual view of the information about the universe of discourse.

The data base or parts thereof as seen by a user of the system (the strings of data) we consider to be at the external level giving an external view on the information about the universe of discourse. The internal storage forms within a computer we consider to be at the internal level being the internal view of the information about the universe of discourse. For the external and internal views the representation forms are of primary interest. The interpretation of those forms is, of course, the interpretation meant in the conceptual view.

Summarizing we have now identified:

o Universe of discourse:

The collection of all objects (entities) that ever have been, are, or ever will be in a selected portion of a real world or postulated world of interest that is being described.

o Conceptual schema:

The description of the possible states of affairs of the universe of discourse including the classifications, rules, laws, etc., of the universe of discourse.

o Information base:

The description of the specific objects (entities) that in a specific instant or period in time are perceived to exist in the universe of discourse and their actual states of affairs that are of interest.

Precise definitions for the above concepts will be given in chapter 2.

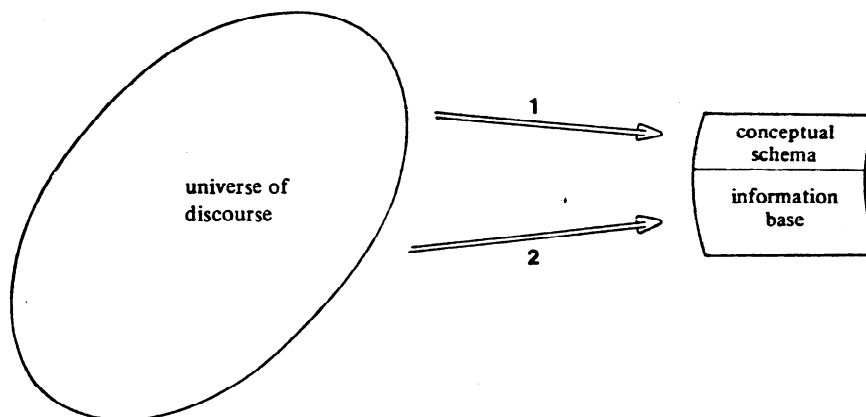


Figure 1.1. Describing the universe of discourse.

The description process is illustrated in figure 1.1; the two numbered processes are:

- 1: Classification, abstraction, generalization, establishing rules, etc, about the universe of discourse and recording them. This is a human process, describing a (shared) mental model of the universe of discourse.
- 2: Recording facts and happenings about the universe of discourse including what entities actually are of interest.

The conceptual schema describes the general rules, etc., of the universe of discourse, which, to a great extent, may govern its behaviour. These rules described in the conceptual schema therefore also control what may or may not occur in the information base. It is for this reason that we do not limit our attention to the conceptual schema and information base; we will also consider the mechanisms involved in manipulating the contents of the information base and the conceptual schema.

1.4. STATIC AND DYNAMIC ASPECTS OF A CONCEPTUAL SCHEMA AND INFORMATION BASE.

Much of the past work on concepts for the conceptual schema has been concentrated on the static aspects, that is, on defining the concepts to be used to describe valid states of a conceptual schema and information base.

However, the set of concepts for the conceptual schema should also cover the dynamic aspects. Firstly, the conceptual schema may change to correctly reflect changes in the selected portion of a real or postulated world. Secondly, dynamic aspects are involved in describing those manipulations which are needed to make known part or all of the conceptual schema and information base.

In some cases, the time scales of changes within the universe of discourse and the corresponding changes in the conceptual schema and information base need not be tied closely together: changes in the universe of discourse may be recorded in the conceptual schema and information base retrospectively and even in a different sequence. In other cases, the time scales are so closely related to each other that the conceptual schema and information base necessarily become part of the universe of discourse; especially in these cases the description of this interaction must also be part of the dynamic aspects.

No clear boundary has been defined between static and dynamic aspects, and the boundary may well be found to vary between different approaches or even to be non-existent in some cases. Some of the ideas introduced on this subject in the present Report have not yet been the subject of wide debate, but may at least serve to indicate areas deserving further study. In particular it is not clear whether different sets of concepts should be used to describe static aspects and dynamic aspects, or whether, at least for some approaches, the same set of concepts may fulfil both purposes.

1.5. INTERACTION BETWEEN THE REAL WORLD AND AN INFORMATION SYSTEM.

A conceptual schema and information base is totally static unless something operates on it to cause change. That something we will call an information processor. We will define an information system as consisting of a conceptual schema, an information base, and an information processor.

An information processor operates to produce change in the information base or conceptual schema only on receipt of a message. A message contains information and/or expresses commands. Messages originate from a part of the real world referred to as the environment, which may be disjoint from, or overlap with, the universe of discourse. On receipt of an appropriate message containing a command an information processor may also operate to make known, by means of a message, information present in the conceptual schema and information base. For further details see chapter 2.

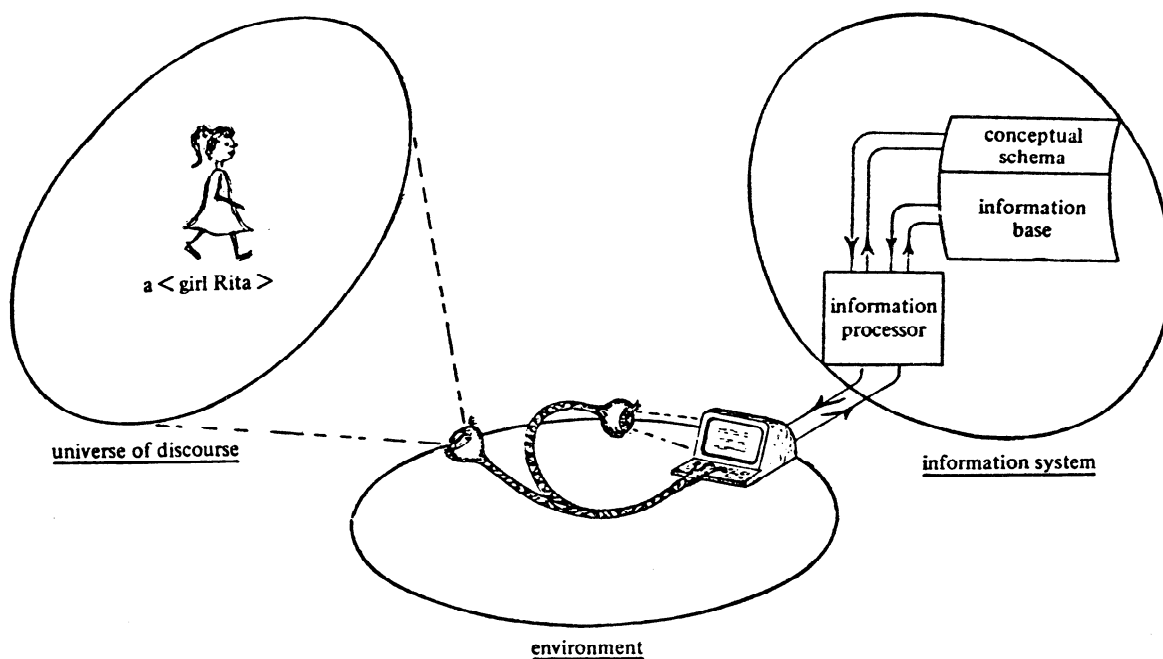


Figure 1.2. Information system and environment.

The information system is distinguished from the environment in the following way:

- o The information system is a formal system, the environment as a whole is not so.
- o The behaviour of the information system is completely defined by behaviour rules and constraints which are established, directly or indirectly, by the environment. The information system on its own initiative never establishes rules for the environment.
- o An information system, being fully predictable, is unable to deviate from the rules or constraints. The environment can deviate from its rules.

Although we may consider the information system together with the environment to be parts of an encompassing system, this latter system may not be formal or fully predictable. Therefore we use the term information system as above, excluding the environment - the users of an information system.

1.6. THE ROLE OF USERS AND INFORMATION PROCESSORS.

The users of an information system can be machines or other systems as well as human beings. A user is anybody or anything that issues commands and messages to the information system and receives messages from the information system. As such they are part of the environment. Some users also have the authority to establish behaviour rules or constraints for the information system.

An information processor transfers messages between the environment and the information base or conceptual schema, as explained above. In doing this it has no initiative of its own; it can only behave exactly as specified by the rules, the whole rules, and nothing but the rules.

Normally an information processor will be a computer system or some parts thereof, but human beings can also play the role of an information processor, provided they do not deviate from prescribed rules or act on their own initiative. Computer systems, on the other hand, can act as users of an information system. An example is a network of information systems communicating with each other. If each has a set of rules which is independent of the others, then each plays the role of user of the other information systems. We therefore conclude, that the role determines whether something must be regarded as a user or as an information processor.

The above formulation of users and information processors in terms of roles implies that the environment and the information system need not necessarily be disjoint. Similarly, if the information base contains information about the users of the information system, the environment and the universe of discourse will not be disjoint. However, even if they are not disjoint they will always be distinguishable from each other.

1.7. GUIDELINES FOR THE DESCRIPTION OF A UNIVERSE OF DISCOURSE.

Sometimes in the literature on various modelling methods for information systems and conceptual schemata, no clear distinction is made between the things and the description of the things, nor between the information meaning and the data representation.

This stems partly from the origins of some approaches which are in effect the data modelling techniques of the early seventies. Partly, however, the reason for this is a debate on fundamentals, which is still going on. The difference is found in whether the conceptual schema must be defined in terms of entities in the universe of discourse itself and states of affairs about them, or in terms of descriptive constructs found in the information base describing the entities of interest in the universe of discourse. Either view is possible and can be presented systematically.

It is most important to note though that the two alternative views above cannot be indiscriminately mixed in the same discussion without leading to confusion, paradox, and error. It is unfortunate but true that many variants of modelling approaches, both in practice and as described in the literature, suffer from precisely these problems.

The current work of WG3 is based on the assumption that the conceptual schema and the information base should describe the conceptual view. This implies that the conceptual schema is defined in terms and constructs referring to things in the universe of discourse itself and expressing states of affairs about those things.

The constructs used in a conceptual schema and information base should be based on the fundamentals of formal logic as theoretical foundation. However, how closely they must follow the spirit and notation forms of these fundamentals is a subject of further investigation. It is quite possible to limit them to elementary constructs expressing these fundamentals. However, it will be

always possible to define, upon these fundamentals, a variety of more complex constructs ("macro constructs"), that may be more convenient or efficient for describing various aspects of a universe of discourse.

The choice of specific macro constructs is based on practical arguments such as ease of understanding and use. That choice is considered to be dependent on the application area of the information system for which a conceptual schema and information base has to be provided.

As already stated, it is important to distinguish carefully between the entities and their descriptions. In these descriptions entities are usually identified by names that refer to those entities. This includes synonyms - different names referring to the same entity - and homonyms - identical names referring to different entities. The relevance of this distinction, not only for information systems in particular, but for human communication in general, has been well-known in language philosophy and linguistics for a long time. Therefore the constructs should provide for synonyms and possibly cope with homonyms.

The conceptual schema not only describes the static aspects and dependencies of the universe of discourse, but also the dynamic aspects. This determines what manipulations of the descriptions are allowable as well as what descriptions may be present in the conceptual schema and information base. Therefore it may be clear that constructs have to be available both for the descriptions and for their manipulation in the information system.

The subject is elaborated further in chapter 3.

1.8. GUIDELINES FOR THE CONTENTS OF A CONCEPTUAL SCHEMA.

Since the selection of what are considered to be general classifications, rules, etc. of the universe of discourse is to a certain extent arbitrary, it follows that the choice of which should be described in the conceptual schema and which in the information base is arbitrary to a similar extent. In practice, however, the systems designer might consider various factors in deciding the boundary of the conceptual schema. These might include:

- describing classes (types, variables) in the universe of discourse rather than individuals (instances),
- describing concepts that are less subject to change rather than concepts that are changing more frequently,
- inclusion of rules or constraints having wide influence on the behaviour of the universe of discourse (and therefore on the behaviour of the conceptual schema and information base) rather than narrow influence.

At all times the following general principles for the conceptual schema should be observed:

* 100 Percent principle:

All relevant general static and dynamic aspects, i.e. all rules, laws, etc., of the universe of discourse should be described in the conceptual schema. The information system

cannot be held responsible for not meeting those described elsewhere, including in particular those in application programs.

* Conceptualization principle:

A conceptual schema should only include conceptually relevant aspects, both static and dynamic, of the universe of discourse, thus excluding all aspects of (external or internal) data representation, physical data organization and access as well as all aspects of particular external user representation such as message formats, data structures, etc.

A more detailed discussion may be found in chapter 3.

1.9. ROLES FOR A CONCEPTUAL SCHEMA.

A fundamental impact of a conceptual schema is that the concepts used harmonize - and to a certain level make possible - human communication. Moreover, these concepts will influence the methods and results of analysing organizations and their information needs. In a way, a conceptual schema constitutes a general agreement concerning how to perceive a universe of discourse. This agreement may alter over time, but supports the evolution of applications over their life cycles as well as changes of this agreement itself (cf. The Helsinki Principle).

A conceptual schema is intended to properly describe the behaviour of a universe of discourse. Therefore the rules given therein, naturally, restrict possible evolutions and manipulations of the description of the universe of discourse, i.e. of the conceptual schema itself as well as of the information base.

Mainly for reasons of human convenience and efficiency, different users within the environment of a common information system will use different forms of external data representing the information. At the same time, for reasons of machine and storage handling efficiency, internal data organizations will be designed and used that may or may not differ from those external forms. In this context, the conceptual schema enforces preservation of meaning in transformations between the various data representations and defines the interpretation of these representations.

Therefore, and considering what is outlined in the previous sections, the following fundamental roles for a conceptual schema have been identified:

1. To provide a common basis for understanding the general behaviour of the universe of discourse;
2. To define the allowed evolution and manipulation of the information about the universe of discourse;
3. To provide a basis for interpretation of external and internal syntactical forms which represent the information about the universe of discourse;
4. To provide a basis of mappings between and among external and internal schemata.

1.10. REQUIREMENTS FOR A CONCEPTUAL SCHEMA FACILITY.

It is anticipated that the data base management systems developed in the future will include a component for handling a conceptual schema definition which will fulfil the roles mentioned in the previous section. In the course of time, provision of such a component should become a standard requirement.

To fulfil the roles indicated above, a conceptual schema facility must satisfy the following requirements:

1. It must provide basic concepts which are suitable for adequately describing both the static and dynamic aspects of a universe of discourse and ipso facto its description in terms of a conceptual schema and information base.
2. It must provide a language in which a conceptual schema can be expressed so as to be readily understandable to a user of the facility.
3. It must provide a language for precisely communicating a conceptual schema to a computer.
4. It should provide for easily modifying the conceptual schema to reflect changes in the general classifications, rules, laws, etc. of the universe of discourse, and for predicting the direct consequences of such changes.
5. The views of the information that different users wish to see are limited to those which do not contradict the assertions in the conceptual schema. If such external schemata are subject to change, the facility should be such that this should not affect the conceptual schema.
6. The conceptual schema should be kept invariant by the facility with respect to changes in the internal (physical) representation of the data within a computer.

The two languages mentioned in 2 and 3 may be the same but are not necessarily so. For the former purpose an additional graphic notation may be helpful.

These six requirements as a minimum must be met by any candidate conceptual schema facility.

1.11. REFERENCES.

- [1] ANSI/X3/SPARC, `Study Group on Data Base Management Systems:
Interim Report 75-02-08`,
In: ACM SIGMOD Newsletter, FDT, Vol 7, No. 2, 1975.

- [2] TSICHRITZIS, D., and KLUG, A. (eds.)
 `The ANSI/X3/SPARC DBMS Framework. Report of Study
Group on Data Base Management Systems`,
 AFIPS Press, Montvale NJ, 1977.

- [3] BSI/DPS13/WG1, `Role and Composition of a Conceptual Schema in an
Overall DBMS Architecture`,
 1976.

- [4] AFNOR/Z6/SC5, `Study of Users Requirements`,
 1977.

- [5] NIJSEN, G.M. `The Next Five Years in Data Base Technology`,
In: Infotech State of the Art Conference, London,
 1977.

- [6] NIJSEN, G.M. `A framework for advanced mass storage applica-
tions`,
In: Medinfo 80, Proceedings of the Third World
Conference on Medical Informatics, Tokyo 1980,
North-Holland Publishing Company, 1980.

- [7] FALKENBERG, E. `On the conceptual approach to data bases`,
In: International Conference on Data Bases,
Aberdeen 1980, Heyden and Son, London, 1980.

This page intentionally left blank

CHAPTER 2. FUNDAMENTALS FOR A CONCEPTUAL SCHEMA AND AN INFORMATION BASE.

=====

2.1. GENERAL CONCEPTS AND DEFINITIONS.

We propose in this Report concepts and terms to be generally used in designing, describing, and using conceptual schemata and information bases. Some of the terms defined in this Report are already found throughout the data base literature, occasionally with conflicting meaning. The purpose of this chapter is to describe the fundamentals and to give short, precise, although intentionally informal, definitions of the concepts and terms, wherever possible conforming with the meaning most closely associated with natural language [1]; e.g. the term "real world" is to be interpreted in ordinary language sense. The definitions are embedded in explanatory prose to inform the reader of the basic concepts and intent of our view of the conceptual schema and information base. The definitions themselves will be summarized in a glossary of terms in appendix A to this Report.

We start by assuming that it is possible to select a portion of a real or hypothetical world that is describable in some chosen precise and formally defined language. All things we perceive or assume to exist in this selected portion of a world are called entities:

ENTITY

Any concrete or abstract thing of interest, including associations among things.

For example, if we select a portion of a world as described in appendix B and in which a certain Registration Authority is assumed to be interested, then entities are the car Ford Mustang PCXX999, the person Mr. Johnson, the date 29 January 1975, etc. A particular example of an abstract entity is an association among other entities, e.g. the "ownership" of the car PCXX999 by Mr. Johnson.

In perceiving or imagining the selected portion of a world we are interested in we may conceive all kinds of states of affairs concerning one or more entities therein. Examples are:

- The car PCXX999 is of model Mustang
- The car PCXX999 has got registrationnumber GMF 117
- The car PCXX999 is distributed by Ford to Smith's garage on 29 January 1975
- Garages sell cars to persons
- Ford is a car manufacturer

We call such states of affairs propositions:

PROPOSITION

A conceivable state of affairs concerning entities about which it is possible to assert or deny that such a state of affairs holds for those entities.

A proposition can concern one entity, several individual entities, groups of entities, etc.

In practice the distinction is often made between propositions about the actual state of individual entities, and propositions about which behaviour of entities may or may not be permissible or possible. The words "rule" and "constraint" refer in particular to propositions of this latter kind.

Actually it will be descriptions of the propositions - sentences - that enable us to discuss entities and their states of affairs at all - that is, to exchange information about entities by describing propositions which hold for them:

SENTENCE

A linguistic object which expresses a proposition.

LINGUISTIC OBJECT

A grammatically allowable construct in a language.

Note, that linguistic objects may be considered entities.

Sentences consist of terms and predicates:

TERM

A linguistic object that refers to an entity.

PREDICATE

A linguistic object, analogous to a verb, which says something about an entity or entities to which term(s) in the sentence refer.

For instance the sentence:

"The car PCXX999 is of model Mustang."

expresses the first example proposition above. In this sentence the verb "is of" formulates the predicate. The terms "the car PCXX999" and "model Mustang" refer to the involved entities.

Often various sentences convey the same information, and, in particular, different terms may refer to the same entity. For example, the term "Mary Jones" is evidently different from the term "Mary Smith". Nevertheless, after Mary Jones has married John Smith, both terms will be associated with the very same girl. Thus, the sentences

"Mary Jones was born in 1955"

and

"Mary Smith was born in 1955"

have the same meaning, and it is evidently the girl who was born in 1955, not the term! The notion of similar information conveyed by different sentences, in many cases resulting from alternate ways to refer to entities, has tremendous importance for flexible and unambiguous communication.

Some terms are simple linguistic objects, as for instance the terms in the above examples. In other cases, however, more complex linguistic objects may be used as terms. In the sentences

"Ford produces the car PCXX999."

and

"The manufacturer that produced the car PCXX999
distributes the car PCXX999 to Smith's garage."

the terms "Ford" and "the manufacturer that produced the car PCXX999" are two terms that refer to the same entity. In this example the first term is only referring to the Ford company. The second term also refers in a certain way to a proposition about the production of a car.

Some linguistic objects play no other role in the descriptions than to be used as names for something else. We will call them lexical objects or names:

LEXICAL OBJECT or NAME

A (simple) linguistic object that is used only to refer to an entity.

In normal cases a lexical object consists solely of one or more nouns.

The special kind of association between the "basic" entities and the lexical objects that refer to them we could call a naming convention. When such a naming convention between an entity and a lexical object is correctly established it is always possible, at least in principle, to identify a causal chain to the use of that lexical object from an instance of "name giving" in the world, i.e. a point in space and time where an appropriate action was taken that asserted, in effect: "Henceforth this entity will be called by the name (i.e. lexical object) so-and-so !".

It is part of the information system designer's job to make sure that all entities of interest can be referred to in some way. For this reason, the information system designer will generally wish to additionally describe, in the conceptual schema and information base, the commonly agreed ways to refer to entities.

It should be carefully noted, that there is no barrier and, indeed, often considerable utility in the same entity having more than one lexical object associated with it. These lexical objects then are synonyms:

SYNONYMS

Different terms that refer to the same entity.

Fundamentally there also is no barrier in several identical lexical objects being associated with different entities. These lexical objects then are homonyms:

HOMONYMS

Identical terms that refer to different entities.

In practice, they may cause some ambiguity. Therefore in some information systems homonyms are excluded. However, this exclusion is certainly not a fundamental or necessary requirement, provided some mechanism exists to resolve ambiguity.

Often we will be interested in entities that are considered to occur together:

ENTITY WORLD

A possible collection of entities that are perceived together.

For example, all cars, registered by the Registration Authority, and all manufacturers, garages and persons involved with those cars, as described in appendix B, may be considered to form an entity world.

Many different entity worlds can be discerned at the same or at different times. Also, an entity can belong to many entity worlds.

A collection of propositions asserted to hold for a given entity world is called a proposition world:

PROPOSITION WORLD

A collection of propositions each of which holds for a given entity world.

A collection of sentences that express the propositions of a proposition world informs us about the relevant entity world.

Our selected portion of a real or hypothetical world involves all possible entities we are interested in. These are the ones we want to discuss:

UNIVERSE OF DISCOURSE

All those entities of interest that have been, are, or ever might be.

The universe of discourse might alternatively be called the universe of possible entities. Note, that the universe of discourse is limited to the possible entities we are interested in and therefore want to discuss or describe. Taking the example of appendix B the universe of discourse of the Registration Authority consists of all cars, manufacturers, car models, garages, persons, etc., that have existed, exist, or ever might exist and in which the Registration Authority will be interested.

All propositions that may hold in any one or more entity worlds that together constitute the universe of discourse, form the universe of possible proposition. However, for an information system designer not all of those propositions are of prime interest. What he is looking for in the first place are those propositions that hold for all possible entity worlds:

NECESSARY PROPOSITION

A proposition asserted to hold for all entity worlds and therefore must be part of all possible proposition worlds.

Since necessary propositions are states of affairs that necessarily hold for the involved entities in all entity worlds, they often have a more "general" character. These necessary propositions form an abstraction of all entity worlds, generalizing what they have in common.

Necessary propositions define which entities may occur in any entity world - possible entities, and in relevant cases, which entities must occur in each entity world - necessary entities.

The classifications, rules, laws, etc., of the universe of discourse, which are mentioned in chapter 1, constitute the necessary propositions. The section B.2 of appendix B describes informally the necessary propositions of our example universe of discourse.

Some necessary propositions that hold for each and every registered car in all entity worlds containing registered cars in our example universe of discourse are:

"A car is of a particular model."

"Each car has a registration number given by the Registration Authority at the time the car is registered."

However, we do not wish to limit the necessary propositions to only general states of affairs. States of affairs involving one or a few particular entities can necessarily hold for all entity worlds. E.g.:

"Only 5 manufacturers can have permission to operate in the same period of time."

"Fuel consumption is between 4 and 25 litres per 100 kilometres."

As already stated, necessary propositions tend to have a more general character, that is, they hold for collections of similar entities - classes of entities:

CLASS (of entities)

All possible entities in the universe of discourse for which a given proposition holds.

Each class of entities is determined exactly by its possible members. Clearly any particular entity may be a member of many classes, so that classes in general are not disjoint.

The proposition that determines the class might be a state of affairs of arbitrary complexity. E.g.:

- The class of Car Manufacturers consists of all possible entities that produce a car.
- The class of Car Owners consists of all possible entities that either belong to the class of Car Manufacturers, Garages, or Persons, and that own a car.

Classes themselves are entities, and, as the examples already show, can be

given names.

The general notion expressed in information processing literature as "type" is that of "class" or more precisely "class-membership":

TYPE (of an entity)

The proposition establishing that an entity is a member of a particular class of entities, implying as well that there is such a class of entities.

In other words the sentences

"The entity x is a Car Manufacturer (type)"

and

"The entity x belongs to the Car Manufacturers (class)"

convey exactly the same information.

A type can be referred to by means of a type-name. Quite often a singular form of a name (noun) in such cases is used as type-name, while the plural form is used as class-name.

Whether a type notion will be associated with a particular class of entities is an arbitrary choice of the information system designer, often inspired by what is considered practical or usual in the user's environment of the conceptual schema.

The notion "instance" or "occurrence" is usually associated with the notion of type:

INSTANCE or OCCURRENCE (of an entity-type)

An individual entity, for which a particular type proposition holds, that is, which belongs to a particular class of entities.

In designing information systems the notions of class and type are used in particular to establish collections of necessary propositions: With a specific class or type, a collection of relevant necessary propositions may be identified, that hold for all possible entities which are members of that specific class. E.g. in the example of the Registration Authority the following necessary propositions hold for all possible entities that are cars:

- a car is produced by a car manufacturer
 - a car has a serial number
 - a car is of a car model
 - a car is given a registration number by the Registration Authority
- etc.

The propositions that determine such classes or types belong themselves, of course, to the necessary propositions in these cases.

The formal description of the necessary propositions is called the conceptual schema:

CONCEPTUAL SCHEMA

A consistent collection of sentences expressing the necessary propositions that hold for a universe of discourse.

It follows from the above that all possible entity worlds constituting the relevant universe of discourse share a conceptual schema. This conceptual schema in fact establishes the universe of discourse as it informs us what exactly the collection of all possible entities may be.

What propositions are necessary propositions, and therefore what the boundaries of the conceptual schema will be, is arbitrary, and depends on how detailed the information system designer wishes to be. Moreover, this may change over time requiring additional changes to an already formulated conceptual schema.

Propositions may hold in a specific entity world in addition to the necessary ones formulated in the conceptual schema. The description of those additional propositions constitutes an information base:

INFORMATION BASE

A collection of sentences, consistent with each other and with the conceptual schema, expressing the propositions other than the necessary propositions that hold for a specific entity world.

Note, however, that the collection of sentences constituting the one conceptual schema and a specific information base together describe all propositions considered relevant for a specific entity world and therefore describe a specific proposition world for that entity world. These propositions are conceived to hold for the entity world, the latter being perceived as "reality". For that reason, this collection of sentences constituting the conceptual schema and information base must necessarily be consistent, if it purports to be a truthful description of those propositions.

Actually it is the information base together with the conceptual schema that in essence establishes a particular entity world. In other words the entity world consists exactly of those concrete or abstract objects - entities - that are referred to by the terms in the sentences contained in the information base and conceptual schema together.

Note, that it may very well be possible to describe one universe of discourse or one particular entity world in more than one conceptual schema and information base. We assume, however, that usually only one conceptual schema and one information base will be part of one information system at a time.

Often, but not necessarily always, an information base is meant to inform us about the entities that occur in the instant or period of time, usually referred to as "now". A "current" state of an information base - an actual information base - however, may refer to a "past" or "future" entity world:

ACTUAL INFORMATION BASE

That information base which exists in a specified instant or a period of time, usually referred to as "now", and which expresses the additional propositions other than the necessary ones, that hold for an entity world.

ACTUAL ENTITY WORLD

A collection of entities of interest that is described in an actual information base and its conceptual schema.

The entity world described in sections B.2 and B.3 of appendix B could be considered as the actual entity world of interest to the Registration Authority, covering at least the period of time from 1975 until today.

2.2. BASIC CONCEPTS AND DEFINITIONS FOR ACTIONS ON THE CONCEPTUAL SCHEMA AND INFORMATION BASE.

The information base and the conceptual schema will change with time in order to reflect changes in the selected portion of a world constituting the universe of discourse, since only sentences asserted to be true of the universe of discourse should be in the information base or conceptual schema.

Changes in the selected portion of a world are for example:

- * Entities appearing or disappearing in the selected portion,
- * An entity changing its state of affairs or associations with other entities,
- * The classification of entities or some rules or constraints about entities changing,
- * The scope of interest changing, so that the selected portion itself expands or shrinks.

All such changes may require changes to both the information base and the conceptual schema. Although the first two kinds of changes might limit resulting changes to the information base only, the latter two kinds mentioned will certainly cause changes to the conceptual schema as well.

The basic notion of information manipulation in an information base or conceptual schema is an elementary action. Three kinds of elementary actions are defined: insertion, deletion, and retrieval.

INSERTION

The addition of a sentence to the information base or conceptual schema. Other sentences, not deducible before insertion may become deducible and therefore become a deducible part of the information base or conceptual schema.

Note, that a deducible sentence will not automatically be actually inserted.

A typical example may be the insertion of the sentence:

"On 29 January 1975, the car GMF117 is distributed to Smith's garage".

From this it might be assumed deducible that from 29 January 1975 onward:

"Smith's garage owns car GMF117."

The next elementary action is defined as:

DELETION

The removal of a previously inserted sentence from the information base or conceptual schema. Any deducible sentence, which cannot be deduced without the deleted sentence, will no longer be deducible and therefore no longer be a deducible part of the information base or conceptual schema.

Note, that a deducible sentence may have been actually inserted. In that case, the deletion of another sentence, on which the deducibility of the sentence essentially depends, will not automatically delete that actually inserted sentence. The deletion of that other sentence only makes impossible the deduction of the actually inserted sentence.

If we consider the example of an insertion mentioned above, and assume that the deducible sentence

"Smith's garage owns car GMF117"

has been actually inserted, then the deletion of the sentence

"On 29 January 1975 car GMF117 is distributed to Smith's garage"

will not result in the deletion of the sentence stating who owns the car, but we can no longer deduce that sentence.

The last elementary action is defined as:

RETRIEVAL

To make known a sentence which has been inserted in the information base or conceptual schema, or is deducible from other sentences in the information base or conceptual schema.

Note, that the retrieval of a deducible sentence from the information base or conceptual schema is possible only if the information system knows how to deduce this sentence from other available or deducible sentences in the conceptual schema and information base.

Combinations of elementary actions intended to achieve a specific result may be allowed. Such combinations are defined as:

ACTION

One or more elementary actions that, as a unit, change a collection of sentences into another collection of sentences in the information base or conceptual schema and/or make known a collection of sentences present in the information base or conceptual schema.

A typical example of an action is the replacement of a particular sentence by another one, i.e. a deletion followed by an insertion. Due to the many cases where this particular class of actions is applicable, it seems to be convenient to formulate this special kind of action as:

MODIFICATION

The replacement of a sentence in the information base or conceptual schema by another one, thereby possibly changing the collection of sentences which are deducible.

To control actions and rule out impermissible ones, it will be necessary to impose rules or constraints on actions. Therefore the following definition is added to define actions that are considered to be atomic "execution units":

PERMISSIBLE ACTION

An action, conforming to specified rules or constraints, which

- changes a presumably consistent collection of sentences in the information base or conceptual schema into a consistent collection of sentences

and/or

- makes known a consistent collection of sentences present in the information base or conceptual schema.

We should point out here that only the final collection of sentences as a result of a permissible action need be a consistent collection of sentences. Intermediate collections, if recognizable, need not be consistent.

Note, that certain permissible actions may change a presumably consistent, but actually not "truthful" collection of sentences into a consistent and truthful one. Such permissible actions will be needed to correct corrupted information bases or conceptual schemata, whatever the reason of the corruption may be. These specific permissible actions may be allowed to ignore certain rules about permissible or required sequences of state of the involved collections of sentences. For example, if it is erroneously stated that a person is married then changing this information to the statement that that person still is single may involve such a special permissible action (cf. the examples in section 2.7).

An elementary action is caused by an elementary command to the information system:

ELEMENTARY COMMAND

The order or trigger for an elementary action to take place.

Both an action and a permissible action are caused by a command to the information system:

COMMAND

The order or trigger for an action or permissible action to take place.

However, as response to the command, the action may be refused if the permissibility would be violated.

Commands and actions must be described in a suitable language:

COMMAND STATEMENT

A linguistic object expressing a command or elementary command.

In other words, presenting a command-statement to the information system constitutes the command.

It will also be necessary to have facilities designed to express the combination of elementary actions and the identification of them as one unit defining a single action or permissible action:

ACTION DESCRIPTION

A linguistic object describing an action or permissible action.

The possible syntactic and semantic complexity of command statements and action descriptions (e.g. structure of description and expressive power in terms of what commands and actions are describable) depends on the language chosen.

2.3. THE BEHAVIOUR OF AN INFORMATION PROCESSOR.

As already mentioned in chapter 1, the interaction between environment and information system takes place by means of messages:

MESSAGE

A collection of one or more sentences and/or command statements to be used as an information exchange between the environment and the information system.

Messages are dealt with by the information processor part of the information system:

INFORMATION PROCESSOR

The mechanism that in response to a command executes an action on the conceptual schema and information base.

The information processor recognizes whether or not messages received from the environment belong to a given language. Messages which do not belong to this language are discarded as irrelevant. Valid messages may express a change in the universe of discourse, or require to make known one or more sentences present in the conceptual schema, or in the information base, or deducible from sentences present in them.

A message expressing a change in the universe of discourse must contain or be accompanied by a command statement identifying the action description - or the action description itself - for a permissible action to effect the appropriate change in the conceptual schema and information base. The information processor interprets the command and changes the information base or conceptual schema according to behaviour rules or constraints. These rules not only determine whether the resulting collection of sentences in the conceptual schema and information base will be consistent, but also decide whether the conceptual schema and information base may be changed at all depending on what is already present in the conceptual schema and information base.

The information processor will refuse the command if the change cannot be effected according to the behaviour rules or constraints. As a result of a refusal, the information base and the conceptual schema will be as if the command had never been issued (i.e. completely unchanged).

If a message expresses a command to make known a collection of sentences present in the conceptual schema and information base or deducible from them, the information processor interprets the command. It issues a message reporting the appropriate collection of sentences according to behaviour rules or constraints which specify when and which collection of sentences present in the conceptual schema and information base shall be reported as a result of the command. The rules include the inference rules in case deducible sentences are involved.

The information processor will refuse the command if any behaviour rules or constraints would be violated in reporting the required sentences.

When it issues a command, the environment needs to know if the command is honoured or refused. Therefore the information processor must issue the environment a message to that effect.

In other words, the result of a permissible action is a function of "controlling" sentences stating rules and constraints, the sentence(s) to be changed or made known, and the incoming message including the command statement and additional sentences if relevant.

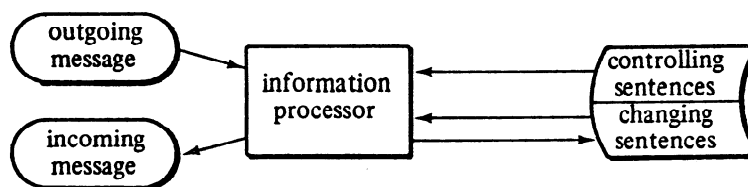


Figure 2.1. The information processor in action.

Usually the controlling sentences are largely in the conceptual schema although sentences in the information base as well can have a controlling role. The sentences to be changed or made known, are in normal application cases, all in the information base. However, the involved sentences are in the conceptual schema in cases where the conceptual schema itself is subject to change or reporting. Note, that in this latter case some of the controlling sentences may be found in the information base also.

A sentence expressing a proposition as such has a truth value. The following principle is axiomatic:

The truth value of a sentence is considered true if it is explicitly stated true by a message accepted from the environment according to the rules known to the information processor. It is also considered true if it is deducible from such explicitly stated sentences in accordance with the inference rules known to the information processor.

The truth value of any sentence, whose denial can be deduced similarly from explicitly stated sentences, is considered to be false.

The truth value of all other sentences is considered to be undecidable.

The task of the information processor is twofold. Sometimes the emphasis lies on deciding whether changing a collection of sentences is permissible or not; at other times the deduction of "new" sentences or the generation of outgoing messages seems to be more important. Therefore the rules or constraints controlling the first role sometimes are called permissive rules or constraints; the rules prescribing the sequence of actions, to be performed including the generation of deducible sentences and messages, are referred to by the term prescriptive rules or constraints.

2.4. INSERTING A CONCEPTUAL SCHEMA - THE MINIMAL CONCEPTUAL SCHEMA.

Initially, the only rules known to the information processor are those built into the information processor itself. This basic set of rules provides the information processor with an interpreter mechanism that enables the information processor to at least recognize and interpret a minimal language. This language allows the environment:

- * to extend the language to include constructs appropriate for the description of the universe of discourse,
- * to specify commands,
- * to specify authorizations,
- * to specify new behaviour and inference (deduction) rules relevant to the conceptual schema and information base,
- * to introduce action descriptions for permissible actions.

Further, the information processor will be equipped with a set of algorithms allowing it to derive new sentences from the already available ones.

These built-in behaviour rules constitute the minimal conceptual schema of the information system.

Given this built-in minimal conceptual schema and an initially empty information base it is possible for the environment to use this minimal language to build up the required conceptual schema as well as the information base in a systematic fashion.

There must be a fundamental rule concerning the proposed insertion of a new rule or constraint in order to decide what to do in the event that this insertion would make certain collections of sentences, already present in the conceptual schema and information base, no longer consistent. Such a fundamental rule can reject the existing sentences or the proposed new rule. It must not accept both.

At the beginning any sentence accepted by the initial built-in behaviour rules probably becomes an additional rule or constraint by being included in the conceptual schema or possibly in the information base. As sentences are added, the number of rules or constraints specified in the conceptual schema and information base increases and so further constrain and control the permissible actions to the conceptual schema and information base.

2.5. BEHAVIOUR RULES FOR THE ENVIRONMENT.

The behaviour rules mentioned so far establish the behaviour of the information processor and what is permissible in the information base or conceptual schema. A distinction has been made between permissive rules and prescriptive rules.

Information systems may issue messages to the environment which are intended to cause change in the environment. The behaviour rules within such information systems must be extended accordingly. This situation, however, does not imply that the information system controls the environment. Firstly these messages are generated according to rules or constraints established by the environment. Secondly, the information system cannot force the environment to obey rules expressed by such messages.

The additional rules consist not only of permissive and prescriptive rules for the information processor, but also of permissive and prescriptive rules for the environment. The permissive rules for the environment establish the criteria needed for the information processor to test the actual information, so that the information system can generate warning messages. The prescriptive rules for the environment form the base for the information processor to generate appropriate requests to the environment.

These latter behaviour rules for the environment will never dictate what is permissible in the information base or conceptual schema including consistency rules for collections of sentences. As far as the information system is concerned, they formulate only what is desirable. As such they form a separate class of rules or constraints in the information system. Note, that a undesirable collection of sentences will nevertheless be consistent.

For example, if a behaviour rule of the universe of discourse prescribes that only black Ford cars should be produced, then the reporting of a red Ford car being produced must not be rejected by the information processor; instead a warning message should be issued to the appropriate authorities in the environment.

Summarizing we may say:

- What is considered to be impossible in the universe of discourse or environment, establishes what is not permissible in the information system including whether a collection of sentences is consistent (behaviour of the information system).
- What is considered to be not permissible in the universe of discourse or environment, is undesirable for the information system, although it is perfectly permissible (behaviour of the environment).

2.6. STATIC AND DYNAMIC RULES AND CONSTRAINTS.

Static aspects of a system are those which apply to each of its individual states. Static rules or constraints establish dependencies between parts of the system at any one instant of time. Dynamic aspects are those which govern the evolution of the system. To discuss dynamic aspects of a system means to dis-

cuss its laws of change. Dynamic rules establish dependencies between parts of the system through several instants of time.

The system we are considering here is the information system together with its environment and the universe of discourse. Rules and constraints deal with dependencies within and between all of these. In this generality, they touch on some subtle problems, in particular when dynamic aspects are involved. For a thorough treatment they require more elaboration of concepts. Therefore, we will here outline a few aspects of dynamic rules in the general sense, and then restrict ourselves to static and dynamic rules in a more special sense.

The causal structure of interdependencies is an important aspect of dynamics in the general sense. A typical example of such dependency concerns co-ordination of permissible actions: There may be rules requiring certain conditions to hold before specific actions can take place (see section 2.8). Another typical issue is to describe what messages cause the information processor to perform which changes to the information base or conceptual schema and to return which answers, say, results of retrievals or reporting on actions (cf. sections 2.2 and 2.3).

A third important example is the subject of authorization. In general, there must be authorization rules that control whether a user is entitled at all to give a command for a particular permissible action changing or retrieving a particular collection of sentences, or is entitled to receive a particular message from the information system. This implies that identification of the source and destination of messages and commands are involved in the enforcement of authorization rules. The subject is not yet discussed further in this Report.

Although a comprehensive information system design must be aware of all above mentioned relevant aspects, the focal points of interest traditionally are the rules and constraints for the sentences administered by the information processor. Therefore, and because of the complexity of the general treatment, we shall consider in this section the rules or constraints in the special sense of permissibility of information base states - that is, collections of sentences - and sequences of information base states.

Static rules and constraints under this restricted view are then concerned with the consistency and permissibility of collections of sentences. The effect of a static rule may be locally restricted to single sentences or it may globally involve several sentences within the same collection of sentences in one state of the information base. An example of a locally effective rule is the requirement that the serial number of a car must be, say, a natural number less than 10 000 000. A rule that requires the serial numbers of cars to be unique has a very broad global effect. To be enforced or checked, it needs a complete survey of all instances of serial numbers of cars registered in the information base. Static rules may also be of very different complexity. Examples of more complex conditions are functional dependency or set inclusion. Detailed examples of static rules may be found in the appendices D, E, and F of this Report.

Dynamic rules under the restricted view in this section are concerned with the permissible transitions from one collection of sentences to a next one and thus specify the possible sequences of information base states. Therefore they will be called transition rules to distinguish them from the dynamic rules in the general sense. Transition rules abstract from causes for changes as well as from effects the changes may have on the environment (e.g. reactions to retrieval results or to triggered messages from the information processor to the en-

vironment). They simply describe which information base states may occur subsequent to other given information base states.

Satisfaction or violation of transition rules can sometimes be checked by inspection of states. In fact, static rules can be regarded as special cases of transition rules. Although static rules define which states are permitted, a static rule can be re-interpreted as stating that certain states are permitted or forbidden, no matter what the previous state was. This can, however, also be expressed as a transition rule. The "no matter what the previous state was" can be taken care of by admitting all information base states as possible most recent information base states.

It may be argued that permissibility of an information base state may depend on any of the former states, rather than just on the most recent one. However, as an additional postulate, it is assumed that the history of states can affect permissibility only in as much as the history is reflected in the most recent state. It is therefore sufficient to take into account the most recent state only (cf. "actual information base" in section 2.1).

Some special attention must be given to the kinds of rules that are involved in derivations of deducible sentences. For example, the fuel consumption rate of a car may be given as the fuel consumption per 100 kilometres. That is, given a sentence stating the kilometrage of a particular car and a sentence stating the amount of fuel consumed, the fuel consumption rate can be derived, provided a sentence stating the definition - rule - for the fuel consumption rate is also available.

Actually, such rules are static rules as they deal with sentences in one information base state. Some people, however, may consider them dynamic rules, as they "control" possible derivation processes. Several authors use the term "derivation rule" for such rules.

Note, that deducible sentences need not only be derived by derivation processes within the information system. It is quite possible that such sentences are explicitly inserted. In such cases these "derivation rules" control the consistency of the resulting collection of sentences including the inserted "derivable" sentence.

The above mentioned rules not only include those needed for what is commonly considered as derivable information. The decision rules needed in automatic process control systems and decision support systems are also in this category. Therefore, they belong to the necessary propositions in most cases and are thus an essential part of the rules described in a conceptual schema.

2.7. EXPRESSING RULES AND CONSTRAINTS.

We will consider two different ways of specifying transition rules, the state-oriented and the action-oriented descriptions.

With the state-oriented descriptions, the rules and constraints are given as requirements on subsequent information base states. A rule or constraint is then basically a description of a set of pairs of information base states <OLD, NEW>. A change from a state OLD to a state NEW is permissible - regardless of how it is effected - if and only if the pair <OLD, NEW> is in the set.

Any transition rule distinguishes permitted pairs of information base states from forbidden ones. Thus, it can be viewed as a binary-valued function which tags each pair of information base states with either "+" (permitted) or "-" (forbidden). This defines a dichotomy on the Cartesian product $S \times S$ of the set S of all information base states with itself. Such a dichotomy is a decomposition of $S \times S$ into two sets, $T+$ of permitted and $T-$ of forbidden changes. $T+$ could be called the positive, and $T-$ the negative extension of the rule. Either of the two sets can be used to describe the dichotomy. The effect of a transition rule is therefore completely captured by either its positive or its negative extension.

With the action-oriented descriptions, the permissible changes are given by admissible action sequences. The permissibility of an action or action sequence may depend on the present state. A rule or constraint is therefore basically a set of pairs which each consist of an information base state component and an action sequence component.

With the action-oriented descriptions, a rule specifies that a transition is permissible if, starting from a permissible state OLD, the transition is effected by an action sequence Q such that the pair $\langle \text{OLD}, Q \rangle$ is in the set described by the rule. Ultimately, permissibility may be traced back to an initial state and all actions performed on it until the present.

To be able to make finite descriptions of virtually infinite sets of action sequences - there is no restriction on the length of action sequences - it is necessary to define classes of actions. An action-oriented rule then refers to classes of action sequences. Conceivably, an action sequence may consist of only one elementary action (for "elementary action" and "action" see section 2.2). Complex rules are formed by composing actions to constitute a permissible action (for given departure states).

A permissible action succeeds or fails as a whole. The actions of which a permissible action is composed might not be permissible individually. Thus, given the rule that an employee must have a salary and must work for a department,

(INSERT "John works for Department Sales",
INSERT "John earns a salary of 20000")

may be permissible, while each individual INSERT action alone would violate the above mentioned rule. Permissibility in this case is dependent on both elementary actions happening together.

To demonstrate examples of static and dynamic rules let us assume that the marital state of persons might be defined as one of the following states: "single", "married", "widowed", or "divorced".

A static rule applicable on sentences within one collection of sentences may be:

If a person is married to another person, then both persons must have a marital state of married.

The following table expresses an example set of state-oriented rules for permissible changes of marital state, defining a permissible sentence in the resulting collection of sentences as a function of a sentence in the initial collection of sentences:

from/to	SINGLE	MARRIED	WIDOWED	DIVORCED	UNKNOWN
SINGLE	-	yes	no	no	yes
MARRIED	no	-	yes	yes	yes
WIDOWED	no	yes	-	no	yes
DIVORCED	no	yes	no	-	yes
UNKNOWN	yes	yes	yes	yes	-

Figure 2.2. Transition matrix for marital state.

An example for an action-oriented rule is the following:

INSERT "x is married to y"
only if both x and y are not married.

These basic description possibilities are discussed in more detail, with some examples, in appendix G; appendices E and F also give some examples.

A suitable language is required for the specification of rules and constraints. The totality of established rules and constraints can be regarded as one comprehensive rule that controls the entire information base. Of course, in any reasonable language such an overall rule will not be given in one piece. Therefore, it is a requirement for the language to allow for composing complex rules and constraints from simpler ones. The decomposition into simpler rules must end with predefined primitive rules. A primitive rule would have to be a simply structured, easily surveyable set of pairs of information base states or of an information base state and an action sequence.

At present, no specific proposal is made for composition of rules and constraints, but a general aspect is considered. The rules may be expressed in a permissive or restrictive style. Composition may be additive or subtractive - that is, one rule may work in the same sense as, or counter to, another. The result may be a permissive or a restrictive rule. This gives a number of composition options, not all of which may be desirable for a practical specification language.

2.8. CO-ORDINATION OF PERMISSIBLE ACTIONS.

It should be noted, that the subject of this section in particular has only rather recently become a topic of research and discussion (e.g. [2, 3, 4, 5]). Therefore this section suggests more the directions in which development might go, rather than demonstrating and summarizing results.

All actions taken by the information processor on the information base and conceptual schema should occur in terms of permissible actions as defined in section 2.2. Because of this, any process performed by the information processor will consist of one or more permissible actions.

A permissible action is considered to be atomic and therefore uninterruptable. It is triggered by an appropriate command. The effect of a command might be a "chain" of permissible actions. That is, a permissible action may issue a command for other permissible actions.

The conceptual framework for co-ordinating permissible actions is based on the following concepts:

EVENT

The fact that something has happened in either the universe of discourse, or the environment, or in the information system;

COMMAND (as in section 2.2)

The order or trigger for an action or permissible action to take place;

PERMISSIBLE ACTION (as in section 2.2)

An action, conforming to specified rules or constraints, which

- changes a presumably consistent collection of sentences in the information base or conceptual schema into a consistent collection of sentences

and/or

- makes known a consistent collection of sentences present in the information base or conceptual schema;

COMMAND CONDITION

The precondition, including synchronization aspects, that must be met before a permissible action may take place.

The information system only reacts because of an event. The dependency between the event and the reaction could be perceived as in figure 2.3.

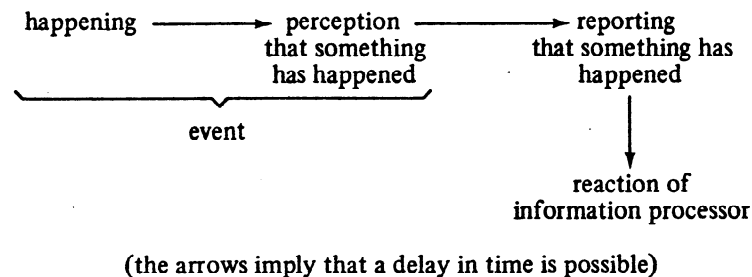


Figure 2.3. Dependency between event and reaction.

As far as the information system is concerned it is not relevant whether the event is the happening or the perception of the happening. However, the event (the stimulus) must cause a report of the event to the information processor, upon which the information processor has to react.

We define two kinds of events: external events and internal events.

EXTERNAL EVENT

An event that occurs in the environment or universe of discourse.

The reporting of such an event is a message to the information system expressing at least a command.

INTERNAL EVENT

An event that occurs because of the termination of some permissible action in the information system.

Depending on the result of such a permissible action, internal events are relevant only if the reporting of the termination must be followed by a succeeding reaction of the information system, or by a message sent to the environment.

Events have certain characteristics, which are important for the information system. In the first place, events may belong to certain types (classes) of event. Closely associated with the notion of event type is the notion of event occurrence.

An important characteristic of the event type, or rather its reporting, is the type(s) of command, associated with the event type, which determine the permissible action or permissible actions that are expected as a reaction of the information system.

Other important characteristics of the event types are:

- the number of event occurrences to be expected;
- the frequency of event occurrences.

Most events also need an associated reporting of certain "parameters" in order to direct the permissible action as to what precisely must be done. In those cases the message reporting the event also contains, or must be accompanied by, one or more sentences, called input sentences for the permissible action.

The permissible actions accomplished by the information processor may insert, retrieve, delete, or modify sentences in the conceptual schema and information base. It is not necessary, however, that input sentences for the permissible action be inserted themselves in the conceptual schema or information base, if the input sentence only provides parameters to the permissible action. Sometimes input sentences will be inserted, at other times they cause the insertion of other derived sentences. An input sentence which is not actually inserted, will generally not be reproducible after the termination of the permissible action.

A permissible action is considered as a "black box", that is, we are interested only in what the permissible action does - what its result is. This, however, may be specified in terms of (elementary) commands for (elementary) actions. Such specifications concentrate on the type of permissible action: The type of permissible action determines what the permissible action will do. This is described by the action description. The sentences involved in the permissible action establish the actual result. Together they establish the actual permissible action, that is, an instance of the permissible action type.

A permissible action is triggered by an event, or more precisely by the command expressed in the message reporting the event. The type of event determines what type of permissible action will be triggered. At that moment the permissible action becomes active. An active permissible action will be uninterruptable

until it is finished. This termination may raise an (internal) event, depending on the result of the permissible action.

At any time several permissible actions can be active in the information system. These permissible actions may belong to the same or different types of permissible actions. Two permissible actions which are active at the same time are considered to operate completely independently of each other.

As already indicated a certain delay of time may occur between the event itself and its reporting to the information processor, or between its reporting and the information processor's reaction. Therefore, it is possible that the sequence in which the events are reported to the information system may differ from the sequence in time of the events themselves occurring in the universe of discourse. If this sequence in time of events is relevant, for instance, because of necessarily preceding states of affairs that must have been recorded first in the conceptual schema and information base, the resulting permissible actions must be co-ordinated, including, if necessary, synchronization of the permissible actions.

The command condition of a certain permissible action consists of the (external and internal) events which must have occurred before triggering the permissible action, and a rule or set of rules which establish in what manner the events determine the necessary condition for the permissible action to be triggered. A command condition can be extended and made more precise by rules about sentences reporting the events or already present in the conceptual schema and information base.

If more than one event is involved in the command condition of a permissible action, then the last event occurring in time, independent of what event this may be, fulfils the command condition and the permissible action is triggered. The "arrival" of the other events except the last one brings the command condition to a "wait-state".

A command condition may have a limited wait-state, that is, the wait-state is only allowed to have a limited duration in time. If within this period the last event needed does not occur, then the permissible action will no longer be triggered because of the events that have already occurred. A new series of events, however, may prepare a new wait-state for a possible permissible action. When no limited wait-state has been defined, the wait-state may last eternally.

One event occurrence can only once take part in the triggering of one permissible action. When a command condition is met and the permissible action is triggered, then the "arrival" of new events can only prepare the command condition for a new triggering of the permissible action.

A special note within the context of command conditions is needed about the events. An event (occurrence) itself is information bearing, as it establishes that something has happened. This is not established in general terms, but in a most specific way. Therefore, an event does not establish that a happening of a certain type has occurred, but that that particular happening (occurrence) with that particular result took place. For example, a certain car PCXX999 has been produced by FORD Motor Company.

As a consequence, internal events are controlled by prescriptive rules for the event. A type of permissible action may possibly terminate in a number of dif-

ferent internal events depending on the result of the permissible action. Such various events may be mutually exclusive, but this is not necessary.

During the permissible action (and actually before the execution of the permissible action itself) the information processor deals with a number of rules and constraints:

1. It recognizes the reporting of the event and the reception of the input sentence according to recognition rules;
2. It evaluates the command condition and triggers the actual permissible action or keeps the command condition in a wait-state;
3. After triggering the permissible action it performs the permissible action itself according to prescriptive rules expressed in the action description of the permissible action manipulating the conceptual schema and information base;
4. It checks the manipulation of the conceptual schema and information base according to the appropriate static and dynamic rules and constraints for the conceptual schema and information base, and the appropriate authorization rules;
5. It possibly reports one or more appropriate internal events according to the prescriptive rules for the internal events;
6. It possibly generates an outgoing message according to prescriptive rules for the outgoing message.

In the following a graphic formalism will be used to discuss some possibilities of co-ordinating permissible actions. This formalism is derived from those used in Petri Network approaches [6]. In the diagrams the command condition mechanism (C) is depicted by a circle. The boxes depict permissible actions (PA). The incoming arrows show the sources of the events involved in the fulfillment of the command condition.

Co-ordination of permissible actions can involve all kinds of "chains" of permissible actions. Several possibilities exist, e.g.:

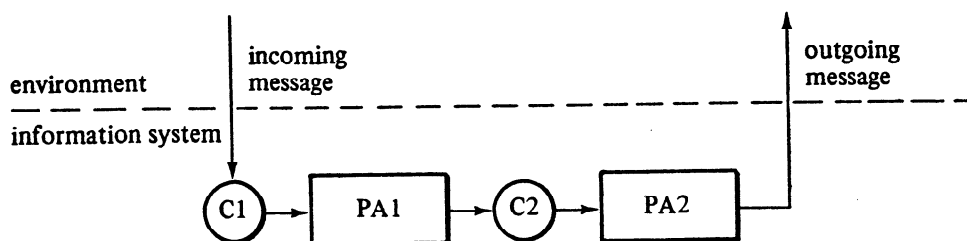


Figure 2.4. Simple sequence of permissible actions.

1. The result of an incoming message is an ordinary sequence of permissible actions. That is, a first permissible action takes place because

of the message from the environment. The arrival (external event) fulfils the command condition for this permissible action. The result of this permissible action (internal event) is the sufficient and only command condition to be fulfilled for a second permissible action. This is shown in figure 2.4.

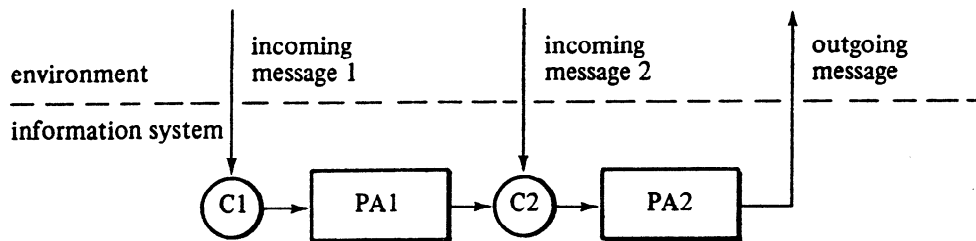


Figure 2.5. Conditional sequence of permissible actions.

2. The ultimate result of an incoming message is a sequence of permissible actions, provided additional incoming messages are received. In figure 2.5 the second permissible action will take place if both the first permissible action has taken place (internal event) and a second incoming message has arrived (external event). The one that occurs latest in time completes the command condition and therefore triggers the second permissible action.

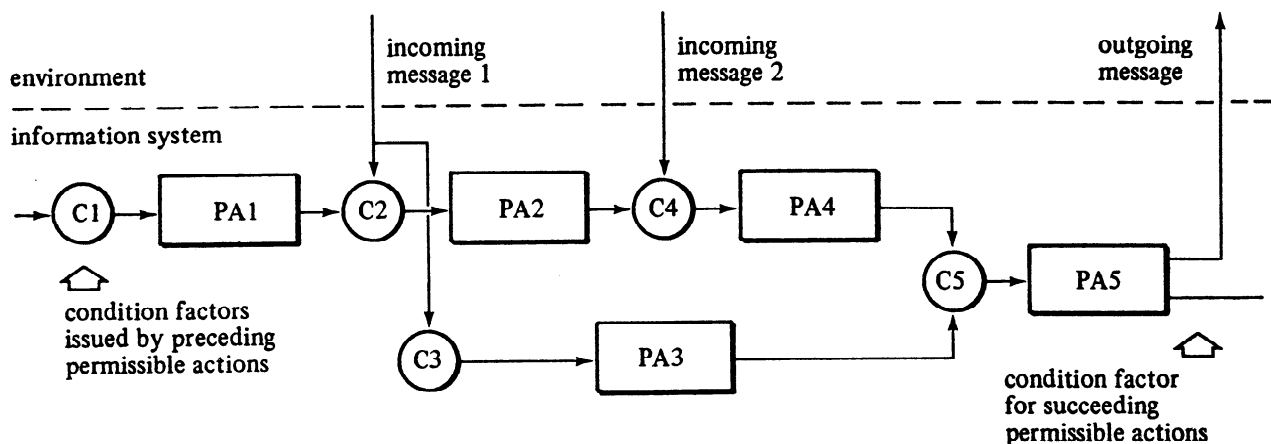


Figure 2.6. Co-ordinated sequences of permissible actions.

3. More complex situations are possible as for instance is depicted in figure 2.6. Note, that in this example permissible action 3 may be concurrent with the sequence of permissible action 2 and permissible action 4. Permissible action 3 and permissible action 4 both are predecessors to permissible action 5. Also note, that the arrival of incoming message 1 is part of the command conditions of both permissible actions 2 and 3.

Appendix H discusses some aspects of co-ordinating permissible actions in more detail and gives some demonstration examples.

2.9. REFERENCES.

- [1] MURRAY, J.A.H. et al.
 `The Oxford English Dictionary` with supplements,
 Clarendon Press, 1933 - 1977.
- [2] HECKENROTH, H., TARDIEU, H., ESPINASSE, B.
 `Modèles et outils pour la conception de la
 cinématique d'un système d'information`,
 In: Rapport de recherche IRIA 310, CETE d'Aix,
 Grasce, Université d'Aix - Marseille, 1980.
- [3] VALETTE, R., COURVOISIER, M.
 `Recherche d'un modèle adapté au système de
 commande de processus à évolution parallèle`,
 In: RAIRO, Vol 11, No 1, 1977.
- [4] BODART, F., PIGNEUR, Y.
 `A model and a language for functional specifi-
 cations and evolution of information system
 dynamics`,
 In: Formal Models and Practical Tools for
 Information System Design, H.S.Schneider (ed),
 North Holland Publishing Company, 1979.
- [5] BODART, F.
 `Logical specifications for system dynamics. An ex-
 tension to PSL`,
 ISDOS working paper No 183, April 1979.
- [6] PETERSON, J.L.
 `PETRI Nets`,
 In: Computing Survey, Vol 9, No 3, 1977.

CHAPTER 3.

SOME CONCEPTS AND PRINCIPLES FOR IMPLEMENTATION.
=====3.1. PRINCIPLES FOR THE CONTENTS AND SCOPE OF A CONCEPTUAL SCHEMA.

To design a conceptual schema, that is, to establish and describe the necessary propositions of the universe of discourse, the information system designer starts by observing the selected portion of the world constituting the universe of discourse and constructing in his mind an abstraction of it, much in the same way as a scientist, by experiment and analysis, constructs a theory of the observed phenomena.

As in science, we would normally expect that such an abstraction would be general enough to encompass many different (preferably all) occurrences of the same kind of phenomena. This abstraction constitutes propositions that necessarily hold in all possible entity worlds, just as the theory must hold for the observed or expected phenomena.

The process of observation, abstraction, and conceptual schema formulation is usually iterative. This is similar to the scientist's attitude during the formulation of a theory: further experimentation - that is, observation of the universe of discourse - is generally required to clarify some aspects of the theory.

Although selection of what is considered to be necessary propositions about the universe of discourse to be described in the conceptual schema is to a certain extent arbitrary, as already is mentioned in chapter 1, the systems designer might consider various factors in deciding the boundary of the the conceptual schema.

An example of a widely accepted and also basic distinction has already been mentioned in chapter 1. It is the distinction between sentences which represent general laws and rules to which possible entities in the universe of discourse have to adhere, and sentences which describe facts about particular entities in the universe of discourse, following the laws and rules described in the conceptual schema. The main reasons for this distinction are practical arguments with respect to the design, implementation and maintenance of the information system, because some sentences are generalizations (conceptual schema) and others are specifics (information base). These generalizations have already been listed in chapter 1, section 1.8, as:

- describing classes (types, variables) in the universe of discourse rather than individuals (instances),
- describing concepts that are less subject to change rather than concepts that are changing more frequently,
- inclusion of rules or constraints having wide influence on

the behaviour of the universe of discourse (and therefore on the behaviour of the conceptual schema and information base) rather than narrow influence.

The guiding principle in selecting the necessary propositions is that they, although stating something about the universe of discourse, are convenient in controlling the consistency of the collection of sentences in the information base and the permissibility of their manipulation. As already stated in section 1.8 of chapter 1, the following general principles for the conceptual schema should be observed at all times:

* 100 Percent principle:

All relevant general static and dynamic aspects, i.e. all rules, laws, etc., of the universe of discourse should be described in the conceptual schema. The information system cannot be held responsible for not meeting those described elsewhere, including in particular those in application programs.

* Conceptualization principle:

A conceptual schema should only include conceptually relevant aspects, both static and dynamic, of the universe of discourse, thus excluding all aspects of (external or internal) data representation, physical data organization and access as well as all aspects of particular external user representation such as message formats, data structures, etc.

Some justifications and remarks may amplify the two principles:

Most of the general rules of the universe of discourse are currently described in application programs, if they are described at all. Sometimes such rules are referred to by the term "validation rules" in the current EDP jargon. In many situations, more than one update program operates on the same data base. In this case a single rule has to be described in each application program which may affect a concrete physical representation of that part of the information base covered by the rule. This results in a redundant representation of the rule, which is source for inconsistency among the various "copies" of the rule. It is obvious that due to such redundancy it is difficult to control, verify, and maintain a set of interrelated rules. It is even more difficult to detect contradictions in such a set of dispersed rules.

User update and query languages are becoming available which provide the (end) user tools to express his requests directly to the machine as opposed to a professional application programmer who writes the program on behalf of the employing user. One does not need too much imagination to think of situations where a user "has no time" to program the rules in his direct instruction of the machine. This may result in a stream of messages entering the information base that should not enter. The result is information base pollution. If all rules are expressed in the conceptual schema, and therefore are controllable by system functions independent of users or application programs, then there is no basis for such pollution.

If all general rules of the universe of discourse are expressed in the conceptual schema, then it is (much) easier to extend or modify these rules in a controlled way as compared to the situation where the rules are scattered over

several application programs. The need for such extension and modification is a normal thing to expect in practice.

It is easier to understand and to teach information systems design where all rules are in the conceptual schema, because of the absence of the previously mentioned kind of redundancy.

A conclusion of the 100 Percent Principle is that a conceptual schema language must be able to describe any set of general rules, etc. of the universe of discourse in a conceptual schema. In order to achieve this, some of these rules may be described in a procedural way while others may be described in a declarative way. This depends on the present state of the art of formal languages.

The Conceptualization Principle says that the conceptual schema must only and exclusively include conceptually relevant aspects of the universe of discourse. Aspects, constructs, or distinctions, which refer to other components or factors of the information system are not allowed to be part of the conceptual schema. Such aspects are, for example:

- Representation aspects of data in the user views;
- Aspects of machine efficiency and physical data organization;
- Organizational aspects of the information system.

By focussing on the conceptually relevant aspects, the conceptual schema design process is relieved of the burden of computer implementation aspects. The design process is in this way significantly simplified. More precise and accurate conceptual schemata will be the result of the conceptual schema design process. Moreover the design processes of the (external) application programs and the (internal) data storage routines will also be simplified as these can be limited to the external and internal representation aspects respectively.

Furthermore, by avoiding aspects of machine efficiency and physical data organization in the conceptual schema, operations at the user interfaces can be made completely independent of those "internal" aspects. Thus, reprogramming of applications in case of physical data reorganizations can be avoided, due to the "isolating" effect of the conceptual schema.

Finally, the observation of this principle also results in better evolution, simply because there is no burden of conceptually irrelevant aspects which could increase the reprogramming effort in cases of conceptual schema changes.

3.2. PRINCIPLES FOR THE DESCRIPTION OF A UNIVERSE OF DISCOURSE.

The theoretical foundation for describing a universe of discourse is, as already mentioned chapter 1, section 1.7, the use of an interpreted formal system of logic. The establishment of such a formal system of logic to describe the universe of discourse requires:

1. The determination of an alphabet for which it is unambiguously and mechanically recognizable whether a given character is in the alphabet, conditions naturally satisfied by conventional character input to a digital data processing system;

2. The provision of a finite set of effective rules (algorithms) that determine which strings of characters are to be taken as well-formed and, in particular, which well-formed strings are to be taken as sentences;
3. The provision of a finite set of effective rules that determine whether a given sentence is an axiom;
4. The provision of a finite set of effective rules that determine whether a given sentence can be immediately deduced from a given set of sentences;
5. The provision of a finite set of rules of interpretation that assign to each sentence a meaning such that the sentence is interpreted as an assertion unambiguously true or false about the universe of discourse.

In this context an axiom is defined as:

AXIOM

Any closed sentence that is asserted to be considered as such by an authorized source.

The axioms and deduction rules are chosen in such a way as to result in:

- each axiom being interpreted as a true assertion about the universe of discourse,
- each sentence immediately deducible from a set of sentences interpreted as true assertions about the universe of discourse being itself interpreted as a true assertion about the universe of discourse.

There is, of course, nothing unusual in these principles. They are simply a precise explication of the ordinary process of deductive inference couched in terms of a formal system so that they may be applied in the context of digital information systems.

Points 1 - 4 above relate to the syntax or grammar of a conceptual schema and information base. Principles of an abstract syntax are discussed in section 3.3, so that any formal language adhering to this abstract syntax can be used to carry out the purposes described in this chapter. Point 5 above relates to the semantics of the conceptual schema and information base. The principles for the semantics are discussed in section 3.4 with a similar objective.

A suitable formal system of logic to describe a universe of discourse may place minimal demands, in principle, on the part of the conceptual schema built in the information processor [1]. That is, the built-in minimal conceptual schema can be limited to bare essentials. In practice, however, many linguistic constructs, that in theory might be unnecessary, may very well be included in an actual information processor for reasons of efficiency, convenience and costs. Moreover, it is recognized that human perceptions of the universe of discourse, as well as communication of those perceptions to others, may differ considerably from individual to individual.

The information system designer must be free to express the conceptual schema

in terms appropriate to the specific universe of discourse of concern and to the user's perception of it. Specifically, no constraints are to be imposed on the entities assumed to exist in the universe of discourse or on the properties they may be asserted to possess. The information system designer is constrained only by the requirement to adhere to the basic principles of logic built into the information processor and to the fundamental need for maintaining consistency.

Considering also that conceptual schemata must be easy to use and to understand by a variety of users, a mechanism for adding linguistic constructs of arbitrary complexity ("macro constructs") to the formal system is necessary so that users of the system may interact with it at any desired level of aggregation of constructs [2, 3, 4]. In order for this principle to be observed without violating the first principle, the formal system must have within itself a mechanism for incorporating the definitions of new constructs in terms of those already present.

Taking into account the 100 Percent and Conceptualization principles, we can already deduce some general guidelines for a system of linguistic constructs for describing universes of discourse. Even then we have the choice among a variety of linguistic constructs. Each of those, however, has ultimately to be based on the fundamentals of formal logic.

In order to illustrate the choices, we list a few examples:

One choice is, to set up a system of concepts without defining dedicated constructs, i.e. elementary constructs that express the fundamentals of formal logic are applied only for the description of a universe of discourse. The major advantage of such a "fundamentalistic approach" is that one can rely upon the long tradition of formal logic. However, it is questionable whether this choice does not overstrain the capabilities of the average information system designer. To overcome this problem the introduction of suitable macro-constructs may be useful.

For example, a well-known possibility might be to distinguish propositions stating the type or class membership of entities and to design constructs to easily express such propositions. Many persons who are involved in practical conceptual schema design appreciate the merits of having a distinct notion of entity-type.

In many existing systems of constructs for conceptual schemata, there is also the distinction of sentences describing all classes of additional propositions possible for all entity worlds of the universe of discourse, and other sentences describing all relevant restrictions with respect to the allowed populations of the former type of propositions. Again, the reason to make this distinction has to do with practical considerations: In conceptual schema design, the determination of types (classes) usually and naturally comes before the determination of restrictions of their populations.

Furthermore, the required language must reflect the concepts established in section 2.2 of chapter 2: command-statement and action-description for insertion, deletion, retrieval, modification, etc. and suitable constructs for expressing the dynamic rules and command-conditions, as outlined in chapter 2, section 2.7 and 2.8.

3.3. ABSTRACT SYNTAX FOR A CONCEPTUAL SCHEMA AND INFORMATION BASE.

Precise definition of a universe of discourse, or, indeed, any precise dialogue about a universe of discourse, whether to a human or a computer, requires a language. In the present context, since communication with a computerised information system is essential, the language is necessarily formal. For a language to be formal it is not enough that its grammar and meaning be precisely specified. It is also essential that there be a decision procedure (an algorithm) which can, by examination of the individual alphabetic characters in an expression, determine whether the expression is grammatically correct in the language, and furthermore what kind of language constructs are involved in the expression. For example, it must be possible to determine which expressions are names, which are sentences, what sequences establish deducibility of a sentence, etc. In such a case a computer process can parse the language completely.

There are many ways of defining the syntax of a formal language, such as Backus-Naur form, production rules, the method chosen in appendix C, and so on. In order not to prejudge the grammatical form of any conceptual schema language, this Report specifies the syntax by defining the grammatical notions without specifying any rules about how these notions might be expressed in any particular language. These notions are: terms and sentences as defined in chapter 2, section 2.1, and grammatical functions called functors [5]:

FUNCTOR

A linguistic object that refers to a function on other linguistic objects taking as arguments (input) a list of linguistic objects (terms, sentences, functors) and yielding as a value (output) a single, uniquely determined linguistic object (term, sentence, functor).

Terms and sentences may involve variables:

VARIABLE

A term which refers to unspecified, indeterminate entities in the universe of discourse.

Sentences may be either open or closed. For example, the sentence

"x is a manufacturer."

which cannot be said to be true or false unless it is known to what x refers. Such sentences are said to be open sentences.

The sentences

"Ford is a manufacturer."

and

"for all x, if x is a manufacturer, then x is a company."

are, however, either true or false, depending on the universe of discourse. Such sentences are called closed sentences. Only closed sentences are unambiguously true or false and for that reason will be the only kind of sentences appearing in a conceptual schema and information base.