SEMANTIC AND ORGANIZATION CONSIDERATIONS IN DATABASE CONCEPTUAL MODELLING:

THE *SE*MANTIC *C*ONCEPTUAL *O*RGANIZATIONAL *M*ODEL (SECOM)

ABDALAH ALSHAWI

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF ASTON IN BIRMINGHAM

JUNE 1991

THE UNIVERSITY OF ASTON IN BIRMINGHAM

SEMANTIC AND ORGANIZATION CONSIDERATIONS IN DATABASE MODELLING:

THE *SE*MANTIC *C*ONCEPTUAL *ORGANIZATION M*ODEL (SECOM)

ABDALAH ALSHAWI

DOCTOR OF PHILOSOPHY

JUNE 1991

This thesis presents a new approach to designing large organizational databases. The approach emphasizes the need for a holistic approach to the design process. The development of the proposed approach was based on a comprehensive examination of the issues of relevance to the design and utilization of databases. Such issues include conceptual modelling, organization theory, and semantic theory.

The conceptual modelling approach presented in this thesis is developed over three design stages, or model perspectives. In *the semantic perspective*, concept definitions were developed based on established semantic principles. Such definitions rely on meaning - provided by intension and extension - to determine intrinsic conceptual definitions. A tool, called meaning-based classification (MBC), is devised to classify concepts based on meaning. Concept classes are then integrated using concept definitions and a set of semantic relations which rely on concept content and form. In *the application perspective*, relationships are semantically defined according to the application environment. Relationship definitions include explicit relationship properties and constraints. *The organization perspective* introduces a new set of relations specifically developed to maintain conformity of conceptual abstractions with the nature of information abstractions implied by user requirements throughout the organization. Such relations are based on the stratification of work hierarchies, defined elsewhere in the thesis. Finally, an example of an application of the proposed approach is presented to illustrate the applicability and practicality of the modelling approach.

DATABASE DESIGN; CONCEPTUAL MODELLING; ABSTRACTION HIERARCHIES

DATA SEMANTICS; LARGE ORGANIZATIONAL DATABASES

Acknowledgement

I am grateful to a number of people for being of tremendous help throughout the period of this research. But most of all, I am thankful to God the All Mighty for enabling the successful completion of this work.

I am indebted to HRH Prince Naif ben Abdalaziz, the Saudi Arabian Minister of the Interior, for all the fatherly support and generosity he kindly extended to me throughout my educational pursuit. His Royal Highness has always been a source of inspiration and encouragement.

For their help and support, I thank My brother Othman AlShawi; Abdullah AlNasir, the Saudi Arabian educational attachè; and L. J. Camaisa.

I am grateful to my research supervisor Paul A. Golder for all that he has done to help make this work a success. I thank him for his advise, encouragement, and most of all, his kindness.

I am thankful to Wathiq Mansoor and family for their friendship ; Sabjet Singh for his outstandingly helpful character; David Ball for his interesting discussions; and A. Lowton and friends of the five-aside football team for the entertaining games.

To my parents:

Mr. M. M. Alshawi &
Mrs F. A. Alqarawy

Table of contents

Chapter 1
Introduction 10

Chapter 2
Introduction to organizations

2.1 Introduction 17
2.2 Organization theory 18
2.3 Organization structure 21
2.4 Organization management 24
2.5 Organization communication 28
2.6 Organization and information 29
2.7 Conclusion 30

Chapter 3
A review of database design methods and modelling techniques

3.1 Database architecture 32
3.2 Database design methodologies 34
3.3 Data modelling 43
3.3.1 Major data models 44
3.3.2 Semantic data modelling 46
3.3.2.1 The entity-relationship model 48
3.3.2.2 Other data models 59
3.3.3 Organization-oriented data models 61
3.4 Conclusion 63

Chapter 4
Organization foundations: The stratification of structure hierarchies

4.1 Structure hierarchies and conceptual abstractions 66
4.2 Established methods of stratification 66
4.2.1 Functional hierarchies 67
4.2.2 Authority and responsibility hierarchies 69
4.3 Alternatives to conventional hierarchies 72
4.3.1 Levels of perceptual abstractions 73
4.3.2 The Work-levels approach 76
4.4 Conclusion 79

Chapter 5
Semantic foundations: Semantics and conceptual modelling

5.1 Semantic foundations 81
5.1.1 Perspectives on semantics 83
5.1.2 Semantic Foundations 87
5.1.2.1 Meaning, designation, and significance 88
5.1.2.2 Reference and extension 91
5.1.2.3 Sense and intension 95
5.2 Semantics and conceptual modelling 99
5.2.1 Representation and modelling 101
5.2.2 Conceptualization and categorisation 104
5.3 Conclusion 107

Chapter 6
SECOM: Introduction and methodology

6.1 Introduction 108
6.2 Design methodology 110
6.3 The definition of user views 111
6.4 The level approach and user views 112
6.5 User views in SECOM 115

Chapter 7
SECOM I: The semantic perspective

7.1 Introduction 118
7.2 Principal conceptual foundations 121
7.2.1 Naming 124
7.2.2 Intension 126
7.2.3 Extension 128
7.2.4 Representation 129
7.3 Meaning-based classification (MBC) 134
7.3.1 Group 1: Name-representation inconsistency 142
7.3.2 Group 2: Class inclusion 143
7.3.3 Group 3: Inter-class conflict 144
7.3.4 Group 4: Irrelevant cases 145
7.4 Semantic relations (SRs) 146
7.4.1 Semantic independence (SR1) 147
7.4.2 Semantic inclusion (SR2) 147
7.4.3 Semantic intersection (SR3) 149
7.4.4 Prevalent convergence (SR31) 151
7.4.5 Atypical convergence (SR11) 153
7.5 Relation identification process 154
7.6 Semantic abstractions 155
7.7 Conclusion 161

Chapter 8
SECOM II: The application perspective

8.1 The semantics of relationships....163
8.2 Relationship definition....168
8.2.1 Relationship extension....169
8.2.2 Structural properties....170
8.2.2.1 Participation....171
8.2.2.2 Cardinality....173
8.2.2.3 Participation dependency....175
8.3 Relationships and semantic relations....176
8.3.1 Relationships and inclusion relation....177
8.3.2 Intersection and convergence....179
8.4 Conclusion....180

Chapter 9
SECOM III: The organization perspective

9.1 Introduction....182
9.2 Structure levels and database views....183
9.3 Hierarchical-abstraction relations (ARs)....185
9.3.1 Semantic abstractions....186
9.3.1.1 Hierarchy-inclusion relation (AR11)....186
9.3.1.2 Hierarchy-convergence relation (AR12)....189
9.3.2 Derived abstractions....191
9.3.2.1 Partition abstractions relation (AR21)....191
9.3.2.2 Summary relations (AR22)....193
9.3.2.3 Relationship abstractions relation (AR23)....194
9.3.3 Representation abstractions relation (AR3)....196
9.4 Conclusion....196

Chapter 10
SECOM: Application in an academic department

10.1 Introduction....197
10.2 User views....198
10.3 DIS conceptual modelling....199
10.4 SECOM application....201
10.4.1 NIET definition....206
10.4.2 MBC classification....215
10.4.3 Concept integration....216
10.4.3.1 The semantic perspective....216
10.4.3.2 The application perspective....221
10.4.3.3 The organization perspective....223
10.5 Conclusion....227

Chapter 12

Conclusions and recommendations....229

References....232

List of Figures

Figure 2.1 Examples of organization theories in the span of this century 19
Figure 2.2 The open system organization 20
Figure 2.3 Organization theories perspective of an organization 21
Figure 2.4 Managerial activities within view of structure (Anthony, 1965) 26

Figure 3.1 The three level architecture of the ANSI/X3/SPARC 33
Figure 3.2 Database design strategies (Ceri & Vetter, 1988) 35
Figure 3.3 Design components of the practical design methodology (Molina, 1979) 38
Figure 3.4 Structured project life cycle (Meurer, 1980) 39
Figure 3.5 The "adaptable" design methodology (Roussopoulos & Yeh, 1984).. 40
Figure 3.6 The DATAID approach to database design (Ceri, 1983) 42
Figure 3.7 An ERM representation of a manufacturing database (Chen, 1976)... 49
Figure 3.8 Entity abstractions (Schiffner & Scheuermann, 1979) 50
Figure 3.9 Relationship abstractions (Schiffner & Scheuermann, 1979) 51
Figure 3.10 The data type approach (dos Santos et al., 1980): 53
Figure 3.11 The NERM equivalent of an ERM diagram (Wang & Shixuan, 1986) 54
Figure 3.12 The role concept in the ERM (Hawryszkiewycz, 1984) 55
Figure 3.13 The structure of the external schema (Sen, 1982) 62

Figure 4.1 Decomposing functional activities 68
Figure 4.2 Functional interdependence (Thompson, 1967) 69
Figure 4.3 Characteristics and examples of perceptual abstraction levels (Jaques, 1978) 77
Figure 4.4 The work-level approach to hierarchic stratification (Rowbottom & Billis, 1989) 78

Figure 5.1 Semantic classification (Bunge, 1974a) 82
Figure 5.2 The significance triangle (Ogden & Richards, 1938) 84
Figure 5.3 The "Structural differential" approach (Korzybski, 1933) 85
Figure 5.4 The meaning relation 90
Figure 5.5 Significance with relation to meaning 91
Figure 5.6 The semantic relation of representation 104
Figure 5.7 Examples of vertical and horizontal relations between classes (Howard, 1983) 106

Figure 7.1 Current modelling approaches to concept definition and integration... 119
Figure 7.2 The significance triangle (Ogden & Richards, 1938) 121
Figure 7.3 Concept definition and integration according to SECOM 123
Figure 7.4 The construct of a concept 124
Figure 7.5 Meaning components and relationships (Bunge, 1967) 125
Figure 7.6 The relationship between a concept and its representations 131
Figure 7.7 An example of a concept multiple representations 133
Figure 7.8 Outcomes of a one-to-one comparisons of concept components: 140
Figure 7.9 Case classification process 141

Figure 7.10 Case parameters for group 1 142
Figure 7.11 Case parameters for group 2 143
Figure 7.12 Case parameters for group 3 144
Figure 7.13 Case parameters for group 4 145
Figure 7.14 The relation of semantic inclusion (SR2) 148
Figure 7.15 The relation of semantic intersection (SR3) 150
Figure 7.16 The relation of prevalent semantic convergence (SR31) 153
Figure 7.17 The relation of atypical convergence (SR11) 154
Figure 7.18 The process of identifying semantic relations between concepts 156
Figure 7.19 An abstraction hierarchy example (Elmasri & Navathe, 1989) 158
Figure 7.20 The SECOM approach to conceptual abstractions 159
Figure 7.21 The three dimensional perspective of conceptual abstractions 161

Figure 8.1 An example of the category concept (Elmasri et al, 1985) 173
Figure 8.2 An example of a relationship over a defining concept 177
Figure 8.3 The extended version of the relationship 178
Figure 8.4 An example of a relationship over semantic convergence 179

Figure 9.1 User views, structure levels, and levels of linguistic abstractions 184
Figure 9.2 Representations contradicting abstraction rules 187
Figure 9.3 A correction of the representation contradiction 187
Figure 9.4 An example of the hierarchy-inclusion relation 188
Figure 9.5 A case of hierarchy-abstraction conflict over meaning convergence 189
Figure 9.6 Rectifying representation inconsistency 190
Figure 9.7 An example of hierarchy-convergence relation 191
Figure 9.8 An example of a relationship abstractions relation 195

Figure 10.1 Head-of-Department view information 202
Figure 10.2 Admissions Officers view information 203
Figure 10.3 Supervisor view information 203
Figure 10.4 Research-student view information 204
Figure 10.5 Representations within available views 205
Figure 10.6 NIET definitions for Head-of-department view 207
Figure 10.7 NIET definitions for UCCA-admissions officer view 209
Figure 10.8 NIET definitions for Research-admissions officer view 210
Figure 10.9 NIET definitions for Supervisor view 211
Figure 10.10 NIET definitions for Research-student view 213
Figure 10.11 "Person" inclusion abstraction 218
Figure 10.12 "Publication" inclusion abstraction 219
Figure 10.13 "Message" inclusion abstraction 219
Figure 10.14 "Report" inclusion abstraction 219
Figure 10.15 An example of an SR31 220
Figure 10.16 A diagrammatic illustration of the DIS application perspective 222

Chapter 1

Introduction

In the past two decades extensive research into database modelling has been conducted, in which a large number of modelling techniques were developed and improved for the purpose of providing effective tools for representing the reality of the organization in a database environment. Such techniques vary considerably in the scope and extent of their analysis.

A state of uncertainty has developed as a result of the large number of models developed and the degree of contrast among such models in particular. In addition, a clear direction for research into the area is lack lacking.

It is evident upon evaluating current modelling techniques that such techniques are focused on specific aspects of the modelling process. Individual methods are usually devised to offer particular solutions to particular problems. As a result, a comprehensive approach to modelling databases is lacking.

The lack of a holistic approach to the design process leads designers to incorporate a number of methods, which are not necessarily homogeneous, into one main methodology. This usually results in the need to compromise certain aspects of the contributing methods in order to achieve an overall approach.

While it is acceptable that data models have been developed to resolve specific design obstacles and/or improve certain aspects of the process itself, such methods do not take into consideration a wide perspective of the design process. Issues of requirements definition, data semantics, information relevance, the design and integration of user views, concept abstraction, classification, and categorisation are usually presented within independent contexts. As a consequence, the successful utilization of a certain design tool is largely dependent upon its implementation in a specific design context, which is essentially the narrow view of the design process as perceived by the method incorporated.

Because of the limited scope of available modelling approaches, it has not been possible to examine the likelihood of interdependence among various design problems. As a result, the possibility of offering more fundamental and comprehensive problem solving mechanisms is diminished.

Existing modelling literature stresses the importance of incorporating the semantics of data into the structure of data models. Nonetheless, there is not a clear definition of data semantics which would provide a frame of reference for developing improved data models, and a method for evaluating a model's semantic strength. This has come about because of the lack of proper understanding of semantic theory in general, and the impact of semantics on database modelling in specific.

In addition, existing modelling techniques do not give appropriate consideration to the organization content of a database. Instead, such aspects are usually left for organization analysts. As a result, a gap has developed between the technical and the organization perspectives of the database design process.

It is contended, here, that until a proper investigation of the fundamentals of database modelling is pursued, and until all issues with relevance to the use of data, and the design of the database are taken into account, tools will be continually developed to deal with issues that are not at the centre of the problems they set out to resolve.

An original approach to examining database modelling is proposed in this thesis, in which the conceptual model is utilized as the main design tool. The originality of the approach is attributed to the broad investigation on which it is based, and the new direction it takes towards examining database design problems. The scope of this investigation includes conceptual modelling, organization theory, and semantic theory. Each of such areas has been found to be of direct impact on data representation in specific, and conceptual modelling in general.

The suggested approach differs from established research directions in that it offers a new perspective on the issues involved in designing databases. Current methods are seen, in this work, as developing out of the gradual progression of modelling research which at its early stages was greatly influenced by technical limitations. In other words, the development of modelling research has not gained the necessary independence from technical specifications. Database modelling research will benefit immensely, in the sense of examining the full extent of utilizing data modelling as a design tool, but only when the investigation is approached from an implementation independent perspective. This direction is in accordance with the view which advocates the need to pursue new strategies towards designing information models.

The modelling approach suggested in this thesis will introduce a new semantic definition of concepts and concept relations. Such a definition emphasizes the importance of relying on concept meaning for concept classification and integration. The importance of concept name and representation is therefore diminished and is considered a property of the meaning of the concept itself. By doing so, the approach permits multiple names and representations of the same concept. As a consequence, view inconsistency, rising as a result of multiple views over similar concepts, is resolved. In addition, conceptual abstractions are formed according to meaning instead of representation, which results in the creation of intrinsic conceptual abstractions, in the sense that such abstractions are based on content instead of form. By relying on content, i.e., meaning, for abstraction construction, concepts with no attributes in common are indirectly linked through concept meaning, thus contributing to the improvement of database integrity by ensuring proper updates.

The proposed approach will also examine the impact of organization structure on information representation and data abstraction. The examination will focus on the reasons underlying the nature of information representation throughout the organization. In other words, the relationship between users and the degree of information abstraction relevant to each user is specified. This relationship is then utilized to manage representation abstractions. More specifically, the approach will introduce new relations (abstraction relations) which are specifically designed to maintain the consistency of information representation according to the relevance of information to users; i.e., detailed and specific information at lower organizational levels, and abstract and general information at the higher ones.

The conceptual modelling approach is developed in three main steps. First, the semantic perspective, which presents semantic definitions of concepts and relations. Second, the application perspective, which presents the relationships between concepts based on their usage within the application environment. Third, the organization perspective, which presents abstraction relations to handle representation abstractions rising as a result of the organization structure.

The examination of the issues involved in this thesis was carried out according to an established research method. Research methods are widely discussed in the literature. Jenkins (1985) described a number of research methods which include: mathematical modelling, experimental simulation, laboratory experiment, free simulation, field experiment, adaptive methodology, field study, group feedback analysis, opinion research, participative or action research, case study, archival research, and philosophical research.

Because of the nature of the research undertaken, and after examining the criteria for selecting an appropriate research method provided by Jenkins (1985), it was concluded that the philosophical approach is most suited. According to Jenkins, the philosophical research method "defines a purely mental pursuit. The researcher thinks and logically reasons causal relationships. The process is intellectual and the aim is for the flow of logic to be explicit, replicable and testable by others."

The approach suggested in this thesis has come about as a result of utilizing available organization literature on organization stratification, and developing a modelling-specific semantic framework out of literature on semantic theory. Therefore, the thesis is organized as follows:

Chapter 2 presents an introduction to organizations with emphasis on major organization concepts. The purpose of this introduction is to illustrate the extent of the impact of organization structure on information and database in general, and conceptual modelling in specific. It is worth noting here, that the impact of organization on information is recognized in information systems literature, nonetheless, it has not received appropriate considerations in database design literature.

Chapter 3 presents a review of main design methods. In addition, important modelling techniques are reviewed to reflect the direction and the state of data modelling research. Organization oriented techniques are outlined in this chapter to illustrate the inadequacy of such techniques in dealing with organization issues in the modelling and design of databases.

Chapter 4 will examine the organization hierarchy for the purpose of achieving a point of reference between the organization and the database. In other words, the impact of organization structure on the database is precisely defined in order to incorporate such a definition into the new approach. Because the organization hierarchy is the main feature of its structure, and since structure is identified in chapter 2 as of critical consequence on information representation, chapter 4 will examine alternative structure hierarchies for the purpose of arriving at specific criteria for relating user information requirements to user position in the structure. Such relations will then be used as a guide for integrating user views.

Chapter 5 presents semantic theory and examines the relation between database modelling and semantics in general. In addition, this chapter will set the semantic foundations underlying the proposed modelling approach, and introduce the basic definitions and terminology utilized in subsequent chapters.

Chapter 6 introduces the proposed approach. In the introduction, the main concepts and a general description of the proposed approach are presented.

Chapter 7 presents the semantic perspective of the proposed approach. This chapter introduces basic concepts definitions. Such definitions are then utilized to identify semantic conceptual classes. A process which handles the creation of such classes is then described. The semantic relations between concepts are then presented, followed by a description of the process for identifying such relations. Finally, semantic abstractions are examined in view of concept definitions and relations developed in this chapter.

Chapter 8 presents the application perspective. In this chapter, the semantics of relationships between concepts as implied by the application environment is examined. In addition, relationship properties are presented with view to the semantic relations defined in chapter 7.

Chapter 9 presents the organization perspective. This chapter provides a detailed examination of the relationship between organization hierarchy levels and database views. A new set of relations is then developed to manage view inconsistencies arising as a result of the impact of hierarchic levels on user information.

Chapter 10 presents an example of SECOM application. This chapter will illustrate the execution of the conceptual modelling operations.

Chapter 11 concludes this thesis. The strength and weakness of the suggested approach are outlined, in addition to recommendations for future research and developments.

Chapter 2

Introduction to organizations

This chapter presents a review of major organization concepts for the purpose of understanding the database environment. In addition, this chapter will define the impact of organization structure on information requirements and representation. Such definition will provide the framework for identifying the nature of the relationship between database users and data abstraction and representation.

2.1 Introduction

Although the term *enterprise* is used more frequently in database design literature, this thesis uses the term organization instead. While enterprise usually refers to an economic unit, the definition of an organization is more general for the reason that it includes non profit organizations such as government agencies. Accordingly, this research signifies organization structure and behaviour as representative of those in all types of social and economic entities.

It has always been challenging to ascertain the precise definition of the term *organization*, and this is simply because of its wide encompassing context. The definition of an organization, in this work, is educed from the following definition:

As mentioned previously, one of the main objectives of the database design approach proposed in this work is to delineate the wider perspective of information as being an integral component of organizations. It is strongly felt, therefore, that information has to be viewed as part of a system. Moreover, it is incumbent upon information specialists to ensure that concepts, principles, and policies underlying the information system are compatible with those of other functions and the organization as a whole. Because of its immediate relationship to the information system, the database has to be viewed by designers as being part of a more complex system than that of the information system; and only then will it be possible to construct tools that are capable of accounting for and handling design problems in a comprehensive and integral fashion.

2.2 Organization theory

Organization theories offer tools for understanding and analysing the structure and behaviour of organizations. Since the beginning of this century, scientists from various disciplines have contributed to the field of organization theory by offering their analysis of organizations from within their respective fields, and contributing what, in their view, are appropriate tools for problem definition and solving.

Three schools of thought are acknowledged as distinct categories for examining the structure and design of organizations: the classical, human relations, and systems theories. Figure 2.1 depicts the three theories with respect to the time of their evolution, as well as examples of each theory.

Classical theory

The classical theory is the traditional school of thought which established organization analysis through structure. According to this theory, all organization activities are based on the division of functions and the arrangement of people, which in essence, are the two principle contributors to the effectiveness of any organization. The underlying concepts of the classical theory are still upheld throughout the theoretical and practical persuasions of organization study.

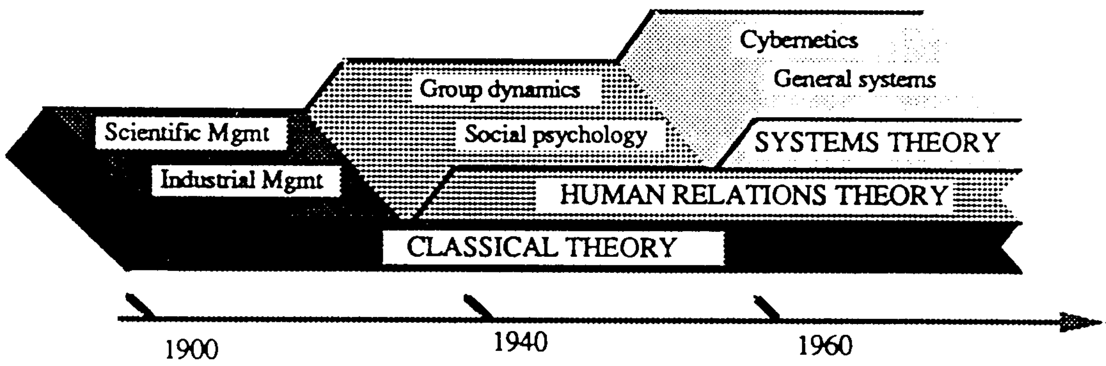


Figure 2.1 Examples of organization theories in the span of this century

Taylor (1947) is accredited with the introduction of the classical theory. The views on organizing, presented by Taylor, emanated from his observation of management and employees in his work environment (Nord, 1976, Ford *et al.*, 1988). Taylor's *scientific management* approach is based on four primary concepts: *division of labour*, *scalar and functional processes*, *structure*, and *span of control* (Scott, 1967). Such concepts are the basis of organization structure in general, and therefore, will be further discussed in section 2.3.

Human relations theory

The human relations theory offers an alternative approach to analysing organizations by incorporating relevant principles of behavioural science. The impact of the human behaviour on work conditions accounts for more than what was implied by the classical theory. Where the classical theory placed emphasis on the structure and co-ordination of functional components, the human relations theory targeted the integration of human behaviour aspects with structure issues.

Systems theory

The systems approach is "a way of thinking which enables us to cope with complex phenomena by identifying their systemic relations" (Elliott, 1980). Systems theory views an organization as an open system in which the environment plays the major role of providing input consisting of new demands, feedback, and environmental constraints on the organization and its individuals (figure 2.2).

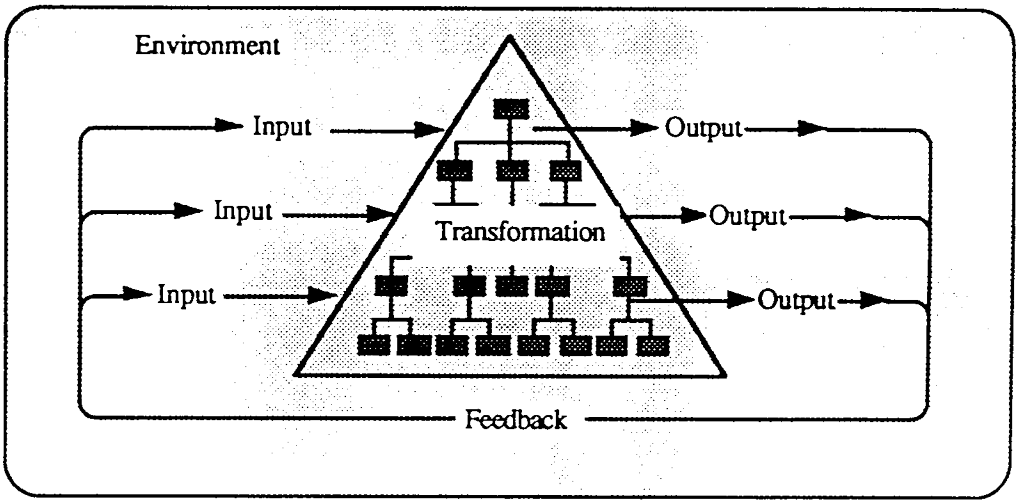


Figure 2.2 The open system organization

Organization theories reflect different perspectives over the same whole, and present different methods of organization analysis. The differences, however, do not suggest conflicting views, nor do they suggest mutual exclusion among the three approaches. In reality, each approach constitutes an outstanding tool for organization analysis and design. Figure 2.3 illustrates the organization as viewed by the different theories.

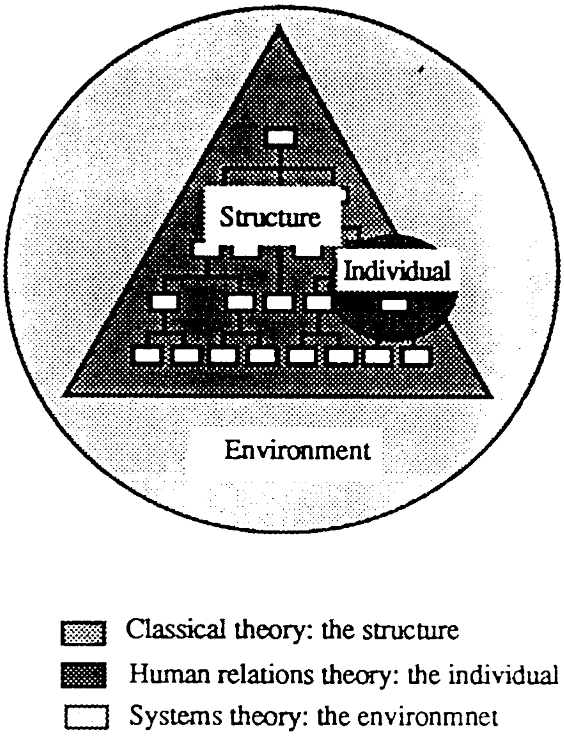


Figure 2.3 Organization theories perspective of an organization

2.3 Organization structure

Although often discussed, organization structure is a topic of contention among various authors on the subject. The main disagreement revolves around the definition and components of system structure. While some authors identify structure as the description of the current state of the system, others point out the processes which led to the existing state of the organization, as being the real structure. Instead of defining structure, a description of prominent organization characteristics will be presented.

The organization hierarchy

The organization hierarchy is the most prominent feature of its formal structure. It comes about as a result of the delegation of authority and responsibility, and consequentially results in the traditional management pattern of the few supervising the many.

The organization hierarchy presents a number of vertical and horizontal segments. Vertical clusters reflect the division of organization activities, which are usually grouped under department names; while horizontal layers reflect the superior-subordinate relationship.

The hierarchy was described by Weber (Gerth & Mills, 1946) as characteristic of all forms of organizations, and was described as follows:

> "The principles of office hierarchy and of levels of graded authority means a firmly ordered system of super- and subordination in which there is a supervision of the lower offices by the higher ones....The principle of hierarchical office authority is found in all bureaucratic structures: in state and ecclesiastical structures as well as in large party organizations and private enterprises."

Authority, power, and control

Although the organization hierarchy is attributed to a number of factors, it is the line of authority that has the most influence on its shape. Authority is the power vested in people with respect to their position in the structure. It enables members of the organization to discharge their responsibilities among those under their command. Davis (1951) described authority as:

The term authority is often associated with the concepts of power and control. Authority in some cases is addressed as a form of power associated with a position independent of personnel traits.

Power, on the other hand, refers to the ability to affect a situation by human action (Gross, 1968). Unlike authority, power may be gained by properties other than those granted by a position in the formal structure. For example, a person may retain power because of his economic status, personal relationships, and/or experience.

Power and authority are the mechanisms by which an organization assures control. Control, on the other hand, is the process of utilizing and directing resources to secure a state of stability. The exercise of control is dependent upon the possession of power, which enables a manager to influence others, by motivating means or otherwise, to perform assigned activities according to existing plans (Child, 1984).

Division of work

Work is divided according to organization activities. According to Litterer (1965), work divisions are based on factors such as efficiency, complexity of work, and variety of requirements.

Scanlan & Keys (1979) pointed out that in any organization, regardless of its type, three activities are central to its continuity. First, an organization produces a valuable utility primary to its existence, e.g., a product in a manufacturing environment, teaching in a university, services for the public in government, etc. Second, it has to make its product(s) available to customers through sales, merchandising, and distribution. Finally, an organization must have the means to finance the first two activities, which involves assets acquisition and record keeping. Scanlan & Keys argued, that all other organization functions are but extensions utilized for the purpose of carrying out one or more of such major activities.

Many authors agree that function, product, and location divisions are the dominant methods of organization division. In addition, the matrix organization has been suggested as an alternative to established methods of departmentation. The matrix structure introduces a slightly different approach from that of the classical hierarchical structure in that the line of authority is altered to suit specific working circumstances. This form of organizing usually arises when it is determined that the successful completion of certain projects is better served by introducing an alteration of the existing chain of command, e.g., establishing alternative communication channels and lines of authority.

2.4 Organization management

Managing is the task assigned to individuals in an organization to ensure efficient co-ordination among the various parts of the organization, and maintain integrity and stability in a changing environment. Because of the variety in managerial activities, the management process has always been hard to precisely define. This fact is reflected by the wide spectrum of suggested definitions of the management process of which the following are given as an example,

Management can be better understood by examining the components or functions of the managerial process. A number of functions have been suggested as fundamental to the managing process (Harisson, 1978), and most are a combination of planning, organizing, staffing, directing, coordinating, reporting, budgeting, controlling, objective setting, and plan formulating activities.

Early organization theories set the grounds for effective management techniques. Taylor's scientific management approach suggested that sound management practices are ones based on the understanding of an organization as an integrated structure. Behaviourists, on the other hand, indicate that the effectiveness of the managerial role is dependent on the understanding of the social and psychological issues underlying worker frame of mind. System proponents indicated that management is the process of applying scientific techniques to a problem situation, and therefore, makes use of management science and operations research techniques to reduce uncertainty and resolve conflicts.

Management levels

As demand increases to provide more services and manufacture more products, the organization is compelled to expand its work force and diversify its endeavour. As the number of workers increases, more managerial positions are required to provide supervision and maintain control, resulting in the creation of additional managerial levels.

Managerial levels are determined by authority in an organization. The levels imply the complexity of assigned tasks throughout the organization. But more important, managerial levels are the stages at which the nature of information representation shifts from detailed to general and from specific to aggregate, down and up the organization structure.

Brown (1971) pointed out an essential function of managerial levels. The levels, it is suggested, are methods of distributing responsibility among employees. Therefore, each manager is responsible for work carried out in the sub-hierarchy under his responsibility. The ultimate responsibility, i.e., the responsibility of all actions taken by members of the organization, lay under the top-executive's scope of responsibility.

"In an employment hierarchy, each ascending stratum includes in its degree of accountability for work, the work of the subordinate stratum below it, until finally the chief executive is accountable for the entire work of the hierarchy. This leads to the notion of considering an executive hierarchy as a series of ascending orders of abstraction concerned with work." (Brown, 1971)

Levels of managerial activities

As mentioned earlier, the managing process involves extensive planning and supervision. According to Anthony (1965), management is carried out at three levels corresponding to the type of activities performed in the organization (figure 2.4).

At the higher levels of the organization pyramid, managers perform activities related to policy formulation, long term planning, and goal setting. This management level is referred to as the *strategic planning* level.

The next management level is concerned with administering current organization activities, such as planning working capital, formulating advertising programmes, and deciding on research projects. Because both planning and control are central to this level, it is labelled *management control*.

The third level is *operational control*, and it is the level in which the tasks carried out have little or no decision making requirements such as inventory control, scheduling, and personnel appraisal.

Classifying management levels according to activities implies the nature of information required in each management level. Higher levels of management require general and abstract information, while lower levels' information requirements are usually more detailed and specific.

It is essential to point out that these levels of managerial activities are different from those of management levels addressed in the previous section. Where as the former is a classification of management according to activities performed in the organization, the latter is a definition of management levels according to status in the organization hierarchy. It is easier to decide on the level of management a manager belongs to according to status, position, and authority. But it is not as obvious to ascertain a manager's level according to the activity(s) performed by the manager, simply because typifying an activity is a difficult task.

Decision making

Decision making is the process of selecting a course of action among available alternatives. In some cases, organizations are viewed as "an extension of individuals making choices and behaving on the basis of their understanding of their environment and their needs" (Pfiffner & Sherwood, 1960). But because managers are assigned the task of making the more critical of the organization decisions, decision making is usually distinguished as a managerial task.

A number of factors are critical to the success of decision making, and the most important is the awareness of the alternatives. It is not unusual for a manager to overlook certain alternatives simply because of existing prejudice based on past

The third level is *operational control*, and it is the level in which the tasks carried out have little or no decision making requirements such as inventory control, scheduling, and personnel appraisal.

Classifying management levels according to activities implies the nature of information required in each management level. Higher levels of management require general and abstract information, while lower levels' information requirements are usually more detailed and specific.

It is essential to point out that these levels of managerial activities are different from those of management levels addressed in the previous section. Where as the former is a classification of management according to activities performed in the organization, the latter is a definition of management levels according to status in the organization hierarchy. It is easier to decide on the level of management a manager belongs to according to status, position, and authority. But it is not as obvious to ascertain a manager's level according to the activity(s) performed by the manager, simply because typifying an activity is a difficult task.

Decision making

Decision making is the process of selecting a course of action among available alternatives. In some cases, organizations are viewed as "an extension of individuals making choices and behaving on the basis of their understanding of their environment and their needs" (Pfiffner & Sherwood, 1960). But because managers are assigned the task of making the more critical of the organization decisions, decision making is usually distinguished as a managerial task.

A number of factors are critical to the success of decision making, and the most important is the awareness of the alternatives. It is not unusual for a manager to overlook certain alternatives simply because of existing prejudice based on past experience. Another important factor is the information available which will enable the decision maker to evaluate alternatives in view of expected consequences. Moreover, a manager must be aware of the rationale underlying the decision making environment.

2.5 Organization communication

Communication is a process essential for the survival of all types of organization. It is the process in which changes in the system environment, as well as changes within the system, are channelled to the appropriate parts of the structure in order to account for evolving circumstances. Pace (1983) asserted that communication is "both the product of and the producer of action."

Understanding human communication requires an understanding of the nature of human behaviour. According to Katz & Kahn (1966), communication is the "social process of the broadest relevance in the functioning of any group, organization, or society". They indicated that "communication - the exchange of information and the transmission of meaning - is the very essence of the social system or an organization."

According to Lewis (1980), for communication to realise its role as an effective tool for information transmission, it must must be coherent, unambiguous, and compatible with the situation in which it is required.

Schneider *et al.* (1975), pointed out a communication issue that is of significant relevance to this work. It is suggested that the language used in sending messages is the key to the effectiveness of organization communication.

The relevance of information to the user is an important criterion for evaluating the effectiveness of the database. This issue will be explored further in subsequent chapters, in order to explain how the suggested design approach deals with this problem area, and contributes to the effectiveness of database utilization by ensuring information relevance, and in the end improves communication.

2.6 Organization and information

The impact of organization on information is widely addressed in information systems literature. Information system design methods take into consideration issues of organization in the development of the information system. Nonetheless, such issues are not included in database design methods.

Extensive work has been done to examine the role of information in an organization. It is recognized in the literature that the relationship between information and organization is bidirectional.

It is plain that information is critical to the survival of an organization. It is also known that organization structure impacts the nature and flow of information. For the purpose of this work, a brief outline of the impact of structure on information requirements is presented.

The review of organization management discussed the distinct levels of managerial activities. The information required to perform managerial activities possesses characteristics relative to the level in which such activities are performed. The most relevant of such characteristics - to this work - is that information required for performing higher level activities are general and abstract in nature; while lower level managerial activities require information that is detailed and specific. Accordingly, the structure of the information system is hierarchical in nature, and such hierarchy is rooted in the organization structure hierarchy.

The information hierarchy is composed of data arranged in hierarchical abstractions. In other words, data required for higher organization levels are abstract data with less detail. Data required at lower levels are detailed and specific. Therefore, there is a relationship between organization structure hierarchies and data abstractions. This relationship will be examined in this work for the purpose of constructing conceptual abstractions that are reflective of the needs of activities throughout the organization hierarchy. By devising such abstractions in the suggested approach, information relevance is immensely improved.

2.7 Conclusion

This chapter presented a review of basic organization concepts. The review provides an illustration of the database environment for the purpose of appreciating the impact of major organization concepts on information utilization and representation. In addition, this chapter presents the basis for examining further organization structure concepts (chapter 4) which will then be utilized in the proposed approach (chapter 9).

Chapter 3

A review of database design methods and modelling techniques

Database design is a process comprising a number of steps in which tools and techniques are employed to simplify the process and help designers identify and resolve design problems. Steps in the design process are grouped under two major phases, logical and physical design. This research is concerned with the former.

Logical design is the process of defining information requirements and creating a database model, independent of physical specifications, which reflects objects in reality. Many techniques have been developed to aid designers in identifying, collecting, and documenting user information requirements. Similarly, data models have been developed to serve as tools for representing reality in a form that is suitable for implementation.

Tools and techniques are utilized by designers within a particular approach specified in a general plan which suggests the course of actions to follow. The plan, or methodology, is developed in accordance with design requirements for the purpose of achieving a comprehensive design.

In this chapter, a review of database design methodologies and modelling techniques is presented. The review will include prominent publications that are representative of the literature and of immediate relevance to the issues discussed in this thesis. But before proceeding with this review, a brief introduction to database architecture, which forms the basis underlying design components, is presented.

3.1 Database architecture

When databases first came in use, data access was a task confined to database professionals. As database research progressed, Data Base Management Systems (DBMSs) were developed to provide users with their information requirements with as much ease and flexibility as possible.

Although earlier work addressed the issue of database structure (Meltzer, 1969, CODASYL, 1971, Senko *et al.*, 1973), simplification of database interface and access mechanisms was greatly enhanced by the three level database architecture introduced by the American National Standard Institute's Standards Planning Requirements Committee (ANSI/ SPARC). In their report (ANSI/X3/SPARC, 1975), the committee described three realms of interest to information, namely: real world, ideas about the real world as perceived by humans, and symbols chosen to represent these ideas on storage medium. In order to represent information realms in a data processing environment, the following three levels of data processing were defined along with mapping functions between the three levels.

The *external schema*, consisting of a number of individual schemas, defines an application view of the database in which objects of interest to that application are represented. The *conceptual schema* level is where all objects of interest in the database are represented. While normally there are many external schemas, there is only one conceptual schema. The *internal schema* level is where objects containing the stored data - defined in the conceptual model - are represented. Figure 3.1 depicts the ANSI/SPARC three level architecture.

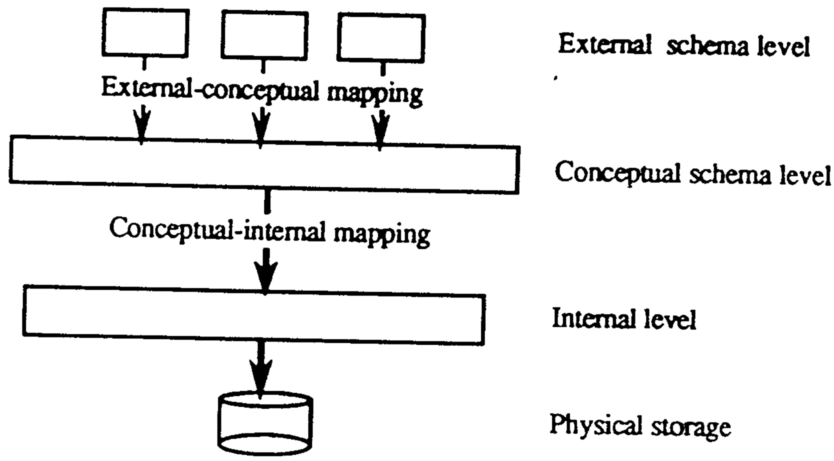


Figure 3.1 The three level architecture of the ANSI/X3/SPARC

The ANSI architecture gained wide acclaim among database researchers for the many advantages stemming from its promotion of data independence. Because users deal with the DBMS at an external level, their information requirements are represented at an external level, or local schema, and in this sense, data organization and storage specifications are isolated. In a data independent structure, changes in the database environment can be easily accounted for in the database without having to deal with details of storage structure. Another advantage of this architecture is that it permits and promotes model coexistence. Because certain data models are suitable for implementation but lack the capability of semantic representation, different models can be employed to represent information at different levels of the design process.

The ANSI architecture went through a number of revisions outlined in a number of consecutive reports published by the committee, as well as attempts by others to improve the architecture through extensions. In De *et al.* (1981), for example, a proposed extension is presented in which the conceptual schema is split into two levels, thus presenting a four level architecture. Beside the external and internal levels, the second and third levels are the enterprise schema as defined by Chen (1977); and the canonical schema which is regarded as "a data structure realization of the enterprise schema" (De *et al.*, 1981).

3.2 Database design methodologies

A design methodology is a specified set of design steps associated with appropriate tools with the common objective of providing designers with a complete set - usually variant in degree of detail - of rules and guide-lines leading to an effective database design.

Design methodologies are usually classified according to the strategy followed by the methodology. Ceri & Navathe (1988) identified the following design strategies: top-down, bottom-up, inside-out, and a mixed strategy (figure 3.2).

In top-down abstract concepts are first defined, then refined to include atomic elements using suggested "context-free, top-down primitives." Bottom-up is opposite to top-down, where elementary concepts are defined first, then integrated, using bottom-up primitives. Inside-out is a disciplined bottom-up in which information flow is used as a guide in the process. In other words, specific information flow is composed to form a more integrated representation. Finally, a mixed strategy is a combination of the above strategies.

Many methodologies have been proposed in the literature, some are more specific than others, and most claim completeness. But it is clear that no single methodology is advantageous over the others, but some are more suitable in certain environments than others. In the remainder of this section, a review of selected design methodologies, based on diversity as well as significance, is presented.

Yao *et al.*'s (1978) approach to database design suggested the following five steps to producing a logical database. *Requirements analysis*, the first step, involves the identification, collection and documentation of user local application requirements

Figure 3.2 Database design strategies (Ceri & Vetter, 1988):
(a) A top-down strategy; (b) A bottom-up strategy; and (c) An inside-out strategy

according to *information and process structure* input specifications presented by Kahn (1976). The output of this step is a formal documentation of user requirements which serve as an input to the following step. The second step is *view modelling* in which local user views are separately modelled using suggested modelling techniques such as Bachman's data structure diagrams (Bachman, 1969), the entity-relationship model

(ERM) (Chen, 1976), the relational model (Codd, 1970), and data abstractions (Smith & Smith, 1977a). The third step is *view integration* which involves merging local views and removing inconsistencies which usually arise as a result of conflicting views over data. The fourth and fifth steps are *view restructuring*, and *schema analysis and mapping*, and are DBMS dependent. The presented approach provides a simple break down of the design process resulting in clearly defined phases. However, the methodology lacks detail, and does not provide facilities for integrating design phases.

Structured logical database design methodology (LDDM) (Kahn, 1979) is a methodology with the objective of producing what the author refers to as a "good" logical database. It is good in the sense that it consists of "step-wise mappings" which can easily be understood by designers. LDDM describes a number of processes in logical database design, they are: real world requirements, logical information structure, global information structure, entity structure, logical entity structure, and logical database structure. Each process is mapped to a step in the methodology, and each step is performed through activities and sub-activities, leading to the completion of a design level.

Steps in LDDM include the following. *Requirements* step for defining information and processing requirements; *entity* step in which entities are abstracted and classified; *relationship* step aggregates relationships defined in requirements step; *Entity structure* step transforms the global information structure through normalization and synthesis into a DBMS independent structure; *refinement* step in which consistency is maintained and redundancies reduced; and last, a *DBMS accommodation* step in which the global information structure is mapped into an appropriate DBMS. The strength of this methodology is in its specificity and integrative approach to the design process.

Another approach towards an integrated methodology is demonstrated by Sundgren (1978), in which the importance of understanding the purpose and role of the database within the wider context of organization and social systems is emphasized. The main theme in this methodology is reflected by the statement that the database "..must never be an end in itself [but rather] should be a tool that helps one or more groups of people - the so called end users or information consumers - to control certain phenomena in the real world".

The process is carried out in two phases, *infological* and *datalogical*. The infological (user-oriented) phase, is where user requirements and reality of database environment are defined and modelled in a structure consisting of concepts and input/output flow. The datalogical (computer-oriented) phase, is where the information structure specified in the infological phase is transformed into a file structure, which is then mapped into designated implementation model.

Although Sundgren pin-points a design aspect that has been overlooked in database research, i.e., database systems within the wider social and economic systems, the approach itself is too general and lacks specific solutions.

Molina (1979) suggested that in their attempt to improve database usability, designers tend to neglect significant database fundamentals. Because the importance of analyzing real world and information realms is distinctly over emphasized, the author asserts, database implementation is bound to encounter obstacles caused by overlooking the database within the data realm, its natural context.

The design process, in this approach, is composed of two main processes, *logical* and *physical*. Unlike many design methods in the literature, this methodology considers the task of determining information requirements as being a pre-design phase in which designer involvement should be limited. The logical design process is carried out in two phases. First is the creation of a data structure, similar to a data model, exhibiting the relationships among data elements. The second phase is the grouping of data elements into data files, and is called logical file design. The two design phases are then broken-down into further sub-phases in which inputs and outputs are mapped into the database structure.

The highly graphic methodology offers a relatively detailed sequence of actions, with provisions for mapping between design levels. Figure 3.3 charts major design components.

Figure 3.3 Main design components of the practical design methodology (Molina, 1979)

Structured systems analysis and design techniques have gained wide acceptance in information systems literature. Meurer (1980) described a design process which, he claims, utilizes such techniques to develop a database supporting business functions within a fraction of the time and cost of traditional system development techniques. The design process involves three phases within a project life cycle. *Structured analysis*, which requires the designer to "study and model the business information flow using a top-down approach and pictures rather than text to describe that flow." The output of this phase consists of process data flow diagrams and data stores, and the interactions between the two. *Structured design,* the second phase, is one in which modelled business systems, programs, and modules are hierarchically coordinated. Finally, *structured implementation* concerns programming and testing techniques to implement structure design specifications. The methodology emphasizes the importance of creating a business model, and supports documentation throughout the design process. Figure 3.4 illustrates the structured project life cycle.

Figure 3.4 Structured project life cycle (Meurer, 1980)

Tucherman *et al.* (1983) presented a modular database design methodology. Although the idea of modular design has been discussed elsewhere (e.g., Weber, 1979; dos

Santos *et al.*, 1980), the "Pragmatic" approach to the design process provides sound theoretical bases for modular specification and construction mechanisms, according to the authors. The design process is carried out at two levels. The *specification level* is one in which the definition of a module is provided with mechanisms for its construction. The *representation level*, on the other hand, involves mapping modules from the specification level into a specific database implementation.

Roussopoulos & Yeh's approach (1984) is based on the assumption that "conceptual modelling can be based on how information flows between an enterprise and its environment and among its components." The approach emphasizes the importance of scrutinizing the enterprise environment, and analyzing and specifying the operational behaviour of the enterprise (figure 3.5).

Database Schema

Figure 3.5 Phases in the "adaptable" design methodology (Roussopoulos & Yeh, 1984)

In *environment analysis*, information relating to activities within the enterprise are collected using conventional information collection techniques, e.g., reviewing existing documents, conducting interviews, and distributing questionnaires. The output of this step is a list of enterprise activities and operations in the form of information flow diagrams. *System analysis and specification* entails hierarchical detailing of each enterprise activity into independent tasks and sub-tasks so that each sub-task could be individually examined to determine the data elements involved. Information regarding each activity, sub-activity, task, and sub-task are documented within the framework of the hierarchical analysis. In *conceptual modelling*, a conceptual schema is developed for each activity defined in system analysis and specification.

In order to design conceptual schemas, the methodology lists the criteria for selecting a suitable modelling technique. The criteria includes degree of semantics, ease of understanding by non designers, and availability of conceptual schema manipulation facilities. Modelling techniques recommended include the ERM (Chen, 1976), the semantic data model (Hammer & Mcleod, 1978), aggregation and generalization abstractions (Smith & Smith, 1977a, 1977b), and Taxis (Mylopoulos & Wong, 1980). The final step, *logical schema design*, is part of the physical design process and is concerned with translating the conceptual schema into a specific DBMS.

The DATAID design methodology (Ceri, 1983) covers the database design process in five steps concurring with the general four phases of database design, i.e., requirements collection and analysis, conceptual design, logical design, and physical design, as illustrated in figure 3.6.

In *requirements collection and analysis*, a natural language reasoning system (NLDA) for analyzing database requirements is employed to define information, operations, and events

Figure 3.6 The DATAID approach to database design (Ceri, 1983)

requirements. Glossaries of data and their properties, operations, and events affecting organization functioning are compiled. In *view conceptual design*, individual conceptual views are defined and application schemas are constructed for each of the glossaries defined in the previous phase. This is performed using a suggested "enriched" extension of the ERM and its graphical representation in which abstraction hierarchies, aggregation, repeating attributes, and identifiers are incorporated. In *view integration*, integrated schemas incorporating all applications are constructed for each of the data, operations, and events schemas. This involves resolving conflicts and inconsistencies among individual schemas. In *logical design*, global conceptual schemas are converted into a specific DBMS data model, and in this case the relational and network models are chosen as target systems. The final step, *physical database design* addresses DBMS mapping.

3.3 Data modelling

Data modelling has been an area of interest in scientific research for many decades. Researchers from various areas of science have been developing and using data models to analyze and solve complex problems. In the field of database design, data models are used to reflect objects and relationships of an application environment, in a representation that is easy to understand and manage.

Many data models have been developed in the last two decades. Earlier data models were utilized for defining data structures and database schemas. But as data independence became an important factor in database design, data models began to assume the new role of being tools for data representation and manipulation.

Since computers deal with tokens and symbols, and unlike humans who deal with the meaning of such symbols, data models have to express the meaning of data as perceived by humans in a form understandable by both designers and users, as well as produce a set of implementable data structures. In other words, data models have to include the semantics of the database environment in their representation, in a fashion that is easy to understand and use.

In this section, data models will be considered under two headings. First, major data models will be briefly reviewed, then a closer examination of other prominent data models, which are referred to as semantic data models will be presented. Note that the grouping of data models under their respective titles is in accordance with a reference convention used in the literature. Such convention, though convenient, does not reflect an accurate, or even an appropriate method of classification. This is simply because such reference suggests discontinuity in the level of model semantics, which, in view of semantic relativeness, is inaccurate.

3.3.1 Major data models

Although there are numerous data models proposed in the literature, database implementation is mainly limited to the three record oriented data models, namely hierarchic, network, and relational models. Such limitation is due to the need to utilize the efficient storage mechanism offered by the three major models.

The fact that DBMSs are limited to implementing such models imposes a constraint on other data models in that they have to incorporate rules for schema translation into an appropriate implementation model. This constraint weakens most models because they are compelled to compromise some of their semantic properties in order to comply with imposed translation procedures.

Of the three major models, the hierarchical and network are graph based models, where nodes represent records and arcs relationships between records. The relational model is table based, where a table as a whole is a relation in which rows represent records and columns represent attributes.

Hierarchical data model

The hierarchical data model is one of the earliest data models in use. It came about as a consequence to hierarchical storage structures of earlier databases. According to Clemons (1985), neither an official body, nor a specific person or paper put forward formal common specifications for the hierarchical data model.

In this model, data types are described by record types arranged in a tree-like structure. Each data type constitutes a root, and nodes are consistent of data sub-types. A tree in turn, consists of link types which indicate the connection between record types.

Because of the tree structure of this model, only one-to-one and one-to-many relationships can be represented. Modelling many-to-many relationships requires additional constructs, i.e., data duplication and additional trees. This is considered a major drawback in hierarchical data models.

Network data model

The network data model structure was defined by the Data Base Task Group (DBTG) committee of the COnference on DAta SYstems Languages (CODASYL, 1971). In network models, data is represented using record types and set types. A record type defines a one-to-many relationship between an owner and member record types. In a set type, a record can participate as an owner or a member record type. Relationships between set types are portrayed by connections between record types.

A major drawback of network data models is their inability to represent some relations, e.g., recursive sets cannot be represented because of inability of the network model to relate entity instances of the same class.

Relational data model

Relational data models are the most widely implemented in today's DBMSs. The concept of a relation in a tabular form as a method for representing information, instead of nodes and arcs, characterizes this category of data models and discriminates it from other data models.

The relational model is based on relational theory, and was put forward by Codd (1970) to "protect users of formal data systems from the potentially disruptive change in data representation caused by growth in the data bank". Codd defined a relation as a set of n-tuples, each of which has its first element drawn from distinct data sets.

The original model lacked semantic representation and did not facilitate the expression of constraints among relations (i.e., functional, join, and multi-level dependencies). The model has undergone a number of changes since it was introduced, and its development is still an active area of research.

An important progression in relational models was presented by Codd (1979) in an extension called RM/T. In this extension, the concepts of entity, property, aggregation, generalization, and association were introduced. *Entities* are designated to enable the relational model to define permanent keys, or system assigned surrogates. This resolved the problem arising from the manipulation of relations with user-assigned keys. Entities and entity types are classified according to their roles. An entity is called characteristic when it plays the role of describing other entities; associative when it interrelates entities of other types; and kernel when it is neither. *Properties* were introduced to support the concept of an atomic object. This is in addition to concepts of *aggregation* and *generalization* presented previously by Smith & Smith (1977a).

The universal relation proposed by Ullman (1984) is another extension of the relational model. The concept of a universal relation regards the database as a single relation with only one relationship between any two entities. In the case of more than one relationship existing between two entities, a new entity type is introduced to satisfy the definition of the universal relation. Database design based on the universal relation implies the definition of one initial relation which involves all attributes relevant to the database, thereafter smaller relations projected from the universal relation are defined until a satisfactory database design is achieved. This may prove to be impractical when dealing with very large databases.

3.3.2 Semantic data modelling

As previously stated, the record structure of major models provides a simple and systematic implementation mechanism. However, and as consequence of the record structure, data semantics are weakened. Kent (1979), among others, pointed out the limitation of record-based models in that they have very limited representational constructs rendering them incapable of explicitly expressing concept relationships. For example, a problem arises in record based models when trying to model an entity with a number of categories, because a field in a record is restricted to one kind of entry.

In order to achieve data models that are both easy to implement and express application semantics, another class of models have been developed and is referred to as semantic data models. Abiteboul & Hull (1984), pointed to the following important precepts of semantic data models. First, a semantic model "represents data about objects and relationships between them in a direct manner [which] allows database designers and users to think in terms of objects that information is stored about directly." Second, semantic models allow the formation of object classes and subclasses with common attributes. Finally, they permit the construction of object types out of other object types.

Many authors have placed considerable significance on the semantic capability of a data model. Improved semantic representation is the key to bringing down the linguistic barriers which have always been obstacles to effective organization communication. People at various levels of the organization have different views of similar data and information contents. This has been pointed out by a number of authors. Lewis (1980), for example, recounts a common dilemma which usually arises as a result of mis-communications between managers and employees at different managerial levels.

"Managers often forget that for an employee to understand the words (instructions, directions, orders) given, the employee must know the manager's purpose of using them. Receivers must interpret the sender's words from the speaker's perspective and their own. The situation is complicated because every business organization, as well as every department within those structures, develops a jargon all its own. Yet the organization must depend on each division understanding the other. When one unit forgets or ignores the fact that there are required and vocational variations in the meaning of words, roadblocks are very quickly established."

The data models reviewed in this section are usually used to complement the three data models discussed earlier. Although a number of models are considered in this section, an extended review of the entity-relationship model will be presented because of its significance in data modelling.

3.3.2.1 The entity-relationship model

The ERM and its diagrammatic representation were introduced as tools for database modelling by Chen (1976), in an attempt to achieve high level data independence, and incorporate as much semantics about real world as possible. This model - as implied by its name - represents real world through entities and relationships, which makes it easy to formulate and understand.

An entity is defined as "a "thing" that can be distinctly identified", e.g., "Person", "Employee", etc. A weak entity is an entity whose existence depends on the existence of another entity. For example, the existence of the entity "Dependent" is dependent upon the existence of the entity "Person". A relationship is "an association among entities", e.g., "Lives-at", "Works-for", etc. Entities and relationships are expressed in terms of their attributes which are drawn from value sets. For example, "Person" entity has the attributes "Name" drawn from value set "Names", "Address" drawn from "Addresses", and so on. An entity is assigned a key by designating an attribute (or group of attributes) with a value that is unique for that entity within the set of entities it belongs to. In a diagrammatic representation, a rectangle symbolises an entity, a double rectangle symbolises a weak entity, and a diamond shape for a relationship. Figure 3.7 illustrates an ERM diagrammatic representation of a manufacturing environment.

In a subsequent publication, Chen (1977) suggested the use of the ERM as basis for representing the enterprise view of data. The ERM, in this context, is utilized for deriving

Figure 3.7 An ERM representation of a manufacturing database (Chen, 1976)

an enterprise conceptual schema definition, which excludes conceptual to internal schema mapping considerations. Such conceptual schema definition is referred to as the enterprise view, and it serves as a tool for modelling a stable and storage-independent database.

Ng (1981) presented a formalized and enriched ERM to include a definition of an entity-relationship relation; a tabular view of these relations; a method for deriving 3NF; and a method for physical ERM representation.

A relation, in this context, is the user's view of the database formulated in an entity-relationship relation. Such a relation is defined as either deterministic or non-deterministic. It is deterministic if an attribute of a relationship or an entity has a one-to-one mapping with a single value in the value set. For instance, an "Employees" entity has the attribute "Age" which has the single value "45". The relation is non-deterministic if the attribute can have many values, e.g., the attribute "Telephone-no" of the entity "Employee" can have a number of numbers. A tabular form is suggested to illustrate entity and relationship relations.

A formal definition of the following four relation types is suggested: *regular entity relation*, in which the key attribute is not dependent on attributes of other entities; *weak entity relation* where the key attribute is dependent on attributes of other entities; *regular relationship relation*, in which no weak entities participate; and a *weak relation* in which one or more weak entity is participant.

Schiffner & Scheuermann (1979) presented an extension in which the ERM is improved to permit coexistence of multiple views by means of abstractions similar to those introduced by Smith & Smith (1977a). Two kinds of entity abstractions are suggested. The first is an entity abstraction by applying a predicate to an entity, the example given is the subsets "Old-Emp" and "Beginner" of the entity "Employee" (figure 3.8a). The second is entity abstraction by decomposition, as in decomposing the entity "Employee" into entity classes "Secretary", "Engineer", and "Trucker" (figure 3.8b). Relationship abstraction is similar to that of entities. The example provided for relationship aggregation is depicted in figures 3.9a and b.

Aston University

Illustration has been removed for copyright restrictions

Figure 3.9 Relationship abstractions (Schiffner & Scheuermann, 1979):
(a) Predicate relationship abstraction; and (b) Relationship decomposition abstraction

Additional concepts were introduced by Scheuermann *et al.* (1980) to improve the ERM extension presented above. In this enhanced form of the ERM, relationships between relationships are introduced based on the fact that relationships between entities can be abstracted into new objects which are related to other entities; as in the example of the relationship "Utilize" between the entity "Machine" and the aggregate object "Employee" illustrated above (i.e., figure 3.9a). A relationship is then said to be either total or partial. It is total on some entity set if each entity in the entity set participates in the relationship; if it is otherwise, then it is called partial. Two kinds of weak relationships are presented.

The first is similar to the weak relationship concept in the ERM; and the second is when an entity instance is dependent upon the existence of a subset or subclass of another entity set.

A data type approach to the ERM was introduced by dos Santos *et al.* (1980). In this approach, additional semantic abstractions are formed based on existing entity and relationship sets. Three constructs are suggested to form such abstractions: sum, product, and correspondence. A *sum abstraction* formed over entity/relationship set types t1, t2, ..., tn is the set t of type ti where 1≤ i ≤n. The abstraction "Employee" can be formed over the entity sets "Secretary", "Engineer", and "Trucker".

A *product abstraction* over set types t1, t2, ..., tn is the set t where an object of type t is formed of n components derived from objects of types t1, t2, ..., tn. For example, the product abstraction "Class" derived from the sets "Teacher", "Course", and "Room".

Last, a *correspondence abstraction* over t1, t2, ..., tn is the set t where objects of set t are derived according to some indexing rule over a set type ti and is referred to as a coset. A correspondence set t may be indexed according to a certain attribute value or based on a related entity. An example of the first, is the indexed set, or cosets of persons of the same age over the entity "Person"; and of the second is the cosets of vehicles driven by a each trucker which involves the entity "Trucker" related to entity "Vehicle" by the relationship "Drive". Figures 3.10a, b, c and d illustrate constructs of sum abstraction, product abstraction, and correspondence indexed by attribute and entity, consecutively.

The Normal Entity-Relationship Model (NERM) (Wang & Shixuan, 1986) is another ERM extension in which abstractions of entities are considered. An entity E in NERM corresponds to a relational schema R(U,F), where U is the set of attributes of E, and F is the set of associations among the attributes of E. All relational schemas are in third normal form, and therefore, entities are said to be normal. A normal entity set is either associative or basic. An associative normal entity set is one in which associations among two or more (possibly associative) entity sets are represented.

Another definition introduced for an entity set which is called *original entity* is an abstraction of a number of entity subtypes. The entity set "Employee", for instance, is an original entity set for the subtypes "Manager" and "Worker", and is diagrammatically represented by double lines on either sides of the entity set rectangles. Subtypes, in this situation, may not have subtypes of their own, but are further classified according to their relationship within the original entity set.

Figure 3.10 The data type approach (dos Santos *et al.*, 1980):

(a) Sum abstraction; (b) Product abstraction

(c) Correspondence indexed by attribute; and (d) Correspondence indexed by entity

NERM diagrammatic techniques, based on "improved ER diagrams using the relational data theory", are presented for the design of the enterprise schema. The schema, is in turn developed through analysing data, designing user oriented ER diagrams, formalization of semantics from ER diagrams, elimination of redundancies, and finally building an NERM diagram of the enterprise schema. Figure 3.11 shows an NERM equivalent of an ER diagram. Note that the two subclasses of the entity "Teacher", i.e., "Lecturer" and "Manager", are created to prevent occurrence of more than one relationship among two entity sets. i.e., "Teacher" and "Student".

The role concept introduced by Bachman (1977) was utilized by Hawryszkiewycz (1984) for extending the ERM to strengthen its management of relationships. Roles are different from types in that types are properties of entity sets, while roles are characterisations of one or more entity set in a database situation.

Figure 3.11 An NERM diagram equivalent of an ERM diagram (Wang & Shixuan, 1986)

A recursive relationship on an entity set is transformed into a relationship between two entity subsets referred to as roles. As such, the entity set is treated as an abstraction of role subsets. The example in figure 3.12 illustrates the recursive relationship "Treatment" over the entity set "Person" which is represented using the roles "Patient" and "Doctor". Roles can also be defined over a number of entities' subsets playing similar roles in a relationship. This definition of roles is useful in grouping sub-entities of more than one entity set under a common representational entity. This is illustrated by the relationship "Borrow" between the entities "Person" and some borrowed entities from different entity sets, e.g. "Record", "Book", "Equipment", etc.

Figure 3.12 The roles "Doctor" and "Patient" for the entity "Person" (Hawryszkiewycz, 1984)

The Entity-Category-Relationship data model (ECR) (Elmasri *et al.*, 1985) is an extension of the ERM in which entity types are grouped in categories according to the roles they play in a relationship. The ECR model provides formal definitions for entity generalization abstractions. Such abstractions are formed by specifying a category in which entity types participating in a relationship are specified. For example, the entity types "Automobile" and "Truck" participating in the same relationship "Ownership" are grouped under the category "Vehicle". On the other hand, subsets of entity types could be represented as categories. For for example, the entity type "Employee" could be represented by the categories "Full-time-employment", "Scientist", and "Technician". Categories are not necessarily disjoint. In other words, it is possible for categories to overlap.

The semantic entity-relationship model (SERM) (Lenzerini, 1985) is designed to integrate the entity-relationship approach with abstraction mechanisms of semantic data models. SERM deals with the modelling process at three stages: object space, constraint language, and manipulation and predicate language.

The objects space is similar to the ERM enterprise model, and it supports aggregation, generalization, and classification abstractions. However, and in order to avoid restrictions on classification levels and strict data type rules, the SERM suggests two parts in the objects space, intensional and extensional. The extensional part addresses three categories of classes: entities, relationships, and domains. Domains refer to entity and relationship groupings according to common attributes. The intensional part addresses data which are not classes of the objects space. Such partition of the objects space serves to establish a distinction between permanent characteristics of data abstractions, i.e., classes, aggregation, and generalization on the one hand, and time varying objects of reality on the other.

In SA-ER (Carswell & Navathe, 1987), methods of structured analysis and entity-relationship modelling are combined to provide a technique for requirements definition and database design. A three step methodology is suggested to transform organization data into a synthesized global schema.

In the first step, structure analysis techniques are implemented to specify the processing requirements. Processes are brokendown to sub-processes which are in turn expanded into further sub-processes, until simple data flow diagrams can be drawn of a sub-process. As a result of the expansion process, a hierarchy of data flow diagrams is realized which reflects a component breakdown of organizational functional processing. A description of the data-flow in each data flow diagram is then recorded, and data elements referenced in process expansion and data-flow description are defined in terms of name, value, and meaning related to the data items. Note that there is no rule to indicate the limit to a sub-process expansion; instead, it is left up to the designer to make such decision. The main output of this step is a specified description of the processes in structured English.

The second step is view modelling. In this step each data-flow representing a sub-process is translated into an ER diagram, and is called a view-segment of that process. View-segments are then merged to form the process view. Categories of entities, relationships, and attributes are formed according to process specifications. Finally, process views are integrated to form the global conceptual schema.

The Enhanced Entity-Relationship Model (EERM) (Elmasri & Navathe, 1989), presents a reflection of most recent advancements and extensions of the ERM. According to the EERM, an entity type with entity sub-type grouping is called a super-class, and sub-groups are called sub-classes. The entity "Employee," for example, is a super-class, while its sub-types "Engineer", "Manager" are referred to as sub-classes. While super-classes and sub-classes are treated at the same level in reality, a distinction is drawn between the two abstractions at the implementation level. Sub-classes inherit all attributes of the super-class entity.

Sub-classes of an entity type may also be grouped according to specific entity characteristics or specialization. The entity "Employee" may have the two groups "Part-time-emp", "Full-time-emp" as its sub-classes. More than one specialization can be defined over an entity set, in which case, a sub-class is a participant in more than one super-class.

Generalization, according to the EERM, is in a way revising the specialization process. It is the process of grouping entities with similar characteristics into a generalized super-class entity. Therefore, sub-classes of a generalized super-class can be treated as specialization of the super-class. In the EERM diagram, an arrow is drawn from the subclass to the super-class and visa a versa to denote a generalization and specialization respectively. Having said that, it is not clear as to whether a specialization or generalization abstraction is best for a certain situation, rather it is left up to the designer to decide based on what is seen best for that situation.

Specialization and generalization can be partial or total over some super-class. It is total when every instance of the super-class must participate in at least one subclass. For example, the specializations (or generalization) "Car" and "Truck" over the super-class "Vehicle" is said to be total if every instance of the super class is either a car or a truck. The abstraction is said to be partial if it is otherwise.

Another property of subclasses of specialization and generalization is that an instance of the super-class can participate in more than one subclass. In this case the subclasses are said to overlap. For example, "Engineer" and "Manager" subclasses of the super class "Employee" are overlapping subclasses because an engineer can be a manager at the same time, and therefore participates in both subclasses. When overlapping is not permitted the subclasses are said to be disjoint.

It is important to mention at this stage, that ER modelling literature includes a far greater number of tools and modelling techniques than presented in this section. The majority of work on the ERM and its extension can be found in ERM conference proceedings (Chen, 1980, Chen, 1981, Chen, 1983, Davis *et al.*, 1983, Lui, 1985, Spaccapietra, 1986, March, 1987, Batini, 1988). However, the above mentioned publications where chosen because of their immediate relevance to the issues discussed in this thesis.

3.3.2.2 Other data models

In the remainder of this section, other prominent data models will be briefly outlined with emphasis on their simplicity and semantics.

Data abstraction methods introduced by Smith & Smith (1977a, 1977b, 1978) are central to most aggregation and generalization abstractions presented in data modelling literature. According to Smith & Smith, abstractions provide effective mechanisms for the "suppression of all details about some object (or activity) except for those related to the understanding of some phenomena of interest".

An aggregation is a structure composed of related objects. For example, the relationship between the objects "Person", "Hotel", "Room", and "Date", is represented by the aggregate structure "Reservation". A generalization, on the other hand, is the classification of objects with some common attributes. For example, the entities "Dog", "Horse", "Cat" are generalized as "Animal". An abstraction may have other abstractions within its structure, resulting in what is referred to as aggregation and generalization hierarchies. Abstractions are not necessarily disjoint, and are called blocks when they are. The methods presented in this work have implications beyond data representation, and the most important of which are the consequences such abstractions have on implementation and storage efficiency.

The Semantic Database Model (SDM) (Hammer & Mcleod, 1978, 1981) is based on class abstractions. A class is "a meaningful collection of entities ...[which] are not in general independent, but rather are logically related by means of interclass connections". Interclass connections are responsible for delivering a database structure that is capable of integrating multiple views over the same information. Beside name and value, attributes in SDM are specified in terms of their belonging to a class or a member of a class. Attributes can be related to other attributes in two ways: *inversion*, where an attribute has the inverse value of another attribute; and *matching*, when the value of an attribute is matched with the value of another attribute. The values of an attribute can be calculated from other information in the database by specifying an associated derivation primitives which are defined on classes, subclasses, and members of classes.

Functional data models are based on the concept of viewing information in the form of functions which was first introduced by Follnus *et al.* (1974). As an example, the functional database model proposed by Kerschberg (1975) is a graph whose nodes are entity-relationship sets, and arcs are functions. Aggregation of entity sets is realized by defining additional functions on the relevant subsets of entity sets. The main drawback of functional data models, according to Abiteboul & Hull (1984), is their inability to "gracefully represent new object types which are built from existing object types...[and] lack of mathematically rigourous formulation of the model and its semantics."

The structural data model presented by Elmasri (1980) is a formalization and an extension of a data model presented by Wiederhold (1977), which in turn is an extension of the relational data model. This model adds the concept of connection between relations of the relational model. Connections are used to represent relationship properties, and classification concepts. Relations are formally defined, and because this model emphasizes the importance of implementation specifications, relations are classified into five types according to their association with other relations in the database. The *primary entity relation* defines a set of tuples closely corresponding to a class of entities. *Nested relations* define a case of repeating groups over an entity set. *Referenced entity relations* define a relation which can only be defined through other existing entity relations. In *lexicons*, repeating attributes that are functionally dependent are separated and grouped. Finally, *association relations* contain data relevant to the interaction or association of entity relations. The main advantage of the structural model is its ability to include structural information about relations in the model.

3.3.3 Organization-oriented data models

Although numerous data models have been proposed in database modelling literature, very few models offer mechanisms for incorporating behavioural and structural organization issues. In this section a review of organization oriented models is presented to evaluate the effectiveness of such methods in accounting for issues intrinsic to information processing in general, and organization structure and its impact on the functioning of the whole of the system in particular.

Sen (1982) presented a data model which is described as management-oriented. The model suggests a hierarchical relationship between user views at the external schema. The hierarchy of external views reflects the three levels of managerial activities, i.e. strategic, control, and operational levels.

At the requirements definition stage, organizational functional activities are identified. For each activity, the views necessary for performing an activity are defined. A defined activity is said to be performed at one or more managerial level, and therefore, views belonging to that activity are related to the appropriate management level, thus creating a hierarchy of views relating to the activity. Views of different activities are then connected using modelling functions that are designed to make appropriate vertical and horizontal connections between the various view, and therefore creating a hierarchical structure of external hierarchical views illustrated in figure 3.13.

Aston University

Illustration has been removed for copyright restrictions

Figure 3.13 The structure of the external schema (Sen, 1982)

The weakness of this approach, however, is in that it deals with user views at an external level; i.e., the hierarchy of user views does not exist at the conceptual level. As such, the approach is only useful for managing database updates, and as such is of benefit at the implementation phase of the database design.

Feldman & Miller (1986) presented a method of constructing a model of entity abstraction clusters. The model utilizes a diagrammatic technique to split an entity into sub-entity abstractions, thus creating a hierarchy of clustered entity abstractions.

> "A clustered entity model is a hierarchy of successively more detailed entity relationship diagram, with a lower-level diagram appearing as a single entity type on the next higher level diagram (cf. structured data flow diagrams) where functions are decomposed into more detailed self-contained diagrams." (Feldman & Miller, 1988)

Three diagrammatic levels are suggested: high-level, subject areas, and information areas. At the lowest level, an information area diagram depicts entity types and interrelationships among entity types. At the next level up, a subject area diagram shows major entity types (i.e., entities duplicating in more than one diagram), information areas, and their interrelationships. At the high-level diagram, the preceding levels are composed to form a diagram of the organization model.

The clustered entity model combines abstraction mechanisms with main principles of structured analysis. The detailing of processes in this case is limited to three levels, and therefore, an analyst is required to make the decision as to the level in which to assign a process or a sub-process. While this approach creates a hierarchy of entity abstractions based on entity participation in a process, the previous approach (Sen, 1982) presents a hierarchy of user views based on the level of management activity in which the user is located. The main problem with this approach, however, is that the hierarchy of clusters is based on existing organization activities. This could prove ineffective in that the designed database is susceptible to the constant changes in organization activities.

3.4 Conclusion

The literature review presented in this chapter has reflected the diverse opinions on database design in general, and data modelling in specific.

It is evident from the review of database design methodologies that an integral approach to resolving design problems is lacking. The diversity in the suggested design approaches has lead to concentrating research in the direction of dealing with specific design issues instead of following a more holistic approach to the design process. For example, the issues of linguistics, organization structure, management, and communication - which are all of relevance to the database - rarely surface in database design literature. As a result, the opportunity of finding more intrinsic solutions to design problems is lessened.

It is also evident from the review of prominent modelling techniques, that semantic inclusion has not been adequately considered. This is manifested in the absence of proper justification for describing a model as semantic. The term semantic has not been used to the extent of its real meaning. Accordingly, data models reference to data semantics requires a full understanding of semantic theory and its implication on data modelling. Such understanding would provide the necessary tools for evaluating the extent of a data model's semantic representation capability.

Although recent database modelling approaches emphasize the importance of identifying intrinsic characteristics of an organization and its environment, there are still issues of critical importance and immediate relevance to the organization and its structure that are overlooked in most established models. The main reason for the inadequacy in including organization issues in data modelling is the inability of modelling specialists to directly associate issues of organization structure and information processing with those of modelling and databases in general.

Chapter 4

Organization foundations: The stratification of structure hierarchies

It was clear from the previous chapter, that organization aspects were not fully acknowledged as being of significance in database design. In the few models which recognized the importance of such issues, limited constructs were provided to take organizational aspects into considerations.

In the proposed approach, the structure of the organization is recognised as being of substantial importance in database design. Since the organization hierarchy was previously identified as the factor of most influence on information processing, and because the stratification of the hierarchy is associated with the execution of the proposed approach, detailed analysis of the hierarchy fundamentals is necessary.

This chapter will examine the concepts underlying the existence of the organization hierarchy, in addition to the consequences of hierarchic structure on data utilization. Moreover, notions underlying hierarchy stratification are examined. Such examination will provide guide-lines to identifying conceptual abstraction levels which will then be utilized as guide-lines for data abstraction levels incorporated in the proposed approach.

4.1 Structure hierarchies and conceptual abstractions

In chapter 2, the relationship between the organization hierarchy and the data abstractions was established. The rise of conceptual abstractions was attributed to the nature of information required by activities at various organization levels.

In the proposed approach, and in accordance with effective information design techniques, information relevance is emphasized as an important factor in database design. Accordingly, new relations will be introduced to link the organization hierarchy with conceptual abstractions, in order to maintain the relevance of information as implied by the nature of information required throughout the organization hierarchy. In other words, conceptual abstractions will be established and maintained in accordance with the way such concepts are utilized in the organization.

It is necessary, however, and before discussing the organization oriented abstractions, to specify the issues of organization hierarchy which give rise to conceptual abstractions. It is also necessary to specify the nature of hierarchy stratification, and the number of levels within its stratification. Such aspects of structure hierarchies will serve as reference for constructing conceptual abstraction.

4.2 Established methods of stratification

In this section, an examination of conventional methods of hierarchy stratification is presented. Emphasis will be placed on the relationship between the hierarchy and the degree of stability in the levels created by such hierarchies. Such analysis will provide the criteria for evaluating stratification mechanisms provided by such hierarchies. Consequently, a stratification identification mechanism will be used as the base upon which the proposed approach can evaluate and make use of data abstractions arising as a result of organization hierarchies.

4.2.1 Functional hierarchies

The decomposition of organization functions into manageable sub-functional processes forms the most basic form of hierarchic structure. In all formal social systems, a model is devised of the functions and activities within the system. This model charts, in varying levels of detail, the functional breakdown of the organization, and the relationships between functions independent of authority channels. It is this independence which distinguishes the functional hierarchic structure from other forms of existing structure hierarchies (presented in subsequent sections).

Business functions divide an organization into vertical divisions or departments in accordance with appropriate subdividing mechanisms (previously presented in 2.3). Each department performs processes that are specific to its nature. Business processes are defined as "groups of logically related decisions and activities required to manage the resources of the business" (IBM, 1984). Activities, on the other hand, can be thought of as the minimum decomposition of functions. According to Martin (1982), an activity is something which becomes a computer or manual procedure, and usually relates no more than seven entities. When the number of entities related in any one activity is more than seven, the activity is then further decomposed. Figure 4.2 illustrates a decomposition of functional areas into activities.

The hierarchy of functional decomposition contributes a number of advantages. In relation to this work, functional hierarchies serve as pointers to the definition of information requirements in general, and entities involved in performing organizational activities in specific. Defining entities through activities does not

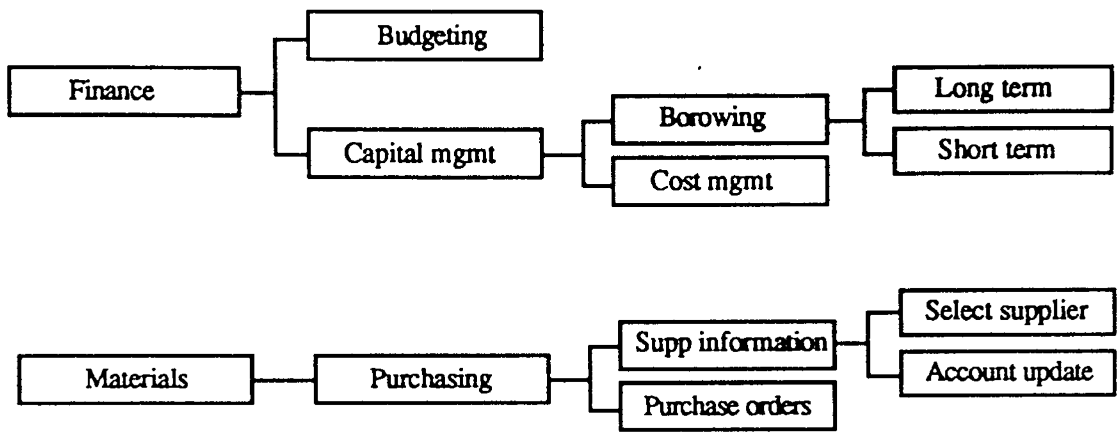


Figure 4.1 Decomposing functional activities

necessarily imply that an entity's existence is contingent upon the existence of a specific activity. Rather, in an organization, entities exist independent of activities, i.e., there will always be information on "Employee", "Customer", "Parts", etc., no matter how or when this information is used.

Functional hierarchies provide uniform representation of organization functions and activities that are independent of responsibility channels. However, the hierarchic representation of such functions usually overlooks functional interdependence as a result of its reductionist approach.

Thompson (1967) described how departments, as part of a system, are connected to one another according to one of three types of interdependence: *pooled*, *sequential*, and *reciprocal*. In *pooled* interdependence, parts contribute to the whole, and the whole supports individual parts. *Sequential* interdependence refers to a part being completely dependent on its functioning on another part in the system. Sequential interdependence is therefore, a special case of the pooled type. Finally, *reciprocal* interdependence depicts a situation in which the output of one part is the input to another. In this case, the interdependence is both sequential and pooled. Figure 4.2a, b, and c illustrates Thompson's description of interdependence types.

Figure 4.2 Functional interdependence (Thompson, 1967):
(a) Pooled; (b) Sequential; and (c) Reciprocal

There is a further domain of analysis concerning the management and optimization of organizations based on departmentation interdependence exemplified by the above illustrated approaches. Such analysis is beyond the purpose of this study. But there is a point to be made here, in that functional analysis is more complicated than suggested in information systems literature, and it is such complexity which renders functional hierarchies inadequate for reflecting the intricacies of functional interdependence and hierarchic stratification. In addition, functional hierarchies do not present a stable base, because they are subject to existing types of activities performed in the organization.

4.2.2 Authority and responsibility hierarchies

In chapter 2, authority was defined as the power vested in selected individuals, in accordance with their position in the structure, to implement control and co-ordination functions. It was also mentioned that authority plays the essential role of enforcing rules and institutionalizing behaviour. However, and having given a prevalent definition of authority according to organization and management literature, it is essential to provide a closer examination of authority relationships in view of the issue under consideration, i.e., being the essence of structure composition.

The concept of authority has always been considered the sole constituent of an organization hierarchy. As such, an illustration of organization structure is usually represented in the form of a description of the current line of authority. However, a closer scrutiny of authority will show that authority lines (or chain of command) are not effective for delivering such representation.

First, and foremost, the chain of command usually suffers frequent transformation in accordance with new plans and changing requirements.

Second, authority as formally defined in management literature does not take into considerations the informal aspects of structure. In reality, authority and responsibility channels can be overlooked and even overruled because of informal factors, such as personal factors. It is not unusual to detect additional communication channels catering for the needs of the informal organization which are not depicted in the formal line of authority.

Third, the concept of authority as defined above, implies a single hierarchy of authority within an organization. Nonetheless, the reality of organization structure suggests otherwise. An employee or department could be under the supervision and command of more than one source of authority, as in the case of matrix organizations. Such reality is a by-product of certain consequences emanating from the need for sharing resources and managing departmental interdependence.

Finally, reference to authority and responsibility chains suggests the existence of definite and uniform relationships among the levels of authority throughout the organization hierarchy. Gordon (1968) indicated that such suggestion is invalid. For example, authority exercised by an organization president over deputy-vice presidents is at a completely different level than that of a foreman authority over workers under the foreman's command. It is not possible to find a superior-subordinate authority relationship that is typical of authority throughout the structure.

A number of authors have investigated the types of authority relationships existing in social structures. Peabody (1957) presented a survey of authority literature with the objective of understanding the source of authority which conditions its acceptance. The survey produced the following classification of authority sources. First, *authority of legitimacy*, which is deep rooted in the definition of employer and employee. Second, *authority of position,* and it is one inherited in the position and not the person. This class of authority has the closest relation to structure hierarchies, and in this sense, it exemplifies the reference made to authority in general. Third, *authority of competence* exhibited by technical knowledge and experience. In this case authority is said to cut across formal hierarchic relationships and communication channels. finally, the *authority of person* emanating from some personal characteristic attributes.

In addition, Jaques (1978) and Rowbottom & Billis (1987) discussed other important aspects of authority, including the inadequacy of authority lines for representing an organization structure.

It is therefore clear from the above discussion, that organization stratification based on authority cannot provide a long term solid approach to identifying organization levels. Because the significance of the stability and decisiveness of any approach to be utilized for determining organization levels is of incontestable importance to the proposed modelling approach (SECOM relies on such levels for the representation of abstraction hierarchies). Both functional and authority hierarchies will not be utilized as bases for reflecting organization stratification. Instead, further examination of alternatives to the approaches presented in this section will be evaluated for the purpose at hand.

4.3 Alternatives to conventional hierarchies

Since reference is usually made, in information systems literature, to the relationship between data abstraction and hierarchies of managerial levels, a definitive mechanism is required - based on the fundamentals of such relationship - which would enable information specialists to define a robust method for identifying managerial hierarchies. Such mechanism needn't be concerned with the intricacies of structural relationships, but should be depended upon to provide valid grounds upon which general reference to data abstractions in relation to structure hierarchies could be made.

Alternative methods to defining basic levels of hierarchic structure date back to Urwick's renowned work on the theory of organizations (Urwick, 1943). Paterson (1972) and Beer (1979) addressed the issue of levels from the point of view of decision making and neurocybernetics, consecutively. However, their work is mainly concerned with the management side of organizations, which makes it susceptible to inheriting certain shortcomings posed by authority channels, e.g., the constant update of such channels.

Presented, in the following section, are two approaches to establishing foundations underlying structure hierarchies. The approaches offer convincing methods for specifying organization levels which could be relied upon to proffer credible and dependable basis for conceptual abstractions. The argument in the first approach is psychologically based, while the second suggests the nature of work as the essence of structure. It is important to note that since the objective of this chapter is to arrive at a method for distinguishing hierarchic levels, the following approaches are briefly discussed to provide a general illustration of suitable methods of level identification. Detailed descriptions of the following approaches can be found in the references provided.

4.3.1 Levels of perceptual abstractions

The importance of understanding human perception to this work, is in that it presents valid basis for reflecting intrinsic characteristics of structure hierarchies. Such basis are less vulnerable to recurrent alterations which authority and functional hierarchies are susceptible to. The fields of perception and cognition suggest that there is a relationship between human perception and the ability to perform complicated tasks. In turn, this relationship can be reflected in social organizations in the form of structure hierarchies. Such hierarchies are therefore referred to as perceptual abstractions. This section will examine the levels of perceptual abstractions as indicators of structure levels, and thus the use of such levels as pointers to the degree of change in the abstraction of user information at each level of the organization structure.

Psychologists have been trying for a number of years to establish the nature of the distribution of human cognitive capabilities. However, it is still not yet clear whether human ability to form abstractions is a linear distribution over a population, or that the distribution is discontinuous because attributes of measurement are multi-valued. An early attempt to introduce the theory of distinct levels of thinking was presented by Goldstein & Sheerer (1939), in which concrete and abstract states were suggested as distinct levels of cognition.

While the common belief is that forming abstractions is a linear and progressive process, there seems to be an opposing view presenting a valid argument which relates that an examination of human ability to form such abstractions will reveal stages at which forming abstractions and categories is transformed. These stages, or levels, are completely different in nature, and any single level cannot be thought of as systematic progression of another level. One of the main proponents of this theory is Jaques, whose work has also been directed towards the examination of structure hierarchies.

Jaques (1965, 1976, 1978), presented the theory of levels of abstraction as an explanation to the existence of the hierarchic structure of bureaucracies. The proposition stated that the stratification in structure represented by levels forming the hierarchic shape, as it has always been recognized, can only be explained by the theory of the distribution discontinuity of human cognitive competence.

Jaques theory of abstraction levels is backed by extensive research into social scientific issues in general, and management and cognitive psychology in specific. It is beyond the scope of this work to discuss such issues, suffice it for the purpose of the current analysis, to put forth the suggested levels of abstraction as a bona fide alternative to conventional approaches. The levels are presented in the following paraphrase of the Jaques' description.

Level-1: Perceptual-concrete

Emphasis at this level are on concreteness and immediate perceptual contact with output of the activity at hand. Work is described, and in a sense conceptually constructed beforehand. Rules and procedures are either provided or presumed, and in ether case they are strictly followed. Work is usually dependent upon a skill required for operating upon objects.

Level-2: Imaginal Concrete

Unlike the first level, goals to be accomplished at this level can no longer be completely specified. Thus ambiguity, which is usually handled by flexible rules, may enter the situation by not presenting a completely specified goals and means. However the output is still cognizant in the sense that it can be imagined in concrete terms.

Level-3: Imaginal Scanning

The tasks performed at this level become impossible to physically oversee all at once, and the scope of activities becomes too wide. The instructions received at this level tend to be in conceptual terms and are translated into imaginal picture of the tasks controlled. As a result of the loss of concreteness at this level, performance evaluation becomes less defined and rather elusive.

Level-4: Conceptual Modelling

The profound change from concrete to abstract mode of thought at work are characteristics of this level. The quality of output is therefore highly dependent on the individual's capacity to think in abstract terms. Tasks are usually based on existing description which serves as an example, but the output is a departure from such description, and thus, yields an original product. This level marks the emergence of true abstraction and innovation with more degree of uncertainty than of that at level-3.

Level-5: Intuitive Theory

The emphasis at this level changes to intuitive relationships with the universal. Individuals at this level have limited contact with the concrete reality of the total field of responsibility. In addition, tasks performed are usually concerned with long-term plans which are then passed to other levels for their implementation. Uncertainty dominates all activities because of the inductive intuitive relationship with the universal. This level completes the total system, because it exhausts the universe of discourse within which all levels may be applied.

In addition, it was suggested in the above approach, that abstraction levels can be detected by the presence of certain characteristics. Figure 4.2 presents a comprehensive characterization and examples of each level.

Before drawing any conclusion as to the effectiveness of perceptual abstraction levels in reflecting structure levels, a second method is examined in the following section.

4.3.2 The Work-levels approach

Rowbottom & Billis (1977, 1987, 1989) identified five levels in a structure hierarchy through which all organization activities are carried out. They have also identified two additional levels, (sixth and seventh), which are usually found in major corporations such as departments of state, and large local authorities. The approach provides sound foundations for identifying stable organization levels.

The stratification is one in which "the range of objectives to be achieved, on the one hand, and the range of environmental circumstances to be taken into account, on the other, broaden and change in quality at successive steps" (Rowbottom & Billis, 1989). Briefly, the work-level approach includes the classifications presented in figure 4.3.

The two alternative methods of stratification identification, i.e., perceptual abstractions and work-level approach, are based on issues more intrinsic to the organization than its functions or line of authority. The bases upon which such alternatives are grounded are, as indicated by such approaches, reflective of time-independent work-hierarchies. Both approaches have psychological bases for their classification. This is an indication of the stability of such methods, simply because such measures are, while being the determinate of the levels themselves, are

Aston University

Content has been removed for copyright reasons

Figure 4.3 Characteristics and examples of perceptual abstraction levels (Jaques, 1976)

Figure 4.4 The work-level approach to hierarchic stratification (Rowbottom & Billis, 1989)

permanent, i.e., human level of thought is always a factor in organizations. This is in converse to methods which are under constant modification and update.

The two approaches present valid foundations for defining the kernel of structure hierarchies, and present reliable guide-lines for the identification of levels in a structure hierarchy. It is essential, as was stressed earlier, to embark upon a secure understanding of the reality of structure and its predominant characteristics, especially its hierarchic shape. From such foundations, the database design at the modelling stage, can safely proceed to represent the reality of an organization in general, and the variety of information representation related to the levels in a hierarchic structure.

Because of their homogeneous composition, either of the later two methods of level identification will be utilized for the purpose of the proposed conceptual modelling approach.

4.4 Conclusion

The purpose of examining methods of organization hierarchy was to arrive at a stable basis upon which the identification of hierarchy stratification can be ascertained. Such identification, in turn, was said to be necessary for locating the areas in which information representation varies in quantity and quality.

The significance of identifying hierarchic levels will emerge more clearly at later stages. It is important to indicate, however, that it is not the objective of this work to identify hierarchic stratification. Rather, the presentation was given to illustrate the methods which could be used to identify such levels. In addition, it is important to realise that only methods which can provide links between structure levels and levels of data abstraction can be qualified for incorporation in a modelling approach which reflects such relationship. Since the relationship between organization and conceptual modelling is determined through the recognition of the reasons underlying abstraction formation, conceptual modelling would be best served by incorporating a method which utilises abstractions for identifying structure levels.

Chapter 5

Semantic foundations: Semantics and conceptual modelling

The subject of data semantics has been an important issue in data modelling since Abrial's acclaimed paper on data semantics (Abrial, 1974). To refer to a model as semantic is to point out a favourable property of that model. A semantic model reflects intrinsic properties of data, the foremost of which is meaning. This has resulted in research directed towards improving the semantics of data models. However, and because of the relative degree of semantic capability in a certain context, describing a model as semantic has become, to a considerable extent, a matter of opinion.

It is clear that such a state has resulted from the lack of proper comprehension, on the part of researchers, of the various implications of semantics on linguistics, logic, and philosophy in general. Such implications have a unique relevance to data modelling which has never been adequately addressed in the context of modelling for database design. Only when such uniqueness is specifically and explicitly delineated could researchers ascertain both the strength and the extent of a model's semantic capability.

This chapter will clarify the issue of semantic relevance to conceptual modelling. The significance of semantic implication on modelling was outlined in chapter 3. In order to gain an appropriate recognition of such implication, a comprehensive evaluation of the consequences of semantic application to conceptual modelling is required, and thus is provided in this chapter.

In addition, this chapter will introduce the basic semantic concepts and definitions upon which the new approach is based. In this sense, this chapter will serve as a reference for the semantic aspect of SECOM and the terminology used in the following chapters.

5.1 Semantic foundations

The term *semantics* has been defined in the modelling context as having to do with the meaning of words. In other contexts, it is defined according to the Greek origin's definition of the word which is composed of the prefix "seme" meaning *to signify*, and the suffix "ics" which suggests organized knowledge (Rapoport, 1965; Lyons, 1977). However, none of the above definitions provide constructive provisions to understanding the meaning and implications of *semantic*. Furthermore, the above definitions can easily lead to mistaking semantics for lexicography (Katz, 1972).

Instead of trying to establish a definition of semantics, a brief overview of the theory and application of semantics in various scientific contexts will be outlined to convey the meaning of semantics and its significance in information and data modelling.

Semantics can be thought of as the science directly concerned with everything and anything that can be classed as a sign or symbol (George,1964). In reality, a countless number of signs and symbols exist, and while some are linguistic in nature, others are non-linguistic and require other forms of understanding.

Whenever a sign or symbol is used to carry information, it is always a subject of examination by the observer. People are preoccupied with the notion of analyzing the meaning of signs and the validity of the information carried within them, be the sign a road map, a simple statement, or a mathematical formula. Such analysis is performed at different levels depending on the observer, the sign, and the context in which it appears. But, in some cases the scrutiny is more critical than in others, as in the case of confirming the constituents of a certain theory.

Because only linguistic terms are of concern to this thesis, reference to symbols, constructs, concepts, and expressions is meant to stand for linguistic terms.

Both the meaning and the truth of a linguistic construct are of critical importance to a number of scientific disciplines. Consequently, the field of semantics is an issue under examination by many branches of science. Bunge (1974a), classified semantics into empirical and non-empirical. It is empirical when semantics "seeks to answer problems concerning certain linguistic facts - such as disclosing the interpretation code inherent in the language - or explaining the speakers' ability or inability to utter and understand new sentences of the language." On the other hand, semantics is non-empirical when it is concerned with conceptual objects, such as mathematical structures. In this case, the concern is not bound to the linguistic construct alone, but also with its reference. Figure 5.1 illustrates Bung's classification of semantics.

Figure 5.1 Semantic classification (Bunge, 1974a)

Earlier, it was mentioned that semantics have a unique bearing on data modelling. The above classification justifies such assertion. Data modelling involves - among other things - aspects of language, perception, observation, and mathematical theory. To project a comprehensive portrayal of semantic implications on conceptual modelling, an illustration of interesting aspects of semantics from the various perspectives involved will be outlined first.

5.1.1 Perspectives on semantics

At the centre of the semantic movement is the understanding of language in general. Understanding the meaning of linguistic constructs is fundamental to effective communications in general, and is critical to proving and communicating scientific theories in particular. Understanding the meaning of a word requires both its definition and context. In addition, the part of reality a word refers to is critical to understanding its meaning.

Because words are human invention, they can be thought of as standing for ideas in the mind of the speaker (and/or writer). Therefore, and in order to come to a full understanding of a word's meaning and truth, its significance in the mind of the speaker has to be ascertained.

Ogden & Richards (1938) illustrated the relationship between a word as a symbol, the idea it signifies, and the fact it refers to in the - since then - well known significance triangle (figure 5.2). A word provokes a certain thought, which in turn refers to a fact. Therefore, to communicate an idea, the speaker uses the symbols which will ensure the formation of the desired thought, which will in turn refer to the same factual context as that of the speakers reference. The authors suggested an *imputed* relationship of *stands for* between the symbol and the referent. However, this relationship will be shown to have certain significance and will be discussed in more detail at a later stage.

Figure 5.2 The significance triangle (Ogden & Richards, 1938)

The primary purpose of language however, is not just to symbolize reality, but rather to communicate and promote notion and purpose. Korzybski (1933) put forward a model of language usage in communication within the context of General Semantics which he called *Structural differential* (figure 5.3). In this model, aspects of reality are observed and represented at different abstraction levels. In figure 5.3, the parabolic shape at the top represents the *empirical world*, and the circle stands for the observer. The rectangles illustrate the levels of linguistic abstraction used by the observer to describe reality; the lines connecting to the rectangles refer to characteristics included in an abstraction from a previous one; while unconnected lines refer to omitted properties.

The lowest level of abstraction (illustrated by the dots in the observer circle) include facts that can be observed by all organisms, which - though communicated using words - could not be explained any further. In other words, the observation is part of

Figure 5.3 The "Structural differential" approach (Korzybski, 1933)

behaviour and its symbolic description does not have further reference to any linguistic expression. For example, the colour *red* cannot be explained in linguistic terms, and only seeing the property of *redness* would communicate the factual existence of the colour *red.* This level is basic to all organisms, and is not part of language. It is, therefore, depicted as a prerequisite to communication. Each progressive level, thereafter, represents a further abstraction of reality based on a filtration of the abstraction at the previous level.

The meaning and interpretation of linguistic expressions are also elementary in philosophy. It has even been indicated that there is no dividing line between semantics and philosophy. Wittgenstein (1953) suggested that philosophy is a critique of language. In more specific terms, it is thought expressed in words which is central to all philosophical arguments.

The origin of a word's meaning has also been assessed from behavioural and physiological points of view. As it was previously stated, analysing the meaning of a word leads to a non referent or elementary concept which cannot be determined any further in linguistic terms. The analysis at such elementary conceptual level is shifted to a different plane which involves the complex components of speech.

Science, on the other hand, is interested in the part of semantics concerned with the sense and reference of predicates and propositions that are part of scientific theories and systems. Science and scientific theory in general has no tolerance for inexact reference to, or inaccurate description of reality. The predicament in science, is that the *significance* of existing linguistic symbols fails to provide accurate concept formation methods for newly procured knowledge. The opposite is just as true, i.e., it is not possible, in certain situations, to ascertain the factual existence of some conceptual proposition signified by a linguistic expression.

Language, in pure scientific sense, is insufficient for describing scientific knowledge, because of the inherent dilemma of logical contradictions in language. A classic example of such contradiction is the *liar paradox* in which the assertion "I am lying" is linguistically valid, but logically does not make sense and has no *truth value*. Therefore, scientific theory has to rely on a combination of conceptual and symbolic expressions to represent scientific theory.

5.1.2 Semantic Foundations

Because of the variety of approaches to understanding semantics, it is not easy to decide on a representative approach which will ensure the contribution of comprehensive semantic understanding. However, it is necessary to provide a complete and pragmatic enumeration of the semantic foundations to be utilized in this work.

The purpose of the following analysis is to come to an understanding of the relationships between words and their meaning. This relationship will be used to improve concept classification and integration by separating the name of a concept from its meaning. Thus, the significance of a concept name will be minimized. Instead, concepts will be examined at the level of their meaning, which is much more reliable than their names. This is simply because names are usually subject to the terminology used at the various contexts of a work environment. In addition, meaning is a better ground for integrating concepts. Because concepts overlap in description, the overlap can be identified through meaning, whereas names can only give weak or no indication at all of such overlap. It is easier, for example, to arrive at the common properties of two concepts once their meaning has been explicitly indicated; on the other hand, the simple indication of the names and form of two concepts would only reveal the common attributes from the point of view of the person evaluating such concepts. The reliance on meaning to provide better grounds for integration has far reaching advantages for conceptual modelling. Such advantages will be discussed further in chapter 8. However, and in order to establish the foundations of the semantic approach, an assessment of the concepts involved is necessary.

The following assessment of basic semantic terminology is drawn from diverging arguments on the subject. But more of Bunge's approach (1974a, 1974b) to semantics has been incorporated because of its clear cut approach to the treatment of the subject.

5.1.2.1 Meaning, designation, and significance

According to Katz (1972), meaning is what semantics is all about. The meaning and significance of *meaning* has been an issue of debate for as far back as the times of the Greek philosophers. Though many theories of meaning have been put forward, it is suggested that a complete theory is still lacking (Allan, 1986). The predicament in *meaning* is that it involves both the components (or words) of an expression and the expression as a whole. In other words, it is adequately possible to arrive at the meaning of a word linguistically and/or ostensibly; but the collective usage of words and expressions escalates meaning to a different level.

In the following, an attempt will be made at clarifying *meaning* in relation to conceptual constructs. It is necessary, in order to secure a satisfactory evaluation of *meaning*, to include two concepts of critical relevance to meaning, namely *signification* and *designation.*

In reality, objects are assigned names to enable their identification. The assignment "Unicorn", for example, is the name given to an idea in mind about a thing possessing properties understood as including {animal, four legged, mammal, single horned}. Whether or not there is a distinction between a name and what it stands for, i.e., unicorn the name and unicorn the animal, is purely philosophical, and is as irrelevant as the factual existence of the named object, i.e., unicorn. Accordingly, it is said that a name designates a construct.

Designation, however, does not provide sufficient meaning, simply because the relationship between designator (or name) and designatum (concept) does not communicate the relationship between conception and reality. This can be better understood by pointing out the fact that some constructs have no names, and imposing designators on such constructs will not provoke any thought. For instance, the construct "Books and Automobiles" has no designator, and is rather described by properties of certain designatums, namely "Books" and "Automobiles". In this case the meaning is comprehensible because both books and automobiles have *significance*. However, the construct "3 and 7", for example, is a construct which has neither a designator nor a signifier, and is thus meaningless even if it were to be assigned a designator, or a name.

In the previous section, *significance* was mentioned in the context of the *significance triangle*. A word, it was asserted, provokes a certain thought, which in turn refers to some fact. The *fact*, in this sense, refers to an acknowledged aspect of reality irrespective of the truth of the fact's existence. With respect to designation, all names which signify designate, but the inverse is false. This in fact is another way of stating that names that are understood to proxy some constructs are signifiers. If they do not signify, then the names are mere designators. This clarifies the naming relationship but does not fully explain *significance*. The relationship between constructs and their meaning has to be clarified in order to understand the significance-meaning relationship. Once meaning is clarified, it will emerge that significance is designation combined with meaning.

Meaning, being the gist of semantic theory, has been defined in accordance with the approach in which it is expounded. Here, the definition of meaning is given according to the *synthetic view* (Mill, 1970), in which *terms* are classified into connotative and non-connotative. According to Bunge (1974a), such distinct classification does not only donate a significant contribution to understanding meaning, but also avoids conflicting with the contention surrounding meaning in main stream philosophy, namely that of the relationship between a name and its holder. Mill's definition of connotative and non-connotative terms is specified and exemplified.

> "A non-connotative term is one which signifies a subject only, or an attribute only. A connotative term is one which denotes a subject, and implies an attribute. By a subject is here meant anything which possesses attributes. Thus John, or London, or England, are names which signify a subject only. Whiteness, length, virtue, signify an attribute only. None of those names, therefore are connotative. But white, long, virtuous, are connotative. The word white, denotes all white things, as snow, paper, the foam of the sea, &c., and implies...the attribute whiteness." (Mill, 1970)

Therefore, an expression is only meaningful if it both denotes a subject and implies an attribute. In other words, a meaningful expression or concept triggers the properties of an object in mind, and points out specific objects possessing such properties. The former characteristic is referred to as the sense of an expression, and the latter is its reference. Accordingly, meaning is indicated as:

Meaning= <Sense, Reference>, or
M= <S,R>

Figure (5.4) illustrates the meaning relation. Therefore, understanding *meaning* requires further analysis of its two main components, namely sense and reference.

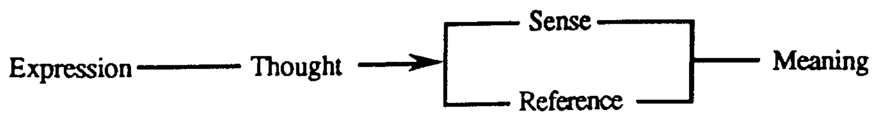


Figure 5.4 The *meaning* relation

Note in the figure above, that the name of the expression is its signifier. If the name does not give the expression meaning, then it is simply a designator and the expression significance is required to provide its meaning. If, however, the meaning of the expression is understood by its name then the construct name is its designator. Figure (5.5) illustrates *significance* in relation to meaning.

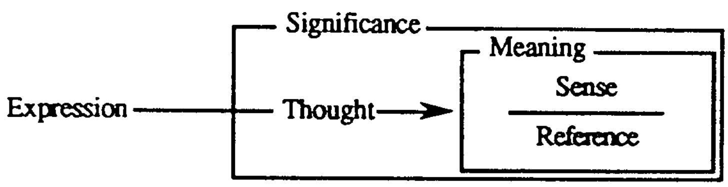


Figure 5.5 Significance with relation to meaning

5.1.2.2 Reference and extension

Reference is the relation between an expression and what or who the expression is about. The relation is defined in the mind of the speaker, i.e., the speaker's knowledge of what is being said is presupposed. Listeners, on the other hand, usually but not always correctly, assume knowledge of what the speaker is referring to. In the case where both speaker and listener are aware of the reference, the topic of a statement can be understood regardless of the statement's truth. For example, when the statement "It is faster than the speed of light" is uttered, unless both the speaker and listener are aware of the referent of such statement, it will not convey any meaning. Here, the speaker could be speaking about, in other words referring to, a rocket, an airplane, car, etc. Note that when "Car" is the referent in the previous example, the reference could be described as successful when it serves some purpose, e.g., exaggeration; but it clearly would have made the statement a false one.

According to Lyons (1977), expressions can refer to individuals or classes. The former is called singular reference, and the latter is a general one. For example, "This automobile" is a singular reference, while "Vehicles" is a general one. Moreover, it is suggested that an expression - given that it does refer - has definite reference when it refers to an individual or a class, and indefinite reference when it does not refer to a specific individual or class. The later classification, however, is less relevant to this context.

Because of the importance of the relation of reference to defining meaning, and since the relation is utilized - in concurrence with the relation of sense - by the suggested approach to identify meaning, reference will be formalized. The formalization is based on Bunge's definition of meaning (this is also true for the formalism of sense in section 5.1.2.3).

Definition 5.1 The reference relation R is a set of ordered pairs of <construct, object>. In addition, the reference of a construct is a subset of the cartesian product of the objects involved in the class. For example, the construct "Managers and Vehicles" has reference over the class of all managers and vehicles.

Definition 5.2 The reference class of a construct C is the set of its referents R(C). Moreover, a construct is said to refer partially to a class c in Ω, when c is part of the reference class of C.

Reference in some cases is made of similar classes, for instance, "Intellectuals" has reference to "Writers", "Lecturers", "Professors", etc. In other cases, reference is made of different classes, e.g., "Army" refers to "People", "Tanks", "Airplanes", etc. In the first case, reference is said to be homogeneous, while in the second it is heterogeneous.

Definition 5.3 The reference class of a predicate is the collection of its arguments. Therefore, if P is a predicate of n-ary predicates with the domain $(A_1 \times A_2 \times \ldots \times A_n)$, its reference class is:

$$R(P) = \left(\bigcup_{1 \leq i \leq n} A_i \right)$$

In other words, the reference of a predicate is the total reference of its participating predicates. For example,

P = "Man is a rational animal"
R(P)= {Humans, Animals}

Corollary 5.1 From the above definition it can be educed that a predicate and its negation have the same reference class. Therefore,

¬P= "Man is not a rational animal",
R(¬P)= R(P) = {Humans, Animals}

Corollary 5.2 The reference class of a *member relation* ($\in$) is the reference class of all sets over which it is defined. In other words,

if $a \in A$, and
$a \in B$, then
$R(a) = R(A) \cup R(B)$

Corollary 5.3 The reference class of a set relation (**R**) is the union of the reference classes of the sets involved in the relation. Therefore, if **R** is a relation between *a* and *b* defined as:

a**R**b, $a \in A$ and $b \in B$, then
$R(\mathbf{R}) = R(A) \cup R(B)$

The concept of reference is associated with the concept of extension. Where *reference* is what an expression is about, extension is a set of concepts that can truthfully apply to that expression. In other words, the extension of an expression is the collection of concepts which result from the valid application of some test predicate on an expression reference. For example, the concept "Land-vehicles" has an extension consisting of all sets of land-vehicles, e.g., "Cars", "Trucks", and "Bikes", since the predicate "is a land vehicle" is true for each of such sets. Extension is, therefore, a set of individual referents determined by both the context and form of the concept.

The relationship between reference and extension can be further clarified with the following example. The expression "Talking-animal" has no extension, but refers to {Animals}. Therefore, while the reference of an expression is not subject to truth testing, its extension is the set of existing referents.

Definition 5.4 The extension of a concept is the part of reference which the truth of its existence can be established.

Definition 5.5 Two concepts are extensionally equivalent if, and only if, they have identical extension sets. In other words, if A and B are expressions, and their extension defined as:

$E(A) = \{a_1, a_2, \dots, a_m\}$, and
$E(B) = \{b_1, b_2, \dots, b_n\}$, then
$E(A) = E(B)$, Iff $a_i \subset E(B) \wedge b_i \subset E(A)$

This, however, does not imply that A and B are equal, only their extensions.

Finally, the reference and extension sets of a concept are not always absolutely defined, but are rather contextually identified. In other words, the definition of a concept reference and extension sets is dependent on the context in which the concept is used. For example, the extension (and reference) of the expression "The creator of the universe" is {God} for most people, but is ϕ for an atheist.

5.1.2.3 Sense and intension

The sense of an expression is as critical to understanding its meaning as reference is. Unlike the reference of an expression, its sense is determined by the speaker and the context in which the expression surfaces.

According to Lyons (1977), the sense of an expression is independent of its referents.

The most direct way of identifying the sense of an expression is by asking the question "what is meant by the expression?" when in fact its reference is already understood but the intension of using the expression is not clear.

It is important to differentiate between the concepts of sense and reference. Expressions with similar reference can have different meaning because the sense of an expression is not implied by its reference. By the same token, similar sense does not indicate meaning similarity, because expressions with similar sense can be stated in more than one way with different referents. To illustrate the distinction between reference and sense in relation to meaning, the following classic example is given to illustrate the first case, namely of co-referential expressions with different sense. Expressions A and B are defined as:

A= "The morning star"
B="The evening star"
R(A) = R(B)= Venus

But since the senses of the two expressions are not the same, the two expressions reflect different meanings.

The second case, where two expressions have similar senses but different reference, is exemplified by the following expressions.

A= "Vehicle"
B="Truck",
and both concepts have the same sense, e.g., Truck,
R(A)≠ R(B), where
R(A)= All types of vehicles, and
R(B)= Trucks

The notion of intension is sometimes used to refer to the sense of a semantic construct. Bunge's (1974a) analysis of intension is extended with respect to the general view of sense.

Definition 5.6 The intension of a construct is a set whose components are drawn from the context of the construct itself.

Corollary 5.5 Two constructs are said to be *co-intensive* when they have the same meaning. That is,

If M(A)=M(B), then
I(A)= I(B)

A well known example of this definition is the following:

If A= "Man kind",
B="Human beings", and
M(A)= M(B), then
I (A)=I (B)={Human beings}, and therefore,
A and B are co-intensive

Definition 5.7 The intension of two constructs is the set theoretical union of their intension sets. In other words, for two constructs A and B,

If A & B, then
I(A & B)= I(A) $\cup$ I(B)

For example,

I(Detailed analysis)= I(detail) $\cup$ I(analysis)

Definition 5.8 If A and B are constructs, and the intension of A is included in the intension of B, then A is referred to as the (or a) *refiner* of B, and B as the *definer* of A. For example, two constructs A and B defined as:

A= "Car"
B= "Automobile"

A *is* B, and is a more specific case of it. Therefore, A is said to refine B, while B defines A.

Corollary 5.5 The intension of the conjunction of two constructs is a subset of the intension of both constructs. In addition, the intension of the relation of disjunction between two constructs is a subset of the intension of the two constructs. More specifically,

I(A $\wedge$ B) $\supseteq$ I(A), I(B),
I(A $\vee$ B) $\subseteq$ I(A), I(B), and
I(A $\vee$ B) = I(A) $\cap$ I(B)

Corollary 5.6 If the intension of a construct is a subset of the intension of another construct, then the intension of both constructs is equal to the intension of the definer of the two. Or,

If I (A) $\subseteq$ I(B), then
I (A $\wedge$ B) = I(B), and
I (A $\vee$ B) = I(A)

Corollary 5.8 The less intensionally defined a construct is the more general its meaning is. The converse is also true. For example, the construct "Mammal" is less intensionally defined than the construct "Man", and is, therefore, the more general of the two.

Definition 5.9 Two constructs A and B are said to be *intensionally independent* if, and only if, they have no *intension* in common. The concepts "Project" and "Student", for instance, are intensionally independent.

Definition 5.10 Two constructs are *intensionally dependent* if, and only if, they are not intensionally independent.

Definition 5.11 The difference in intension (I^-) between two dependent constructs A and B, is an intension set containing the intension of A not in B, and B not in A. Thus,

$$I^-(A, B) = (I(A) - I(B)) \cup (I(B) - I(A))$$

Finally, intension and extension have an inverse relation, in that the more meaning a concept has, the smaller is its extension set. For instance, the concept "Car" has more intension than that of "Family-car"; the former concept extends more than the latter, because it includes all cars.

Definition 5.12 If two constructs A and B are co-intensive, then they are co-extensive. Furthermore, if the intension of a construct A is a subset of the intension of B, then A's extension is included in that of B. That is,

i) If $I(A) = I(B)$, then
$E(A) = E(B)$, and
ii) If $I(A) \subseteq I(B)$, then
$E(A) \supseteq E(B)$

For example, in the first case the intension of "Employee" and "Worker" are the same, and so is their extension. In the second case, the intension of "Manager" is a subset of the intension of "Employee", i.e., every manager is an employee but not vice versa. Accordingly, the extension of "Employee" is larger than, or equal to that of "Manager".

Note that the converse of (i), above, is not necessarily true. Co-extension does not imply co-intension. This is an area of argument in mathematics - concerning set definition - where it is accepted that two sets are equal if they contain the same elements. Semantically this is not granted. Some authors will go so far as to suggest that the case should not be so even in mathematics (Schreider, 1975; Bealer 1982; Hautamaki, 1984). The example usually given to support the semanticist view on this issue is that of two sets given intensionally as {featherless, biped} and {rational animal}. The extensions of each of the two sets equals the set of all human beings. But according to Schreider (1975), such conclusion defies the purpose of the comparison.

equivalent.

The definitions of intension and extension provided in this section will be further discussed in chapter 7 for the purpose of the proposed modelling approach.

5.2 Semantics and conceptual modelling

Conceptual modelling is the process of reflecting interesting aspects of reality in a conceptual form. Objects and their interrelationships are represented by properties of such objects. Such representation takes place regardless of the true or physical existence of objects in reality. The aim of conceptual modelling, in general, is to produce a replication of the reality of the environment under investigation.

There are a number of problems in conceptual modelling which have not received proper treatment in the literature. An obvious one is that of defining concepts in a way that is universal to all users. It is not uncommon, for instance, to have the same concept unknowingly defined in different ways based on its use, leading to redundancy and inconsistency.

Related to the problem of concept definition is that of concept naming. This difficulty is greatest when concepts refer to non-physical objects. The description of a concept in this case, can only be captured by reference to an abstract object(s). Having to identify such objects is at times taxing, especially when it is the case that the conceptual representation has to be as accurate and determinate as possible. Furthermore, the representation has to be capable of adapting to an ever changing reality.

It is an aim of conceptual models to improve database modelling techniques by enabling the inclusion of conceptual semantics, which in turn gives conceptual models additional capacity in reflecting reality. It was mentioned in the beginning of this chapter that many modelling approaches claim to have taken into consideration issues of semantic incorporation. Such approaches fail to point out the degree of semantic inclusion. Such failure is mainly due to the lack of adequate comprehension of the notion of semantics and the relation between meaning and modelling.

It seems that the direction which suggested that it is inconsequential to examine issues of semantics was set in Abrial's paper on data semantics when it was explicitly stated that:

The relation between semantics and modelling should be specified and highlighted before any attempt at proposing a *semantic* approach to conceptual modelling. The specification of such relation will prove advantageous, especially when taking into consideration the fact that semantic inclusion in modelling is desirable. Therefore, further explorations of semantics by database modelling research will provide a chance for further investment in semantic theory, which in turn will result in improved database modelling techniques and utilization.

To come to a clearer understanding of the impact of semantics on conceptual modelling, the concepts of *representation* and *categorisation*, which have significant bearing on both semantics and conceptual modelling, are defined in the following sections in the context of semantic application to database modelling.

5.2.1 Representation and modelling

A data model is a representation of reality. In a database, a data model serves to represent concepts and relationships between concepts for the purpose of storing and retrieving data.

A concept representation is an illustration of nominated properties, in which the nomination is a selective process so as to ensure correct representation of the concept' meaning and form. When concepts are accurately represented, i.e., their meanings are readily comprehended, the process of modelling can be performed successfully knowing that the concepts and relationships modelled are true representatives of objects in reality. This issue is at the centre of the relation between semantics and modelling.

It is acknowledged in information and data modelling literature, that the conflicting views, terminology, and perception in general, in information systems and database design situations are usually the main contributors to the failure of many systems.

The problem of variance in perception over similar phenomena is also recognized in organization literature when the issue of communications is at question. It is explicitly, and indeed rightly, referred to as one of inconsistency of semantic perception. This problem was addressed in chapter 2 (section 2.5) and was described as the major contributor to communication breakdown. In addition, it was pointed out that according to Schneider *et al.* (1975),

Nonetheless, there does not seem to be an effective method of managing such difficulty. An examination of some suggested approaches to resolving the predicament would reveal a lack of true and thorough understanding of the scope of such problem. For example, in a certain case it was suggested that systems should be broken down into sub-systems according to the levels in which terminology is likely to differ; thus, creating a system of inter-communication through some translation routines within the sub-systems. Strictly stated, Clearly, the above suggested approach may lead to increasing the terminology gap when each user community becomes less aware of newly developed terminology in other user communities. In addition, the approach suggests leaving the problem for system programmers to deal with.

Undoubtedly, the best solution that can be achieved is to ensure that the conceptual representation is effective enough to allow for co-existing views based on meaning. In other words, a semantic representation.

A semantic representation of an object is one in which the representation is a relation between the name of the object, i.e., the symbol or designator; the concept symbolized by the designator; the properties representing the designated concept; and the real object. This composite relation is illustrated in figure 5.6a, and exemplified in figure 5.6b using the symbol "Employee" (which happens to be a signifier in this case, because it invokes the idea of an *employee*), which refers to a person with employee properties. Other relations are illustrated in the figure.

The representation of an object is the simple relation of its existence. However, and in order to reflect a network of objects and relationships within a system, a more elaborate form of representation is employed. In the case of conceptual modelling, a system can be diagrammatically and/or schematically represented. Each approach offers an alternative representation of the system. A diagram offers a graphic representation with emphasis on the relationships and connections between objects. A schema, on the other hand, offers a textual representation of concepts and relationships, and their properties.

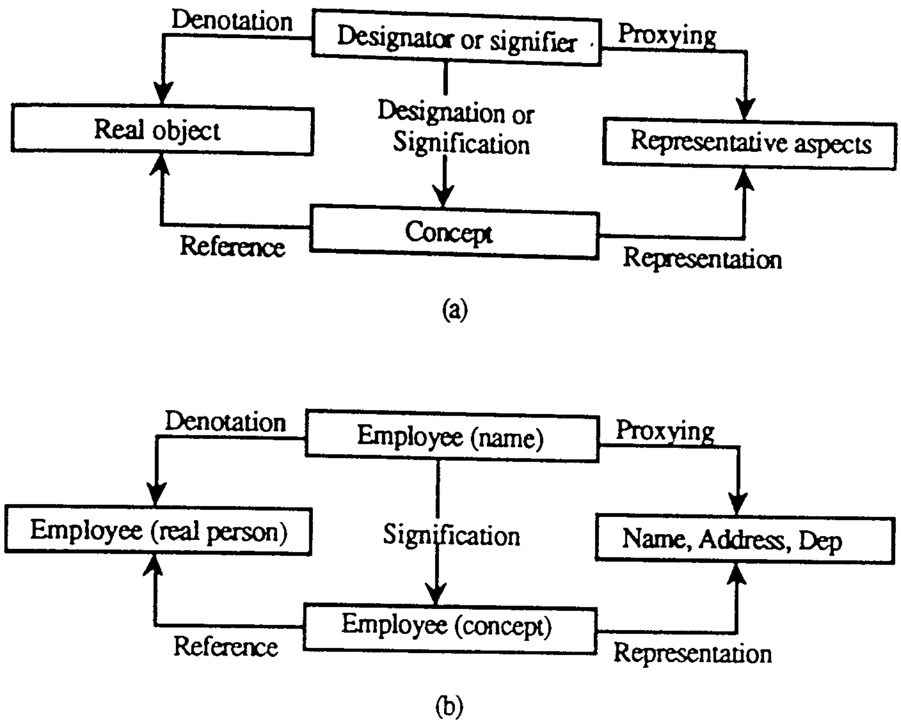


Figure 5.6 The semantic relation of representation:

(a) Representation in a semantic context; and

(b) an example of this relation

5.2.2 Conceptualization and categorisation

The essence of the relationship between semantics and conceptualization is rooted in the definition of semantics. Semantic theory has always been implicated with concept acquisition, formation, and classification which constitutes the fundamental theory of knowledge in general.

Conceptualization is the process in which attributes of objects are recognized and organized in the mind in the form of an idea. A concept is, therefore, an idea about an object.

Object recognition, on the other hand, is a complicated process in which people are constantly engaged. The process serves the purpose of acquiring and memorising knowledge about newly encountered objects. Nonetheless, object recognition by perception cannot cope with the unlimited number of, and variety in objects in existence. To overcome such difficulty, objects are classified into categories in which objects of common attributes are abstracted and aggregated. Such categories are referred to as *semantic categories* (Howard, 1983).

Classes and categories vary in degree of detail and content. Some are more abstract, detailed, or defined than others, just as objects can be classified according to different attributes. The structure of classes and categories in human memory is critical to understanding the process of learning and knowledge acquisition in general.

Classification and categorisation are also basic to understanding the meaning and use of natural language. Natural language, in turn, is the main tool for communicating ideas and impressions about reality. In this sense, language is an object of investigation by semanticists and philosophers interested in identifying the origin and use of language on the one hand; and on the other by psychologists and behaviourists with the intension of forming a theory of human learning and memory.

Rosch & Mervis (1975) suggested that categories are formed with distinctive structure correlating to that of the real-world. In other words, the relation between attributes of objects in reality is the basis underlying the structure of object categorisation. Rosch & Mervis view natural semantic categories as "networks of overlapping attributes."

In other work, Rosch (1973, 1978) further suggested a vertical and horizontal relationship between categories of similar classes according to object attributes. For example, in figure 5.7, the class of trees has a vertical relationship with "Oak", "Maple", and "Birch", which indicate a level of category inclusion. On the other hand, the horizontal relationship between "Red-Oak" and "White-Oak", reflects the segmentation at a certain level.

Figure 5.7 Examples of vertical and horizontal relations between classes (Howard, 1983)

The above notion of vertical relationships between categories bears some resemblance to Korzybski's work discussed previously (section 5.1.1). Both approaches suggest a stratification of objects based on perception and properties. In more specific terms, Roach's approach stratifies categories according to their internal structure and the degree of attribute inclusion. On the other hand, Korzybski's levels are of linguistic descriptions based on the observations of objects in reality. The resemblance between the two approaches is not fortuitous, because it is an indication of an unambiguous relationship between semantics and perception. Such relationship, though not explicit, is usually presumed.

It would be lucid, however, to ascertain the existence of a third dimension to the relationship between semantics and perception. Namely, a relationship between semantics and the levels in an organization hierarchy (discussed in the previous chapter). A brief scrutiny would serve to establish the validity of implicating semantics, perception, and organization hierarchies in a specific inter-relational proximity with critical consequences on database design as implied in this work.

It was specified, in the previous chapter, that the formation of intrinsic structure hierarchies is based on the ability of people to perceive and abstract reality. Jaques (1978), linked hierarchical levels with individual work-capacity, which is in turn related to the time-span of activities performed at a certain level. The relationship between work-capacity and time-span is defined as follows

The relationship between structure hierarchies, and abstraction and conceptualization levels has been illustrated elsewhere in similar terms (Goldstein & Scheerer, 1939; Cherry, 1957; Harvey *et al.*, 1961; Rowbottom & Billis, 1977, 1987, 1989). This is pointing out the strong link between conception and categorisation on the one hand, and abstraction and the levels in social organizations on the other. Such a link will emerge in the course of the successive chapters, and will be restated in more definite terms in the following chapter.

5.3 Conclusion

This chapter explained the main semantic foundations required to understand the foundations and application of semantics to conceptual modelling in general, and the proposed modelling approach in particular. It is necessary to explicitly point out the meaning of semantics in order to achieve a better utilization of concept meaning, definition, and representation. This, in the end, will result in establishing a semantic modelling approach based on definite semantic principles. It will also, as will be shown in the following chapters, ensure completeness and consistency of the database, and accuracy in reflecting reality.

Chapter 6

SECOM: Introduction and methodology

The examination of contemporary conceptual modelling techniques illustrated the inadequacy of such methods in dealing with the intricate nature and implications of organization structure. The presence of such inadequacy is more evident when considering the impact of organization structure (chapters 2 and 4) and semantic theory (chapter 5) on the database.

In this chapter, a new modelling approach is introduced. The model has emerged as a result of analysis of conceptual modelling in general, and of all issues of impact on data utilization in an organization. The proposed approach is aimed at presenting a practical and dependable method of designing databases. Because organization and semantic considerations are incorporated into the definition and structure of the proposed approach, it is anticipated that the database environment itself will gain improvements in management performance, communication, and general organization functioning.

6.1 Introduction

SECOM is a three dimensional conceptual modelling approach. The first aspect of SECOM is semantic modelling. The previous chapter underscored the insufficient treatment of semantics in database design. It was specifically stated that the term semantic has been loosely employed. It was also indicated that justified reference to a model's semantic capability necessitates the awareness and incorporation of pertinent semantic principles. Accordingly, the proposed approach is committed to making the most of semantic theory by utilizing relevant semantic concepts delineated in the previous chapter.

The second aspect of SECOM, is one which is investigated by mainstream database modelling approaches, namely data modelling. Conceptual modelling, in this context, is referred to as application modelling, and it is defined as the process of specifying, and more essentially relating concepts as required by the application environment. Because of the advantages of the ERM, its approach to relationship definition and management will be incorporated in this work.

Organization structure is the third aspect of SECOM. Because SECOM is especially intended for the design of large organization databases, its composition takes into consideration the organization structure which was established as a major contributor to the constitution of information flow and processing. Incorporating structure considerations into a modelling approach is not an original idea. Information systems literature, for example, embodies an extensive study of system modelling based on organization structure. However, database design has not taken advantage of the findings in information systems literature which relate information usage and flow to system structure. Aside from the few approaches mentioned in chapter 3, the concept of organization structure hardly surfaces in database design literature.

In accordance with the three aspects of SECOM, three types of relationships exist within the modelling approach. The first, is the relationship between concepts according to their semantic definition. This relationship depicts intrinsic links between concepts. The second, is a relationship defined according to the use of concepts in a database application, and is in accordance with relationships as defined in the ERM. Finally, a relationship of structure hierarchy between concepts which will serve to ensure relevant representation of concepts within the organization hierarchy.

Because of the degree of mutual exclusiveness among the three dimensions of SECOM, each will be presented in a separate chapter. Thus, the following chapters will consecutively present the semantic, application, and organization aspects of the proposed approach. In addition, a case of SECOM application is presented to illustrate its methods of execution.

6.2 Design methodology

It was mentioned in chapter 3, that the database design process is divided into logical and physical phases. To simplify the process, the two main phases are carried out in sequence. It is important that each stage of the process takes into consideration the orientation of the whole process. In other words, the design process is a successful integration of effective component methods. Unless the design method of some segment fits appropriately within the whole of the process, the method and the whole approach is virtually ineffectual.

Logical design methodologies were also introduced in chapter 3, in the context of a framework in which the process can be carried out with as much definition and specificity as made possible by the systematic nature of the methodology itself. Because the approach introduced in this chapter is mainly based on user definition of data, the methodology utilized in this context is the *view modelling and integration* (VMI) put forth by Navathe & Schkolnick (1978) (also in Navathe & Gadgil, 1982).

In VMI, the main object is to produce a "'community view' of a single database which truly reflects the aggregation of views with different expectations, backgrounds and technical expertise" (Navathe & Gadgil, 1982). The design phases, according to VMI, are view modelling, view integration, model optimization and analysis, and model mapping. This work is primarily concerned with the first two phases. View design and integration are introduced below from the point of view of the SECOM.

6.3 The definition of user views

The term *user* is utilized in reference to a person (or persons) whose work requires an interaction with the database. It is irrelevant whether or not the user is actually performing the task of interacting with the database, i.e., personally inputting or outputing data. Therefore, a *user view* is the definition of user data requirements for performing certain functions. Naturally, and because a user may preside over multi-functional tasks, a user can have more than one view at any time. The term view itself, on the other hand, is simply defined by Date (1983) as "a virtual table", where the constituents of the table are data elements drawn from the database.

McFadden (1985) defined a user view as "a subset of data required by a particular user to make a decision or carry out some action". This definition refers to a user view by virtue of its function. On the other hand, the structure of a user view is defined as "a model or representation of data requirements for one user" (Fleming & von Halle, 1989). It can be said that the former definition is the concern of the requirement specification phase of the design process, while the latter is an issue for conceptual modelling.

It is worth mentioning at this stage, that the progression from defining user requirements into subsequent design phases is an area that is lacking in definition and specificity. Its being so is mainly due to the fact that the progression itself is neither a recognized independent phase, nor part of any of the established phases. Though it is not the intent, in this context, to address such predicament, it is strongly felt that examining the nature and substance of user requirements is the key to ensuring a sound succession from the first step of the design process into the next one. The following analysis of user views is from both requirements and modelling perspectives.

User data requirements, along with other related details, constitute the contents of the database. The data required by users is necessary to enable the performance of specific tasks and the making of certain decisions. According to McFadden (1985), user views are identified by "reviewing tasks that are performed or decisions that are made by users and by reviewing the data required for these tasks and decisions."

Clearly, therefore, data elements involved in any view are part of the organization information system. In other words, data elements identified for storage in the database are the constituents of an application area within a broader view of the organization information system. Although data is defined by dissecting the information system, its structure in the database has to be capable of presenting itself in the manner in which it is utilized in the information system, i.e., relative to the user requirements. Relevance of data to user needs has been emphasised as a matter of critical importance to the effectiveness of database utilization. To a great extent, this is one of the main challenges to the design approach proposed in this work.

6.4 The level approach and user views

It has been established in organization and information systems literature, that information required by a user has certain characteristics relative to the level in the organization hierarchy in which the user is positioned. Although most writers classify the characterization of information - in relation to the user position - in accordance with the traditional three managerial levels structure, it is not appropriate to exclude the presence of continuity in the changing nature of information characteristics within the stratification of a managerial hierarchy. In other words, the change in the quality and quantity of information throughout the organization structure is gradual and spreads over more than three levels. Two issues are addressed to clarify this argument: the relationship between information and users, and management levels.

The quality of information used in an organization is related, in terms of level of detail and magnitude, to the activity utilizing such information. Activities, on the other hand, vary with respect to their information utilization in accordance with the managerial level in which they are performed. The issue of information relevance to, and relationship with organization activities has been thoroughly addressed in the literature with the underlying presumption that detailed information is required for activities at lower levels in the organization, and less detailed abstract information are required at higher ones. Therefore, it will suffice for the purpose of reference to point out the illustration of the topic presented in section 2.6.

The other issue with distinct consequences on information characterization is organization management levels. It is presumed in the literature, as was stated in chapter 2, that an organization hierarchy consists of three managerial levels; namely operational, tactical, and strategic levels. However, there seems to be an inclination towards overlooking the fact that management stratification based on the levels was in conformity with a collective perspective over the possible types of decisions made within a specifically intended classification of organization activities. As suggested by its proposer (Anthony, 1965), the approach was intended as a framework for planning and control systems. Yadav (1983) pointed out the inadequacies of the three level approach in stating the following,

> "Anthony's main contribution is the conceptual framework which recognizes the fact that information needs are different at different levels of managerial activities. This framework provides insight, but is difficult to apply in determining information requirements. In the real situation, there are no clear-cut boundaries for different levels of activities.

> Furthermore, information requirements are affected by organization structure and processing in addition to the levels of activities".

Ciborra (1981), on the other hand, described organization analysis and information systems design methods as being,

> "locked into the conception of the organization and its information systems as a pyramid composed of three layers: operations management, control, and strategic planning."

Others would even go to the extent of describing the traditional three level approach as no more than,

> "useful for discussion purposes" (Shahabuddin, 1987).

Nonetheless, reference is generally made to the three managerial levels in an organization irrespective of the context in which management levels are discussed. The only possible justification for doing so, is *a* definition of management which relates that management *is* decision making.

Two questions are to be asked at this point to appraise the legitimacy of utilizing the three level management approach for the purpose of information systems analysis for database design in general (in contrast to information for decision making). First, is decision making exclusive to managerial tasks? Second, is decision making the only managerial task? The answer to both questions is no.

In the case of the second question, the answer is no because there is more to management than just decision making (see other definitions of management in section 2.4). But this question is the less relevant of the two. In the case of the first question, and in the same way in which at times management is defined as the process of decision making, decision making is also part of every task there is to be performed. Therefore, it can be said that each individual in the organization is a decision maker. This was explicitly stated in chapter 2, where Pfiffner & Sherwood

(1960) defined an organization as "...an extension of individuals making choices," indicating the extent of the decision making process.

The above leads to the conclusion that management levels as defined by Anthony (1965) cannot provide a definitive basis for classifying information for the purpose of providing specific and clear-cut reference to the degree and scope of information relevance to a user. Therefore, the alternative method of identifying hierarchy levels according to the work-level approach described by Rowbottom & Billis (1989) will be utilized, and henceforth, will be used in association with reference to a user position in an organization.

6.5 User views in SECOM

The number of levels suggested by Rowbottom & Billis (1989) is five in most cases, and seven levels in very large organizations (which is in accordance with Jaques' stratification approach, also presented in chapter 4). This, as was previously pointed out, does not exclude the possibility of the existence of smaller number of levels in some structure hierarchies, which is evident in the cases of small organizations and very wide-span structures.

The organization hierarchy, from here on, will be described as having an *n* number of levels, where *n* is an integer between two and seven inclusive. Note that in the case of a single level hierarchy, utilizing a design approach on the scale of SECOM would prove counter productive, because it is liable to being over-elaborate for dealing with the design of small organization database.

Accordingly, a user view will be referred to by a view index which will indicate the view number and the level (of the user position) in which the view is utilized, i.e.,

Vl,n

Where l is the view number, and n is the level in which the view is utilized.

The view index will serve the following purposes. First, a view numbered according to a user will be distinguished as belonging to that user. Second, because a view is traceable to a user, it is possible to re-introduce a user view at any point in time for update purposes. Third, a view indexed according to an organization level will project the position of the user, which in turn will imply the kind of relationships and constraints to incorporate among concepts within user views in order to account for the implications of organization structure on concept representation. Finally, a concept can be related to a user through the user view, and thus allowing for the existence of a degree of variety in the definition and - more important - the constraints associated with concepts based on various user specifications.

A user view has another distinction which is going to be exploited by SECOM. A view, as mentioned earlier, is the mechanism by which data is made available to users for the fulfilment of certain tasks. Within a wider perspective, views are tools utilized to facilitate the performance of organization activities. Sen (1979) suggested that a view inherits, vicariously, the responsibility of contributing to the successful performance of organization activities. A user view, according to Sen, is part of the larger picture of an activity. In more specific terms,

> "Each person (or group) has a separate responsibility and also a definite data requirement to meet that responsibility. In other words, he has a definite view of the data. Thus, the view is not only the data requirements of an individual in the department, but also it has a definite responsibility attached to it. A view is thus a subset of an activity."

Based to the above perspective, a relationship can be established between views based on relationships within an activity and between activities. This *superficial* relationship can be delineated by a network of database views in which a user view contributes to the completion of an activity or part of it.

Such aspect of database views can be considered from an even broader perspective. Namely, that relationships between views inherit certain constraints within and among activities. The constraints are acquired from within the definition of an activity as implied by the level in which it is performed. In other words, the nature of an activity and the level in which it is performed implies constraints on the data required for the fulfilment of such activity. What is essential here, is to procure the implications of considering user views from organization activities' perspective. More specifically, it is the abstraction of concepts within views as implied by user positions in the varying levels of the organization hierarchy that is central to the SECOM approach.

Chapter 7

SECOM I: The semantic perspective

The semantic perspective is concerned with establishing the rules for defining intrinsic semantic relations between concepts. Such rules, as will be illustrated, will ensure the creation of a time-independent conceptual model which will constitute the skeleton upon which higher levels application and organization relationships are established.

7.1 Introduction

Conceptual modelling is the process of constructing a system of concepts representing reality. What makes a conceptual model semantic is the fact that such representation is based on conceptual semantic definitions. A conceptual definition from a conventional view is representation itself. In other words, a concept definition is based on properties elected to represent that concept. De *et al.* (1982), for example, suggested that because data is abstracted through user perceived properties, a concept is defined as "an abstraction regarding something that can be distinctively identified by its properties or attributes." This definition, especially when perceived from a low-level implementation perspective, implies that a concept is only what is being represented. Ultimately, such conviction has impeding consequences on concept integration by overlooking properties that are necessary for the composition of a dependable conceptual model.

Simple representation, therefore, cannot provide sufficient methods of concept definition. Nor indeed can interpretation - dependent on representation or otherwise -

deliver a competent method of concept definition and interrelation identification. Interpretation, as was indicated previously, is subject to a complicated process involving both philosophical and psychological aspects. It is unsuitable to rely on personal interpretation of concepts to serve as basis for concept definition, and conceptual modelling in general. Figure 7.1 illustrates the inadequacy of relying on the simple name and representation of a concept.

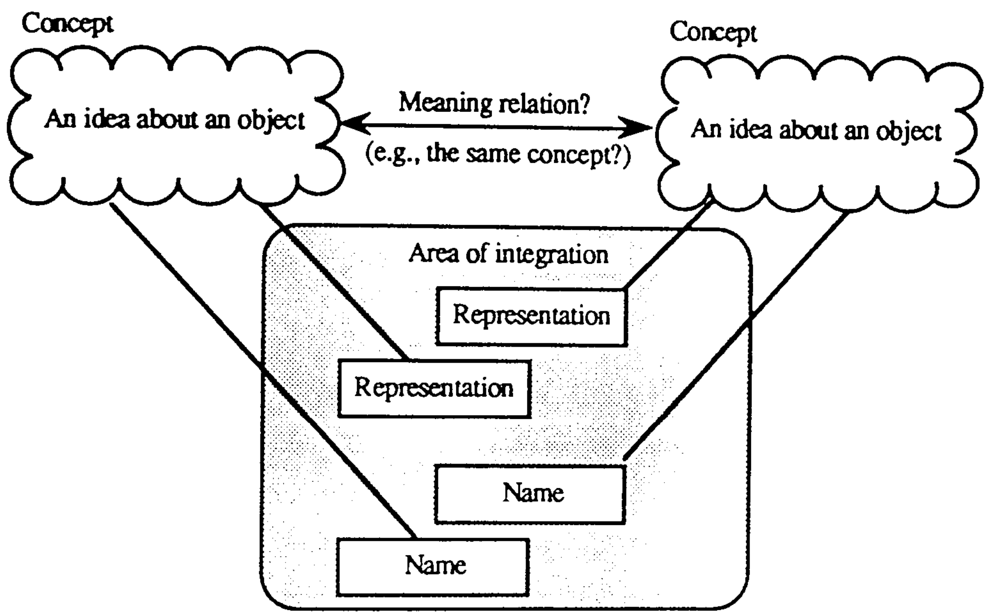


Figure 7.1 Current modelling approaches to concept definition and integration

The figure implies that a concept is more than a name or representation, and accordingly, conventional concept integration methods, i.e., integration based on representation and name, are bound to encounter problems of inconsistencies because they fail to give proper considerations to the concept itself. Users may have different names for similar concepts, and require different representations of such concepts; resulting, in certain situations, in giving the impression that different concepts are integrated while the integration is of different representations and names of the same concept.

There seems to be a consensus among database specialists and users over the view which relates that unless all users agree on the meaning of database representations, database utilization is destined to failure. This is because it is the integration of various views over the same data in a database environment which gives rise to the delicate process of integrating such views in a way that does not infringe on the substance and/or structure of any single view.

Weber (1977) explained that users have different views of reality because of the variety of individual mental models. As such, a competent database utilization is intensely dependent on the mutual agreement about the information contents of common parts of different mental models. Thus, Weber made it a representation requirement that user views be considered on a broader level of analysis than that of a simple subset of the database:

But for one restriction, the above quote reflects the general direction of the approach suggested in this work. Here, views (or users) are not seen to have control over observation of reality. Rather, the contrast in view constitution is acknowledged as being a by-product of the formal stratification of work hierarchies.

It is clear, therefore, that effective database design has to rely on a comprehensive system of concepts that is not solely dependent on representation, and is capable of allowing for the coexistence of database views. Such a system, or conceptual model, can be realized through the application of semantic theory, as will be illustrated in this chapter.

7.2 Principal conceptual foundations

It is important before venturing into the definition of concepts and conceptual modelling to establish the basic terminology involved in such definition. The term concept has been defined previously as an abstract entity. As such, a concept is something that cannot be precisely defined, i.e., it can not be specifically pointed out. Therefore, and in order to enable conceptual analysis, concept name, meaning, and representation will be utilized as pointers to the concept itself. Such pointers, or representatives will be examined within a specific framework that is especially devised to enable conceptual analysis, and ultimately, conceptual modelling.

Before presenting the structure in which concepts are defined, it is beneficial to reconsider Ogden & Richards' (1938) significance triangle (figure 7.2) which was presented in chapter 5.

In the figure, there is a clear indication of the distinction made between three levels of semantic analysis. Thoughts or concepts are treated on a different level than one in which symbols used to *symbolize* such concepts are considered. The objects *referred* to by the concepts are treated on a third level.

TRUE

Figure 7.2 The significance triangle (Ogden & Richards, 1938)

In semantics, the separation of symbols, concepts, and referents is considered the key to resolving an ancient perplexity surrounding words, meaning, and language in general. But what is important here, is the contribution that could be made by adopting the mentioned considerations in conceptual modelling. To find out what benefits there are in doing so, further scrutiny of each of the aforementioned aspects of the triangle is necessary.

To begin with, the separation of symbols or terms from concepts results in contracting the meaning gap between a term and the concept it triggers in the mind. Thus, two terms can be better evaluated with respect to their meaning by examining the concepts they stand for in addition to the terms themselves. But even more precise the evaluation of meaning can be rendered by examining the terms which stand for the concepts, which in turn refer to objects in reality. Put differently, if objects in reality can be examined independent of concepts and terms, then no need would emanate for considering either concepts or terms in the evaluation of meaning. For example, and on a basic level, when two people examine a physical object, the need to speak of their agreement on the manifestation of the object under observation is redundant. However, such is not always the case; objects are not always physical in nature, and thus, mental and linguistic frameworks are inevitable.

Keeping in mind the importance of the collective agreement among users over the meaning of database representations, a system of conceptual definition and structure has to be devised to make available an effective and methodical approach to evaluating meaning for the purpose of building mutually accepted representations, or a common conceptual schema. In view of the elaborate dimensions of meaning stated previously, a minimum requirement of such system would be its awareness of such dimensions. Ultimately, and in order for an acceptable - and not necessarily exact - representation of reality, a conceptual model has to allow for variance over database views by permitting contrast in concepts, while at the same time ensuring exact reference to reality.

For an effective model, based on the above view, to materialize, each of the semantic components presented earlier has to be addressed individually. In other words, terms or names of concepts have to be considered detached from concepts, in the sense that they are only tokens; concepts and their meanings have to be defined within a conceptual system in relation to the reality of their factual existence; and finally, concept representations are to be linked to the concepts they represent. In relation to the conventional approach to concept modelling and integration presented in figure 7.1, the new approach will examine concepts according to their meaning (i.e., intension and extension), as illustrated in figure 7.3.

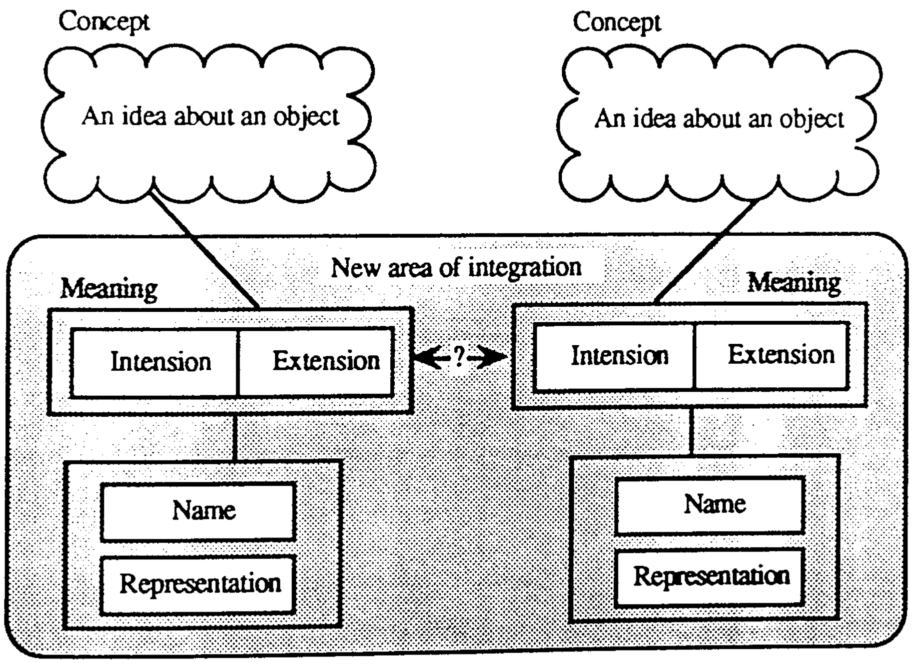


Figure 7.3 Concept definition and integration according to SECOM

The need to evaluate a concept - using its name, intension, extension, and representation as bases for such evaluation - leads to the concept construct illustrated in figure 7.4 (note that T stands for representation). Thus, concepts based on such constructs will be utilized for the definition of the first level of the conceptual model.

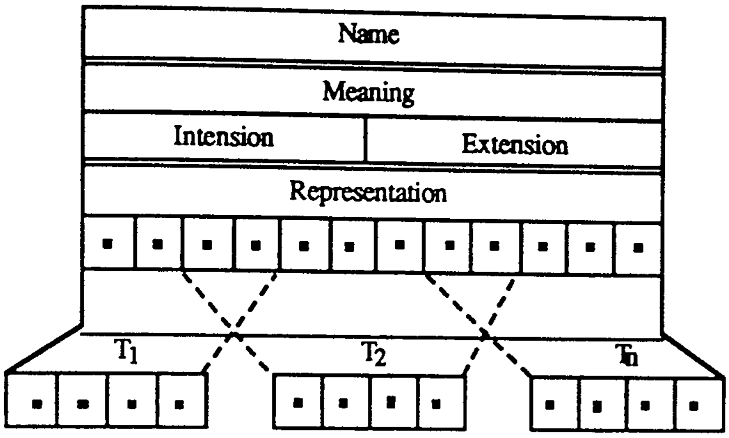


Figure 7.4 The construct of a concept

7.2.1 Naming

Names, in general, are assigned to concepts to point out their physical existence. The name of a concept is a substitution for the specification of a collection of observable concept properties (and not necessarily a complete one). Thus, "Lecturer", for example, stands for a group of properties possessed by a person in a lecturing position.

However, and because some concepts are abstract in nature and, thus, lack concrete peculiarities, this is not always the case. Therefore, to make the previous assertion more accurate, it must be said that a concept name relates specific properties of objects in reality, be they physical or otherwise.

Bunge (1967) presented a clear depiction of the relationship between a concept and its name. In figure 7.5, two relations demonstrate the connections among three distinct levels contributing to the semantics of linguistic terminology.

Figure 7.5 Meaning components and relationships (Bunge, 1967)

In the figure, the first level embodies the words used to stand for concepts, and this level belongs to natural language. The second level is that of concepts, and it belongs to mental models. The final level distinguishes reality. The relations among the levels are those of designation and reference, and both were discussed in chapter 5.

What can be elicited, here, is the insignificance of *signifying* words or terms (in opposition to non-signifying ones, e.g. "and", "or", etc.) in isolation of both the concepts they designate and the facts they stand for. Because objects can be referenced by a number of terms, and concepts can be designated by more than one term, words used for reference to objects in reality should not systematically be made to stand for concepts referring to the same objects; nor should words be incontrovertibly endorsed for reference to objects irrespective of the concepts the words signify. In other words, neither words nor referents alone can extend an exact indication as to the distinctness of concepts. As an example of the first case, the term "Vehicle" referring to the real object "Car" cannot guarantee the designation of the concept of "Vehicle"; and of the second case, the term "Publication" which designates the concept "Newspaper" can refer to the object "Book" instead (or in addition to).

Therefore, separating names from concepts (or at least the awareness of the difference) will ensure the narrowing of the scope of meaning analysis of concepts definition and integration. It will also serve as an effective tool for resolving naming anomalies, concept classification, and abstraction definition.

7.2.2 Intension

Intension in the context of meaning was discussed under the semantic analysis presented in chapter 5. Here, intension in the context of conceptual modelling is presented for the purpose of illustrating the applicability of intension to concept structure and definition.

The *intension* of a concept is the set of properties contributing to the precise identification of that concept. Therefore, the specification of concept properties is essential for the determination of its intension and meaning.

It is, however, not practical - especially on an implementation level - to enumerate and encode what could amount to a very large number of properties. Such enumeration is at times impossible, as in the case of indeterminate or infinite sets of objects. Nonetheless, it is possible, according to Bunge (1967), to procure properties that are sufficient for the identification of a concept.

Such specific properties are referred to, in the above, as earmarks. The intension of a concept will be presumed to be the set of earmark properties. Therefore, for a concept C, the intension is defined as:

$I(C) = \{P_1, P_2, \ldots, P_n\}$, where P_i is an earmark property of the concept C

In addition, and because a concept intension may include the whole of another concept's properties, the intension of a concept is permitted, for practical reasons, to include the name of another concept whose properties are also included in the defined concept. For example, the concept "Employee" includes all the attributes of the concept "Person", as such, the intension of "Employee" is indicated as:

I(Employee)= {(Person), P1, P2, ..., Pn}

This would not only reduce a concept intension, but will also be useful in determining the important semantic relation of concept inclusion (to be discussed in a subsequent section).

Moreover, and to simplify matters further, an intension is permitted to include sentences and/or expressions similar to those used in concept description. Therefore, the concept "Employee" could be intensionally defined as:

I(Employee)= {(Person), on the employees list}

It is worth mentioning that although it might seem as if there is little difference between intension and description - which is the method currently used to designate entities or concepts - the difference, though subtle, is significant, and is recognized in semantic theory (Scott, 1967). Briefly, it is stated that *description* refers to properties that are not necessarily possessed by a concept. Rather, description is often influenced by personal prejudice. Moreover, when accurate, a person's description of a certain concept is not usually equal to another person's description of the same concept. In semantic theory, there is an additional difference between intension and description which involves the truth of a concept existence. This aspect of the difference is not relevant here, and thus, will not be pursued.

Having stated the difference between intension and description, it is important to understand that two persons' definition of a concept intension are not absolute either. Nonetheless, the approach to defining a concept with view to the definition of intension would provide a better indication of the concept in mind.

7.2.3 Extension

Extension is the other component of meaning. Its function, as outlined in chapter 5, is to complement intension in affirming meaning. Intension alone is not a definite indicator of the concept's meaning. For example, the concepts "Manager" may be defined in the same way by two different individuals, e.g., {(Employee), superior of other employees}. However, if the same individuals were to be asked as to who in their view is a "Manager", they may well give different answers, e.g., {Foreman} and {Chairman}. Clearly, the two concepts are different, hence the importance of extension.

There are other advantages to utilizing the concept of extension than in its use as a meaning component. Such advantages will be discussed at a later stage. However, a readily distinguished usefulness of employing extension is in its indication of concept overlap. This is demonstrated in the following example.

Supposing that in a modelling environment the concepts "Employee" and "Student" where defined as follows:

I(Employee)= {A person on the company pay-roll}
E(Employee)= (Employee)
I(Student)= {A person on the university student registration list}
E(Student)= (Student)

Under the above circumstance, the extensions of the two concepts indicate their mutually exclusiveness, i.e., the two concepts are disjoint.

However, if it were the case that an employee can also be a student and not vice versa, then, this would have instead been expressed as:

I(Employee)= {A person on the company pay-roll}
E(Employee)= (Employee)
I(Student)= {A person on the university student registration list}
E(Student)= (Student, Employee)

Such, would imply that an employee can be (and not is a) student. This, as can be clearly perceived, can be extended to exemplify further properties of each concept.

The implication of such a direct approach to indicating the applicability of concept definition is far reaching, especially in terms of indicating the direction of concept overlap. This will be demonstrated in the context of semantic relations, the topic of a subsequent section.

7.2.4 Representation

From users' point of view, representation is the most important aspect of the database. It is the means through which users acquire needed information. It is this definition, from a design perspective, which makes representation an issue critical to the successful utilization of the database.

Conceptual representation is the specific portrayal of certain concept properties for the purpose of depicting aspects of reality that are of some interest to users. As in the case of a concept name, the representing properties usually point out the concept represented. This is especially true when such properties are descriptions of physical attributes of the concept. However, some attributes are logical in nature; for instance, the concept "Employee" may be represented by the attribute "Skill" which is not a physical description of the concept itself. But even more elaborate is the case when the concept itself is non-physical, i.e., abstract, as in the concept of "Management".

In reality, representation is usually subjective and is generally dependent on personal interpretation. In this sense, characteristics of represented objects are designated according to a personal view of the object reality. However, in a database context, representations are vigorously objective, i.e., they are pre-defined to serve specific purposes and, therefore, subjectivity is irrelevant. It is worth mentioning here, that the definition of representation, as reflected above, revokes the reference of a mirror or a reflection of reality usually made to a database. The database is rather a reflection of certain aspects of reality that are of specific interest to database users. In other words, there is no place, in a database environment, for personal interpretation. In addition, it is inappropriate to speak of the interpretation variety of database representations at an implementation level. But this is not to discount the impact of interpreting reality on the part of users at database definition stages, i.e., requirement definition and conceptual modelling.

Aside from establishing accurate database conception, there might not be obvious benefits to adopting the suggested view. Nonetheless, the separation of the concept of representation, as being a reflection of reality, from it being a tool for conveying the description of interesting concept characteristics ensures the effectiveness of concept integration on the basis that representations are not concepts, and therefore, should not take part in the integration process as such. Taking such a position would secure the integrity of the database by preventing conflicting views from inhabiting the database, covertly or otherwise. As will be illustrated at a later stage, a better method of dealing with conflicting views will provide means for the explicit statement and management of such discrepancies, which will eventually abolish the need to push concepts into sustaining more meaning and representation than credibly or practically possible.

What the aforementioned leads to is at the root of database design predicaments encountered at stages of representation, modelling, and integration of concepts; and ultimately, is at the centre of the repercussions of such difficulties on database utilization. The predicament can simply be stated, as it has been in the literature, as follows. A number of users at times require varying representations of the same concept. What this results in is a context of inconsistency and eventual redundancies. Such problem has been recognized long ago. Nonetheless, there does not seem to be an effective solution (yet), as clearly indicated by Jansson (1985).

To tackle this problem, representation in the context of SECOM will be dealt with at a different level than that of the concept represented. In more explicit terms, representation will be considered as a relationship between nominated properties and the concept represented. Such distinction will allow for the coexistence of multiple representations of the same concept. Naturally, concept representations may overlap. Figure 7.6 illustrates the relationship between concept properties and concept representations.

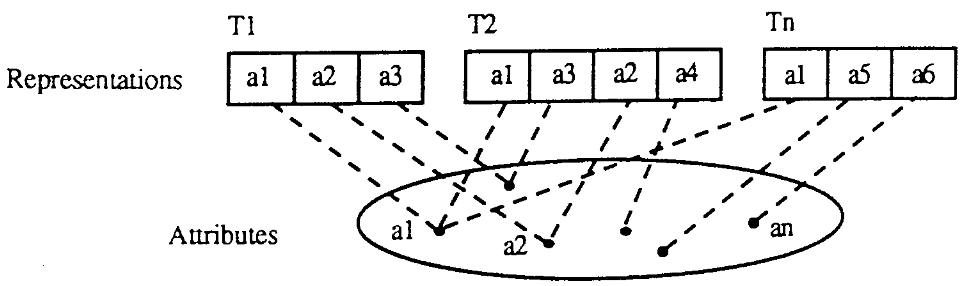


Figure 7.6 The relationship between a concept and its representations

Within the framework of the proposed approach, concepts, in terms of their representation, will be defined as follows.

Definition 7.1 A concept has a number of properties, each of which indicates a solicited peculiarity of that concept. Therefore, the concept C_i is defined in terms of its properties as:

$C_i = \{a_{k,i}\}$, where $a_{n,i}$ is the k^{th} property of the i^{th} concept; $i > 0$ and $k > 0$.

In terms of its representation a concept is defined as:

$C_i = \{T_{m,i}\}$, where $T_{m,i}$ is the m^{th} representation of the i^{th} concept; and $m \geq 0$.

A concept representation, on the other hand, is defined as:

$T_{m,i} = \{a_{k,i}\}$, where $a_{n,i}$ is the k^{th} property of the i^{th} concept; and $k > 0$.

Note that the number of concept representations is allowed to be zero (indicated by $m \geq 0$), i.e., a non-represented concept. This is to enable the existence of concepts in the database definition, even when they are not in use. However, no representation can be an empty representation.

Note, also, that properties within each representation are indexed according to the concept identifier and not to that of the representation. Therefore, the set of non-repeating representation attributes form a proper subset of concept properties. For example, the concept "Employee" can have the following representations:

T1= {Emp#, Salary, Dep#, Dependents}
T2= {Emp#, Address, Gender}
Tn= {Emp#, Qualifications, Dep#}

The concept is then said to have the following properties:

C(Employee)= {Emp#, Address,Age, Gender, Qualifications, Dep#, Salary, Dependents}

Figure 7.7 illustrates the relationship between the concept "Employee" and its representations.

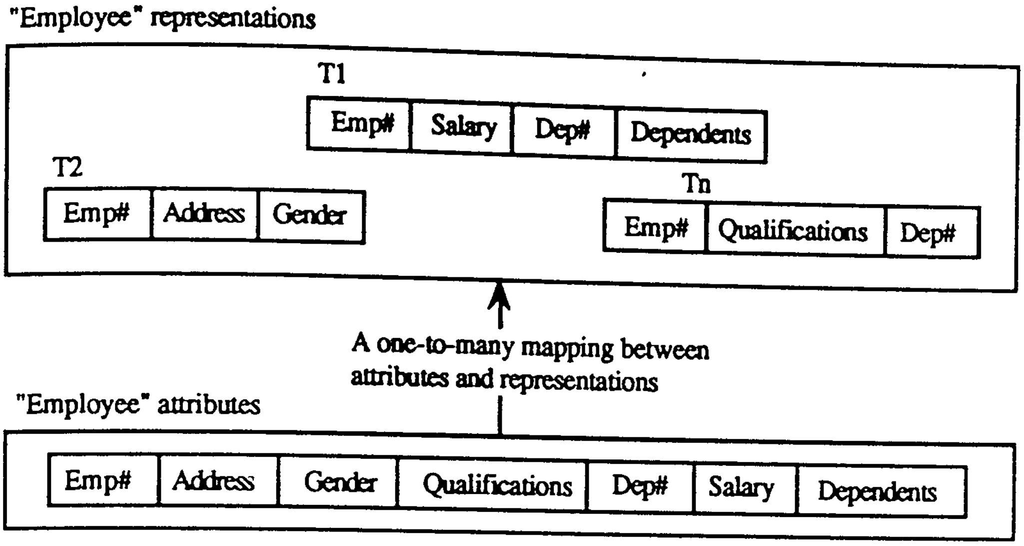


Figure 7.7 An example of a concept multiple representations

The concept of property, of which a representation is composed, is similar - both in terms of definition and function - from that of attribute as specified in the literature. Thus, and to avoid repetition, basic property (or attribute from here on) definition will be drawn from the literature. In this case, Elmasri & Navathe's (1989) introduction to the subject will be associated with the brief clarification of attributes provided in the following analysis.

An attribute draws its value from a value domain (or the value set) which contains all the possible values of that attribute. Thus, an attribute value domain D is specified as a function of the attribute definition over the value set, or

$V(A) = \{v_1, \dots, v_n\}$, where A is the attribute definition, V is the value set, and v1, ... , vn are all possible values of A.

Attributes are either single or multi-valued. The attribute "Age", for example, can only take a single value; whereas the attribute "Qualifications" can include more than one value. In the case of a multi-valued attribute, the constraints on the number of possible values is specified within an attribute definition. Moreover, the previous definition is extended to include all possible combinations of values, i.e., the possible values of an attribute is a subset of the power set of V. Or,

$V(A)= P(V)$, where $P(V)$ is the power set of V

On the other hand, attributes may be a decomposition of more than one value, in which case, the value set is the cartesian product of the power set. In other words,

$V(A)= P(V_1) X ... X P(V_n)$, where $V_1, ... , V_n$ are the value sets of individual values.

To identify a concept, an attribute with a unique value from within the domain of attributes is designated as key attribute. If no attribute possesses such uniqueness, a combination of attributes can be designated as the concept identifier. Each concept representation has to include, as an attribute, a concept identifying attribute(s).

At this stage, the specification of concept structure constituents is complete. What has been established, so far, should enable the formation of a sound context for a precise concept identification for the purpose of comparing and contrasting concepts in preparation for their integration. Moreover, such a context will provide the basis upon which subsequent modelling phases can be carried out; and ultimately, facilitates the creation of an effective conceptual model.

7.3 Meaning-based classification (MBC)

It was previously stated, that prior to the application of SECOM, an initial list of concepts and their attributes has presumably been compiled as an output of requirements definition. Prior to conceptual modelling, design stages are not responsible for pointing out any naming, meaning, or representation differences and/or inconsistencies. What will be addressed in this section is just that, i.e. the identification of semantic concept classes, in the sense that any naming or representation inconsistency is identified resolved.

In the literature, concept classes and classification have not received deserved treatment. Concepts are simply grouped according to their names, or types designated by attributes. Moreover, inconsistencies in concept definitions are resolved within the framework of concept integration. Therefore, integration of concepts or entities is the pointer to any naming inconsistencies, which are then treated with view to having two different concepts represented by the same name, and/or the same concept referred to by different names.

Here, naming inconsistencies is further removed from its usual context, with a view to the fact that the name of a concept holds an *unreliable* relationship with the named object (in the sense that it is context bound). As such, the concept name is inconsequential when it comes to concept integration as will be illustrated in this section. This, however, is not to deny the existence of naming discrepancy, nor is it an attempt to diminish the significance of resolving it. Rather, it is suggesting that because a concept is a mental construct, its name should be seen to play the simple role of pointing out that concept and not providing its meaning. In this sense, concept integration should be perceived independently of its name, or representation for the same matter. This convention is in line with semantic theory and its dictum on concepts and concept naming.

Before examining concept classes and the implication of classification on conceptual modelling, a brief assessment of the concepts of class and classification is appropriate.

Classification is a process in which people engage for the purpose of learning and thinking in general. By classifying objects, physical or otherwise, a person is capable of acquiring new knowledge without having to undergo a complete re-learning process. Howard (1983) explicitly stated the purpose of conceptual classes by asserting that in order to deal with the world's infinite variations, What is implied in the above, is that concepts within the same class can still maintain differences. But further examination will also reveal that what differences in stimuli amount to, are variations (elective or otherwise) in concept perception which - by virtue of its class membership - leaves the meaning of a concept intact.

The question, in the framework of database conceptual modelling, must certainly be this: how would it be possible to consider certain nonidentical concept stimuli are equal, and that such differences should not underlie conditions for concept class membership? From a purely logical point of view (e.g., computer processing), the answer is that it is not possible. In other words, either objects within a class are equivalent or they are not.

A very direct approach to resolving the conflict of classification, as stated above, would be to restrict class definition to the inclusion of only strictly identical concepts. While it is certainly impossible in reality, in a database environment, such solution - if not impossible to implement - defies the object of the database; because of the eventuality of substantial redundancies and inconsistencies.

However, the definition of the relationship of representation between a concept and some of its attributes, provides an effective method of preserving the authenticity of conceptual classes at both database conceptual and implementation levels. More specifically, allowing for the existence of multiple concept representations based on meaning will shift the classification bases from representation to meaning. This method of classification is referred to, here, as *Meaning-based classification* (MBC).

MBC is a method of classification in which concepts' meaning, defined by intension and extension, are the bases for classification. The variety in concept representations and names per se, within a conceptual class under MBC, is irrelevant in deciding class membership. This is saying, that classification is based on the idea in the mind -which has been established as an effective reflection of the object of interaction - and not the word(s) standing for such idea, or the properties elected to represent it, for both have been known and shown to be bias prone.

A conceptual class is therefore said to include concepts of identical meanings. What remains to be resolved is representation and naming inconsistencies within and among members of a conceptual class. To simplify matters, a name will be considered a representation property. This is in accordance with what is already standard practice, i.e., concepts are usually defined and given a name in a system of concepts. In the literature, a class is assigned the same name designating the concept type contained within that class. For example the concept class "Employee" is the class including "Employee" entities. Here, the situation is different. A class, as was previously stated, may include concepts of different names. For example, the concepts "Automobile" and "Vehicle" would belong to the same concept class provided that they have the same meaning, i.e., the naming inconsistency is a result of the difference in terminology. Note, however, that in this example, it is not easy to decide on what reference to assign to the concept the two names are used to refer to, i.e., to the common mental image of the concept. This is simply because any indication of a name would be influenced by personal view and may instigate different concepts for different people. Moreover, an indicated concept name may coincide with a name of a representation within the conceptual class resulting in confusion. This raises the question of the what name to assign to a conceptual class?

An appropriate concept class name would obviously be one designating the idea of the concept classified. But since such name varies according to the terminology used, it can not be effective. But clearly, whatever name maybe assigned to a concept class would prompt various ideas, keeping in mind that the purpose of a name is just that. This may eventually lead to confusion. Therefore, and to make sure that a class name is not a hindrance, non-designating terms will be utilized for class names, i.e., no idea(s) would be conceived because of the name. Under the circumstances, the best non-signifying terms to be utilised are numbers. Numbers have no meaning, and that is exactly what is required in the situation, especially when considering the fact that a class name plays the simple role of identifying a class in the context of MBC. Therefore, a concept class will be identified by a number and a representation by its name. With this in mind, further analysis of MBC can proceed.

MBC involves all four aspects of concept structure (name, intension, extension, and representation) for the purpose of classifying concepts according to meaning, while at the same time identifying naming inconsistencies and representation multiplicity. Naturally, an initial examination of the list of concepts compiled at requirements stage will not reflect relationships between concepts in terms of equivalence of meaning, multiplicity of representation, or naming conflicts. Thus, it will not be possible to initially differentiate between concepts and representations.

Application of MBC entails a straightforward, one-to-one evaluation of concepts' components. The purpose of such evaluation is to establish the similarity in name, intension, extension, and representation among concepts. Similar representations will be identified and linked to common name; and concepts with similar meanings will be grouped under the same conceptual class. The comparison is carried out according to the following criteria.

1) Name (N): Two concepts have the same name if, and only if, their names are literally identical, i.e., irrespective of their reference or designation.

2) Intension (I): A concept intension is identical to that of another concept if, and only if, their intension sets are identical, otherwise the two intensions are different.

3) Extension (E): A concept extension is identical to that of another concept if, and only if, their extension sets are identical, otherwise the two extensions are different.

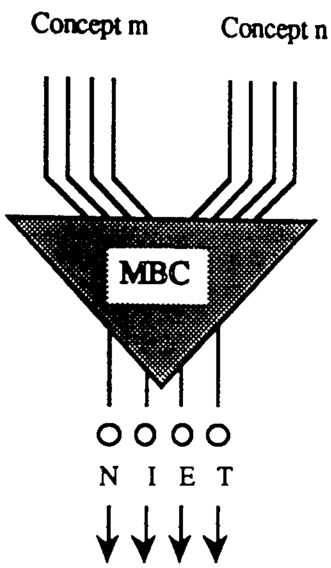
4) Representation (T): Two concepts have identical representations if, and only if, their representations' attributes are identical.

Now that the criteria have been established, the evaluation can schematically be performed. Mathematically, comparing four different components of two concepts results in sixteen different combinations (2^n or $2^4 = 16$), and are all explicitly illustrated in figure 7.8. Note that in the figure, outcomes zero and fifteen are distinguished for reasons which will be explained later.

Each of the outcomes resulting from the above comparison is then put through a filtration process in which each possibility is evaluated for consistency in name-representation, meaning equivalence, multiplicity of representation, and relevance to the classification process. Such process is at the centre of MBC and is depicted in figure 7.9.

It is worth stating, at this point, that there are more than one way of filtering the possible cases of figure 7.8, mainly because the analysis of such cases is dependent on individual concept components, i.e., name in relation representation; and combined components in relation other combined components, i.e., name-representation in relation to intension-extension (or meaning).

The filtration process in figure 7.9 starts by identifying cases in which meaning is different, identical representations have different names, and different representations with identical names (G1). By doing so, the process ensures that each representation has a unique name (i.e., representations of different concepts). The next step is to find the cases in which representations have the same meaning (G2). In other words, identifying conceptual classes. The final test (G3), is to identify the cases in which representation (which are consistent in terms of name because G1 is true for such cases) appear in different conceptual classes. Each test above results in groups of cases. An examination of each of such groups and the cases with each group is necessary in order to come to full understanding of the application of such process.



Possible combinations: $2^n = 2^4 = 16$

(0) ○ ○ ○ ○	4) ○ ○ ● ○	8) ○ ○ ○ ●	12) ○ ○ ● ●
1) ● ○ ○ ○	5) ● ○ ● ○	9) ● ○ ○ ●	13) ● ○ ● ●
2) ○ ● ○ ○	6) ○ ● ● ○	10) ○ ● ○ ●	14) ○ ● ● ●
3) ● ● ○ ○	7) ● ● ● ○	11) ● ● ○ ●	(15) ● ● ● ●

● Different
○ Identical

Figure 7.8 Outcomes of a one-to-one comparisons of concept components:

Name (N), Intension (I), Extension (E), and Representation (T)

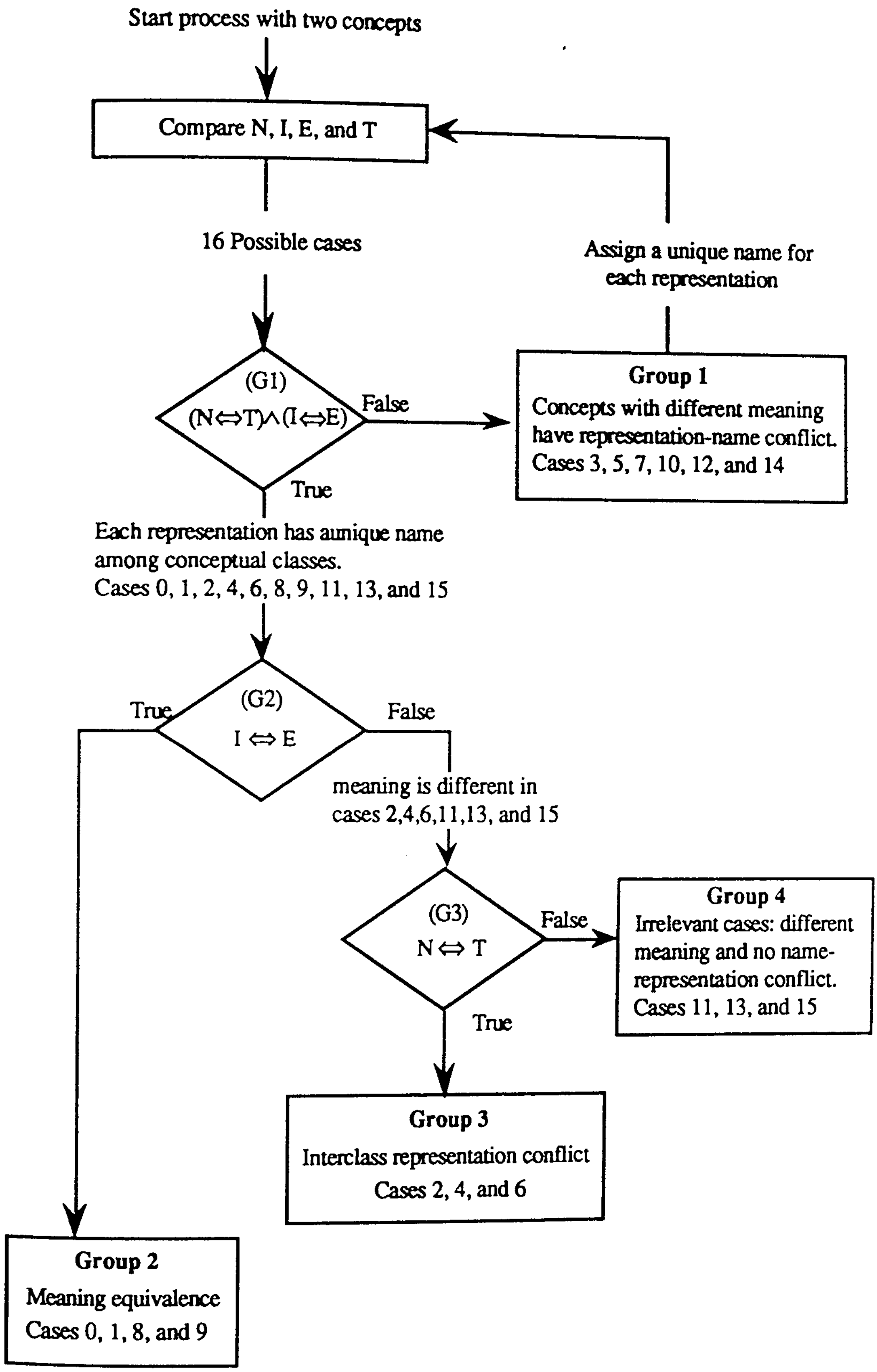


Figure 7.9 Case classification process

7.3.1 Group 1: Name-representation inconsistency

Name-representation inconsistency is the first group of cases resulting from the application of MBC. This group includes all cases in which similar representations have different names, and different representations having the same name. This group is identified by direct examination of the name and representation states (indicated by group 1 in figure 7.9). In other words, two concepts are compared for their name and representation equivalence as specified in the criteria provided earlier. More specifically, for two concepts C_m and C_n,

$$N(C_n) = N(C_m) \wedge T(C_n) \neq T(C_m), \qquad \text{(G1)}$$

If the above condition does not hold, then this is an indication of inconsistency, and it calls for resolution. For example, the two concepts C_m, C_n have the following names and representations:

C_m: "Employee"= {Emp#, Name, Address}
C_n: "Manager" = {Emp#, Name, Address}

In this situation, and before progressing any further, such inconsistency should be resolved. This could be done simply by adopting a name surrogate which is included in the definition of the representation. Cases within this group include 3,5,7,10,12, and 14 (figure 7.10).

Group 1: Name-representation conflict

Case	Name	Intension	Extension	Representation
3	●	●	○	○
5	●	○	●	○
7	●	●	●	○
10	○	●	○	●
12	○	○	●	●
14	○	●	●	●

Figure 7.10 Case parameters for group 1

7.3.2 Group 2: Class inclusion

After resolving naming inconsistencies, the remaining cases undergo the next evaluation process. At this stage, concepts satisfying the condition for the first group are examined for meaning equivalence. In other words, concepts with consistent name-representations are evaluated for equivalence of meaning as specified in the comparison criteria. Keeping in mind that classification is based on meaning, concepts with similar meaning belong to the same concept class. Therefore, this group includes cases which satisfy the following condition:

$$I(C_m) = I(C_n) \wedge E(C_m) = E(C_n) \qquad \text{(G2)}$$

Four cases, out of the remaining ten, satisfy the above condition. The four cases are illustrated in figure 7.11, and are examined separately thereafter.

Group 2: Class inclusion

Case	Name	Intension	Extension	Representation
0	○	○	○	○
1	●	○	○	○
8	○	○	○	●
9	●	○	○	●

Figure 7.11 Case parameters for group 2

Case 0: Absolute coincidence

In this case, name, intension, extension, and representation coincide. This is one of two special cases (the other being case 15) in which the outcome of the comparison is anticipated in advance. Though it is simply integral at this stage, the inclusion of this case in the process is essential in view of the possibility of automating the suggested approach.

Cases 1, 8, and 9: Representation multiplicity

In this case, two concepts with similar meaning have different names, representations, or both; as a result of the differences between views over the same concept. More specifically, a concept may have two similar representations with different names (case 1); two different representations with similar names (case 8); or different names for different representations of that concept (case 9). As a result, the two representations are identified as two representations of the same concept. Consequently, the concept attributes are incremented to include any attributes within the representations that are not already in existing set of attributes.

7.3.3 Group 3: Inter-class conflict

This group includes cases which satisfy condition G1 but not G2. In other words, and as illustrated in figure 7.12, the two concepts' name-representations are consistent, but their meaning varies.

Group 3: Inter-class representation conflict

Case	Name	Intension	Extension	Representation
2	○	●	○	○
4	○	○	●	○
6	○	●	●	○

Figure 7.12 Case parameters for group 3

This group includes the cases 2,4, and 6; and is specified according to the following condition.

$$\{ [N(C_n) = N(C_m) \wedge T((C_n) = T(C_m)] \wedge \neg [I(C_n)= I(C_m) \wedge E(C_n)= E(C_m)] \} \qquad (G3)$$

What this results in is a situation in which two concepts with the same name-representation belonging to two different concept classes. This is inconsistency as recognized in existing data models. It is also the only inconsistency recognized in the literature. Therefore, and to avoid repetition, the cases in this group are treated according to established methods of resolving naming inconsistency. As an example of this group, consider the two concepts C_m and C_n defined as,

$T(C_n)$= {Reg#, Eng-size, Owner}, $M(C_n)$= Ship; and
$T(C_m)$= {{Reg#, Eng-size, Owner}, and $M(C_m)$= Car, then
C_n and C_m are not elements of the same concept class; and thus,
$T(C_n)$ and $T(C_m)$ are a case of inter-class conflict

7.3.4 Group 4: Irrelevant cases

Finally, this group includes cases that do not satisfy any of the conditions stated above (i.e., G1, G2, or G3). By examining such cases (figure 7.13), it is possible to conclude that they have no bearing on the current analysis, i.e., concept classification, simply because such cases have no name-representation conflicts, and have different meanings. This group satisfy the following condition,

$$[N(C_n) \neq N(C_m) \wedge T((C_n) \neq T(C_m)] \wedge \neg [(I(C_n)= I(C_m) \wedge E(C_n)= E(C_m)] \} \quad (G4)$$

Group 4: Irrelevant cases

Case	Name	Intension	Extension	Representation
11	●	●	○	●
13	●	○	●	●
15	●	●	●	●

Figure 7.13 Case parameters for group 4

This group includes cases 11, 13, and 15. Note, here, that case 15 is the other special case in which the outcome was predicted. Nonetheless, it is included here for the reasons stated under case 0. Note, also, that irrelevance in the context of classification, i.e., G4, has no implication on further analysis of concepts for semantic relations.

7.4 Semantic relations (SRs)

In the previous section, conceptual classes were semantically formed and naming-representation inconsistency resolved. The next step is to link conceptual classes according to meaning so as to form an integrated system of concepts. Such a system will itself, thereafter, be the foundation upon which further relationships (the subject of chapters 8 and 9) are defined. For now, however, application and organization structure independent semantic relations between conceptual classes will be defined. Such relations will be referred to as *semantic relations,* or SRs for brief.

There are a number of suggestions in the literature regarding the definition of semantic relations, a number of which were presented within the semantic models discussed in the literature review. By reviewing such models, a conclusion can be drawn to the effect that a concise definition of SRs and their application in database design within an established semantic context, as one in which SECOM is presented, is lacking. However, there are promising approaches which, upon extending, can provide a sound framework for defining semantic relations. In this work, the approach suggested by Kauppi (1967) (and re-introduced by Kangassalo (1982)), will be extended and refined so that it is consistent with the framework of the suggested approach.

Before introducing SRs, it is important to mention that semantic relations are established among concepts at a conceptual class level. In this sense, a concept name and/or representation is not an issue in semantic relations. Rather, it is meaning, elucidated by concept intension and extension, that is the subject of SRs. The following semantic relations are based on the semantic foundations and definitions presented in chapter 5.

7.4.1 Semantic independence (SR1)

Semantic independence is an indication of the complete detachment of the intensions of two concepts. This relation is defined as follows.

Definition 7.2 Two concepts are semantically independent (i.e., have an <SR1> relation) if, and only if, they have no properties in common. In other words,

C_n <SR1> C_m, Iff
$I(C_n) \cap I(C_m) = \emptyset$

The establishment of semantic independence eliminates the possibility of instance sharing between concepts. Therefore, two semantically independent concepts have no update consequences on one another. This relation has an additional use in defining another semantic relation which will be presented subsequently.

7.4.2 Semantic inclusion (SR2)

The relation of semantic inclusion holds between two concepts when the meaning of one is included in the meaning of the other. In this relation, the more general concept, in terms of meaning, is referred to as the definer; while the specific one is referred to as the refiner. This relation is defined as follows.

Definition 7.3 The relation of semantic inclusion (<SR2>)is defined over two concepts, C_n and C_m if and only if, all the properties of one concept are included in the other. Or,

C_n < SR2> C_m Iff
$I(C_n) \subset I(C_m)$
Accordingly, C_n defines C_m, and C_m refines C_n

The definition of SR2 implies that the larger the concept intension the more specific its meaning is. Note that in the definition, the identification of common intension elements is only necessary for SR2 to hold between two concepts. The satisfaction of the relation condition is dependent upon the detection of common extension components. Such identification implies the presence of attribute sharing among the two concepts. Therefore, only when the intersection of the extensions of the two concepts involved is non-empty that SR2 holds. To illustrate an SR2 relation, the following example, which is also depicted in figure 7.14, is offered.

I(Person)= {Person}
I(Employee)= {Persons on the organization's pay-roll}
I(Engineer)= {Person whose an employee and in an engineer position}, then
I(Person) $\subset$ I(Employee) $\subset$ I(Engineer), and thus
(Person <SR2> Employee), and
(Employee <SR2> Engineer); and by the same token
(Person <SR2> Engineer)

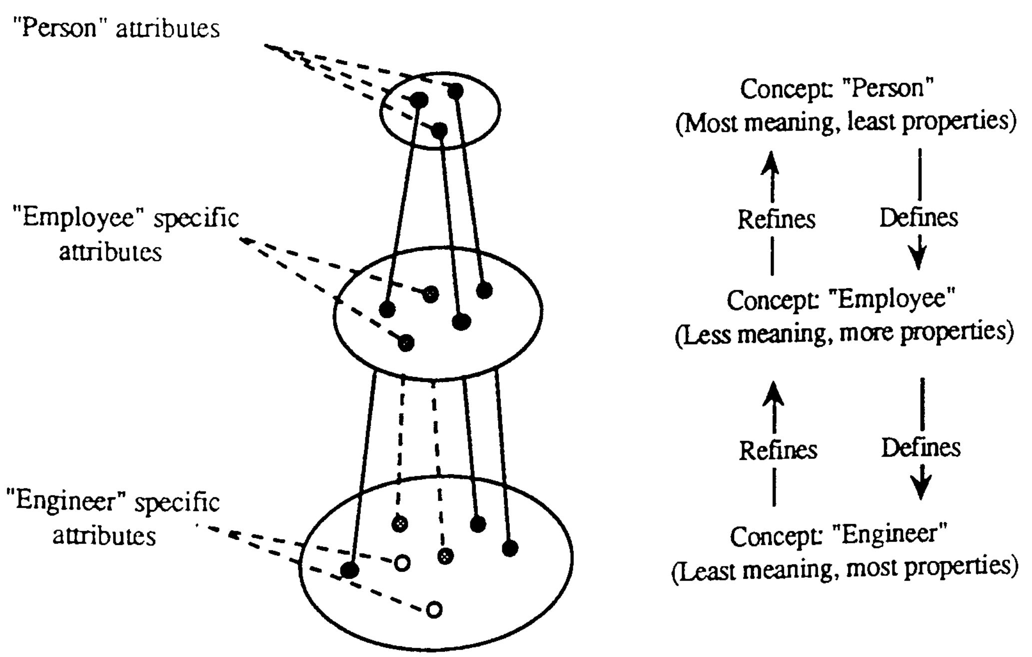


Figure 7.14 The relation of semantic inclusion (SR2)

The semantic relation of meaning definition and refinement is an extension to the acknowledged *ISA* relation identified in the literature. What SR2 offers in addition to a traditional *ISA* relation is that it enables multiple representations of a concept to hold an *ISA* relation with other representations of another concept. This is simply saying that because the multiplicity of concept representation is recognised in SECOM, an *ISA* relation should not be installed among representations, as it is the case in the literature, but rather, it should be devised between concepts if consistency is to be maintained.

7.4.3 Semantic intersection (SR3)

The relation of semantic intersection is established when two concept have some properties in common, no concept includes the other, and both have a common definer. The identification of a common definer in this case is used to indicate property sharing. Semantic intersection is defined as follows.

Definition 7.4 Two concepts, C_m and C_n, semantically intersect (or have an <SR3> relation), if, and only if, the intension intersection of the two concepts is a non-empty proper subset of each of the concepts' intensions, and at least one of the concepts extends over the other. Or,

C_n <SR3> C_m Iff
$I(C_n) \cap I(C_m) = D$, where D is a set such that
$D \subset I(C_n)$, $D \subset I(C_m)$, and $D \neq \emptyset$

Note that the set D has to be *properly* included (i.e., a proper subset) in both concepts to differentiate this relation from that of semantic inclusion.

Though not sufficient, this relation is necessary for the explicit indication of instance sharing among the concepts involved. Thus, an SR3 is intended to maintain database integrity by ensuring proper updates.

There are, however, different specifications of this relation in that two concepts may have different possible combinations of extensions. The first two being that either of the two concepts extends over the other, and the third is when both concepts extend over each other. This is illustrated in the following example which is depicted in figure 7.15.

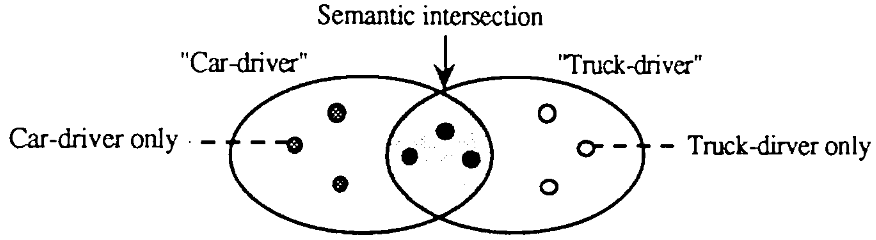


Figure 7.15 The relation of semantic intersection (SR3)

The two concepts “Car-driver” and “Truck-driver” have properties in common, i.e., (Person <SR2> Car-driver) and (Person <SR2> Truck-driver). If the meaning of the two concepts were to be defined as:

I(Car-driver)= {Person who is licensed to drive a car}
E(Car-driver)= (Car-driver, Truck-driver), and
I(Truck-driver)= {Person who is licensed to drive a truck}
E(Truck-driver)= (Truck-driver), then
(Car-driver <SR3> Truck-driver)

However, and since there is a difference between a car-driver being a truck-driver and the converse, there will be a distinction made to indicate the direction of the semantic intersection. Therefore, the notion <SR3> will be used whenever the intersection is mutual; and doubling the pointer in the notation to indicate the direction otherwise. Thus for the example above, the relation is indicated as (Car-driver <SR3>> Truck-driver) to indicate that "Car-driver" extends over "Truck-driver", but not the opposite.

Other cases include (Student <<SR> Supervisor), if a "Supervisor" can also be a "Student" but not the converse; and (Manager <SR3> Engineer), if "Engineer" can be a "Manager" and vice versa. Diagrammatically, an arrow pointing in either or both directions is used to illustrate the above distinction.

7.4.4 Prevalent convergence (SR31)

This relation is identified when a concept has more than one definer, and the definers semantically intersect (i.e., have an <SR3> between them). This relation is an indication of concept specialization. More specifically, two semantically intersecting concepts participate in formulating the meaning of a third concept. The reference *Prevalent* in the name of this relation is an indication of the fact that the convergence is customary by virtue of the close meaning relationship between the converging concepts. Prevalent convergence is defined as follows.

Definition 7.5 The relation of prevalent convergence is identified when two concepts, C_m and C_n, semantically intersect (i.e., C_m <SR3> C_n), and there is a third concept C_k which includes the meaning of both C_m and C_n. In such case, the relation is defined between the new concept and each of its definers. In other words,

If C_m <SR3> C_n, and
(C_m <SR2> $C_k \wedge C_n$ <SR2> C_k), then
C_m <SR31> C_k and C_n <SR31> C_k

The definition above implies that the relation of definition (SR2) is replaced by the intersection convergence relation. There are two reasons for replacing an SR2 with an SR31. The first, and the less important of the two, is so as to point out the presence of a specialization abstraction. Though such indication does not have a direct impact on the actual modelling process, it is outlined here in case concept abstraction identification is performed in a bottom-up fashion (see section 3.2); in which case it is necessary to indicate the definers of a concept at the time it is encountered. The other reason for substituting an SR2 with an SR31 is the need for predicating a specific relation between two concepts participating in defining a third one. This will be explained further under the third aspect of SECOM, i.e., chapter 9.

The relation of prevalent convergence can involve more than two concepts as specialization definers. In which case, the same considerations are applicable.

In terms of updates, the concepts involved in prevalent convergence follow similar update rules as those of semantic inclusion. In other words, changes in any of the defining concepts result in similar changes in the same instances (if any) of the co-definer and the specialization. On the other hand, an update in the specialization has a direct update consequence on both definers. The relation of intersecting convergence is illustrated in the following example.

Let "Engineering-manager" be a concept defined by "Engineer", i.e., (Engineer <SR2> Engineering-manager). Assume, in addition, that "Engineering-manager" is also defined by "Manager", i.e., (Manager <SR2> Engineering-manager). Therefore, there is a convergence in the concepts "Engineer" and "Manager". Since "Manager" intersects with "Engineer", or (Manager <SR3> Engineer). Accordingly, the relation of intersecting convergence is identified between the specialization "Engineering-manager" and each of its definers, or (Engineer <SR31> Engineering-manager) and (Manager <SR31> Engineering-manager). This example is depicted in figure 7.16.

The concept "Motor-home" is a good example of this relation. While it includes in its meaning, i.e., is a "Vehicle", it is also a "Residence". Thus, each of "Residence" and "Vehicle" are definers of "Motor-home", or (Residence <SR11> Motor-home) and (Vehicle <SR11> Motor-home). Another example, which though less likely, is possible, is depicted by figure 7.17.

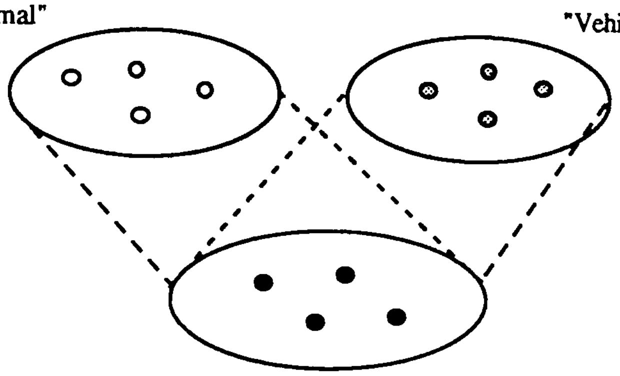


Figure 7.17 The relation of atypical convergence (SR11)

7.5 Relation identification process

SR11's presentation renders the system of semantic relations complete. However, it should be stated here, that defining semantic relations among concepts has to be performed in a specific order of execution if an optimum identification of such relations is to be secured. While the order of presenting such relations was given to aid comprehension, actual implementation would benefit from a slight re-ordering.

It seems logical to begin by ascertaining the existence of semantic connection between two concepts. The absence of such actuality would establish semantic independence (SR1). The next, and final step in this case, is to directly examine the presence of atypical convergence (SR11) and terminate the examination of the two concepts.

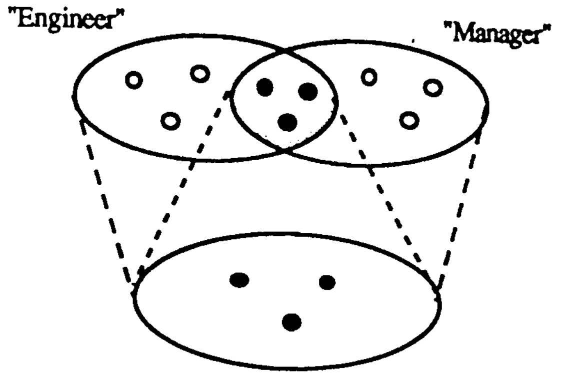


Figure 7.16 The relation of prevalent semantic convergence (SR31)

7.4.5 Atypical convergence (SR11)

Whereas SR31 is defined as the convergence of two intersecting concepts, SR11 is defined as the convergence of two semantically independent concepts. This relation holds between two concepts by virtue of their incorporation into a concept which otherwise would not have not been perceived as *usual*. SR11 is defined as follows.

Definition 7.6 Two concepts, C_m and C_n, have an atypical convergence relation between them if the two concepts are semantically independent, and there is a third concept C_k which includes the meaning of C_n and C_m. In other words,

If C_m <SR1> C_n, and
(C_m <SR2> $C_k \wedge C_n$ <SR2> C_k), then
C_m <SR11> C_k and C_n <SR11> C_k

Although this relation exists in reality, it is less frequently encountered than an SR31. This is because it is uncommon to find a concept which includes in its meaning the meanings of two (or more) different concepts. Nonetheless, and because of the differences in applying the two versions of convergence (as will subsequently be demonstrated), it was necessary to examine the two relations separately.

Missing page(s) from the bound copy

Pages 154

On the other hand, the omission of semantic independence would systematically indicate the presence of semantic intersection. However, such identification needs further examination to ascertain the degree of overlap in the meaning of the two concepts. Here, the intersection itself is examined to find out whether or not it is equal to either of the two concepts' meaning. If it is, then the relation of *semantic inclusion* (SR2) is established, and the process is terminated. Otherwise, the relation of intersection is still the default. This leaves one final examination to be carried out, and it is that of semantic convergence. If the two concepts converge to form the meaning of a third concept then the relation of *prevalent convergence* (SR31) is established; otherwise, the relation of *semantic intersection* (SR3) is passed. Figure 7.18 is provided to illustrate the schematic approach to establishing semantic relations.

7.6 Semantic abstractions

The benefits of devising a purely semantic approach to designing conceptual models can be seen in the application of such methods to the formation of conceptual abstractions. Conceptual abstractions, according to the proposed approach, have advantages over the conventional view of data abstractions as presented in the literature.

It is widely acknowledged in the literature, and has been said in this work, that user perceptions of data varies according to personal perspective. It is also recognized that such variety in perception usually calls for the utilisation of complicated solutions when it is determined, and is a cause of concern over inconsistency when it isn't. In other words, conflicting views are not always readily ascertained, and as such, the obscure capacity in which they may exist poses a threat, to the database integrity and consistency, which may only be realized when such conflicts materialize at further stages, e.g., when creating new concepts or deleting existing ones.

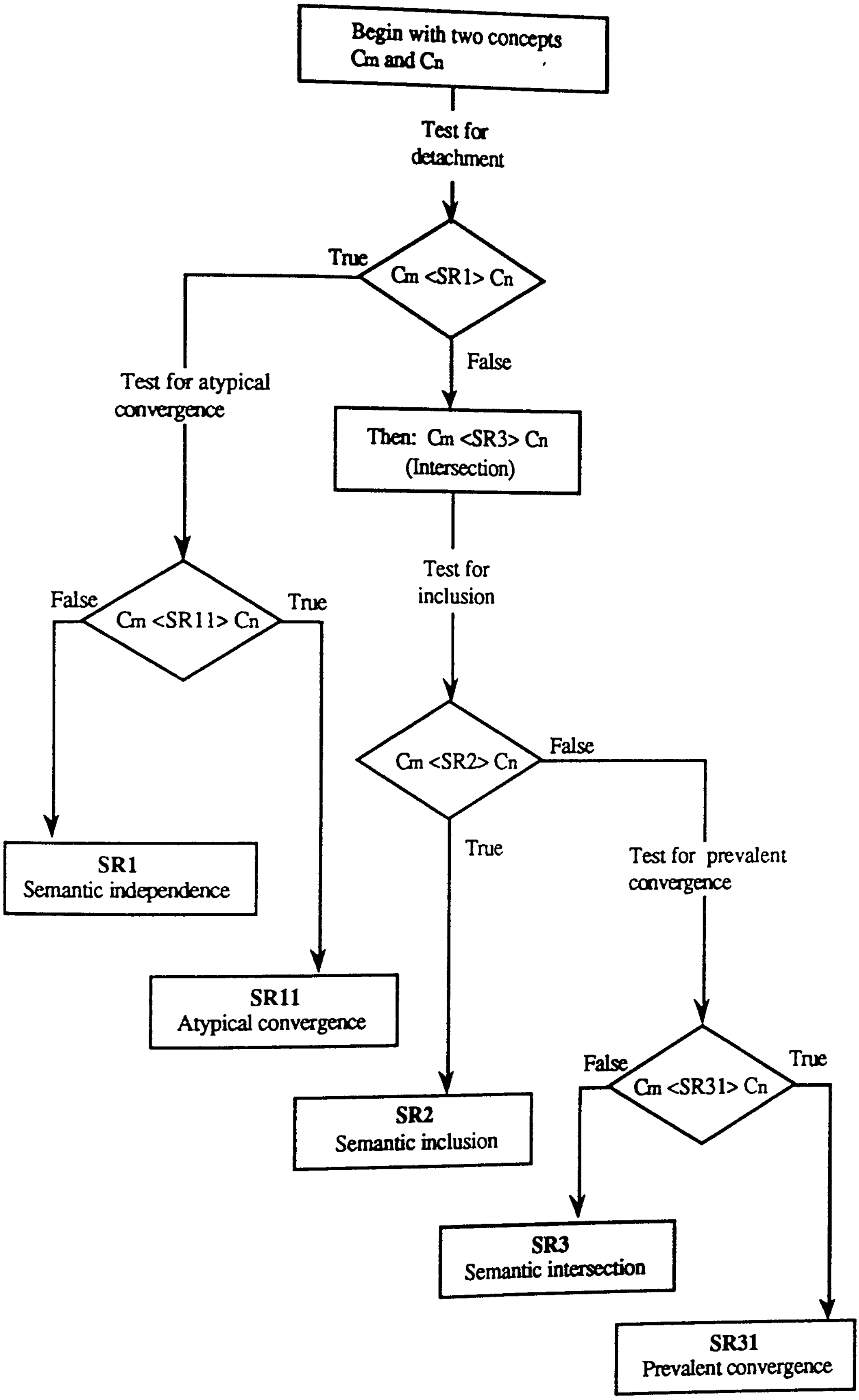


Figure 7.18 The process of identifying semantic relations between concepts

The solution offered in this work is based on the basic convention that no two users can have conflicting views over data when its meaning has been established in advance. With this in mind, it is hard to conceive of a situation in which different representations of different concepts can contradict or indiscernibly overlap. While the MBC method provides an inconsistency-free context, the semantic relations offer a practical method of grouping conceptual classes.

An example drawn from Elmasri & Navathe (1989) is reproduced below (figure 7.19), to give an illustration of the strength of MBC and SRs in the framework of "Advanced data modelling concepts". This example is particularly chosen because it projects the general direction of, and approach to data abstraction in the literature.

The example in the figure displays an abstraction hierarchy for a university database in which generalization and specialization abstractions of entities are depicted according to the *Enhanced Entity-Relationship approach* (EER) (Elmasri & Navathe, 1989). Although the connections between the various entities in the figure were presented in chapter 3, a brief re-examination of such connection is necessary in this context.

According to the EER, entities are classified according to their type. An abstract entity type has a super-class and sub-class entities in which a sub-class has an *IS-A* relation with a super-class. Participation of sub-classes in a super-class is said to be either total or partial depending on whether all entities in the super-class have to participate in the sub-class or otherwise. In figure 7.19, for example, "Grad-student" and "Undergrad-student" entities have a total participation in "Student" entity, indicated by the double line drawn form "Student" to each of the two entities. Overlap and disjoint in participation is indicated by the circled litters "O" and "d", respectively. This results in the possibility of four relationships existing between entity classes, namely

Figure 7.19 An abstraction hierarchy example (Elmasri & Navathe, 1989)

disjoint-total, disjoint-partial, overlapping-total, or overlapping-partial. The subset symbol in the figure indicates the direction of the sub-class/super-class relationship, e.g., "Staff" has a sub-class relationship with the super-class "Employee". Finally, sub-classes can participate in more than one super-class to form a specialization lattice, e.g., "Employee" and "Student" are the components of the specialization super-class "Student-assistant". If this example were to be reconsidered from the *semantic aspect* of SECOM, the abstraction hierarchy depicted in figure 7.20, would emerge.

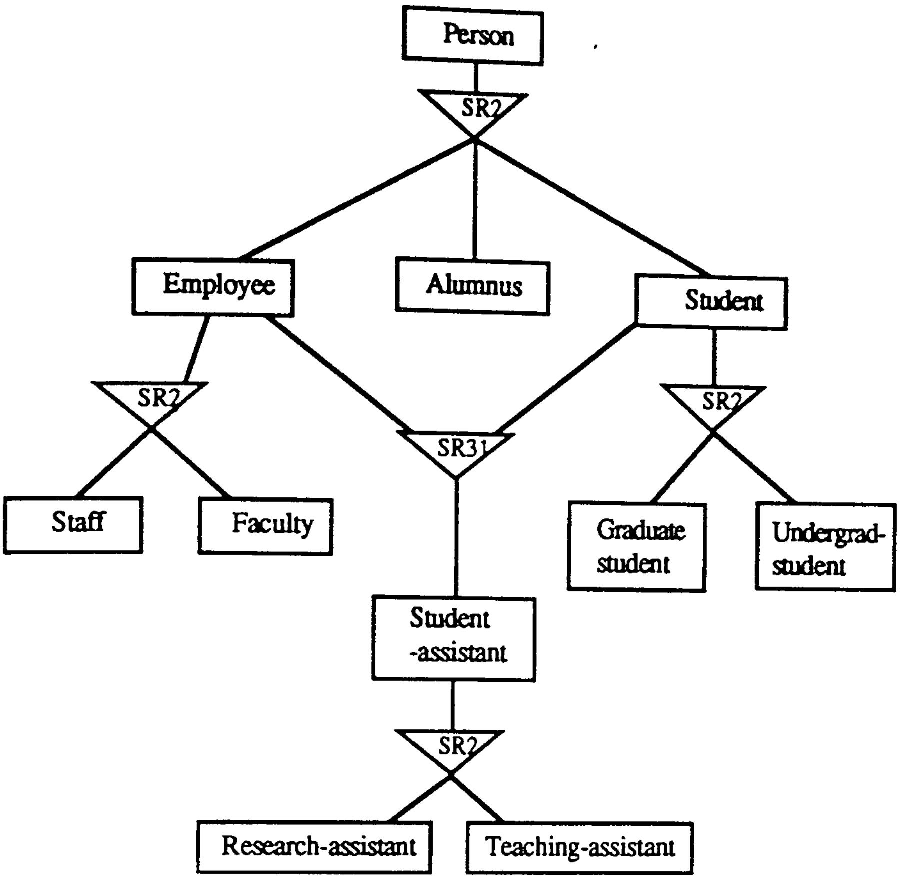


Figure 7.20 The SECOM approach to conceptual abstractions

The entity "Person" defines (i.e., has an SR2 relation with) the entities "Employee", "Alumnus", and "Student". This is explicitly indicated in figure 7.20. What this implies, is that the attributes of person form a proper sub-set of all of its refined entities' attributes.

There are two different conditions on the participation of "Person" sub-class concepts as shown in the EER diagram. The first is that of disjoint, and the second is total participation. In SECOM, however, indication of such constraints is provided within the semantic definition of concepts. More specifically, overlap and disjoint are intrinsically illustrated by including all such entities to which the concept "Person" applies within the extension of "Person" (i.e., E(Person)= {Employee, Alumnus, Student}). On the other hand, total and partial participation is a property of "Person", as such it is included in its extension (i.e., E(Person)= {Person, Employee, Alumnus, Student}). This is a direct and logical indication of the fact that a "Person" has to be one of the three refined concepts. There is also the additional advantage of indicating the direction of concept overlap. Therefore, it is possible to explicitly state that an "Employee", for example, can be a "Student" but not the opposite; simply by including "Employee" in the extension of "Student", but not the opposite. This extends over all possibilities of overlap.

There is another semantic relation involved in the example, and it is that of prevalent convergence (SR31) between "Employee" and "Student" over "Student-assistant". The implication, here, is that the meaning of "Student-assistant" is made up of both of its meaning contributors. In this case, the attributes of "Student-assistant" are drawn from both "Employee" and "Student", and attribute update follows accordingly.

The example above demonstrates the advantages of SECOM utilization over many suggested approaches exemplified by the EER. However, the example is a relatively simple one. Deserved appreciation of SECOM application would clearly emerge in more elaborate situations, and upon completing its constructs.

In addition, the utilities provided in the semantic aspect of the proposed approach are capable of the systematic identification and manifestation of abstraction hierarchies in situations in which their presence is too obscure to be maintained. More explicitly, the reliance upon meaning, instead of representation, in abstraction maintenance, practically translates into the formation of abstractions that seem less obvious from a representation perspective. This in turn, removes of the responsibility of building effective conceptual abstractions from relying upon designer modelling experience, and substantially downgrades the significance of knowledge of the modelling environment.

Another advantage of SECOM semantic aspect, is that more than one view of the same conceptual abstraction can be had. This is a direct consequence to representation multiplicity provision. It is therefore permissible for the same abstraction hierarchy to be seen differently by different individuals, and according to their own views of the concepts involved. This view of abstraction gives a third dimension to the conventional vertical and horizontal depths of abstraction hierarchies. The third dimension being the link between concepts and their representation. This is explicitly demonstrated in figure 7.21, in which relationships between representations are implied by meaning abstractions, which are in turn illustrated in a different dimension.

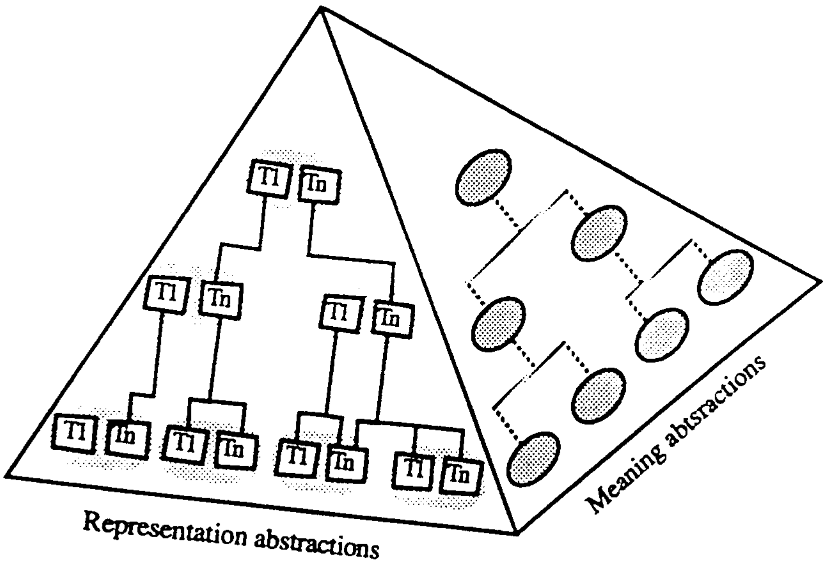


Figure 7.21 The three dimensional perspective of conceptual abstractions

7.7 Conclusion

In this chapter, the first modelling level was established. The semantic aspect of SECOM presented the basic foundations upon which consequent modelling constructs will be based. Such foundations, as was stated earlier, are independent of database specifications. In other words, they have no bearing on concept relationships imposed by application and/or organization structure. Some advantages of semantic application to conceptual modelling were pointed out in this chapter. However, a complete appreciation of the effectiveness and practicality of the proposed approach will only be realised at the stage in which an integral modelling system is put forward.

Chapter 8

SECOM II: The application perspective

In the previous chapter, basic conceptual foundations were introduced. A system of concept definitions, classification, and integration was constructed based on established semantic theory. Such system offers sound and stable foundations upon which further concept relationships can be defined. A category of such relationships is one in which concepts are related according to their usage in the database application environment. This category of relationships is the subject of this chapter. Therefore, the second perspective of SECOM will delineate the definitions and rules pertaining to the utilization of concept relationships and the application of such relationships in conceptual modelling.

8.1 The semantics of relationships

A relationship is a proposition linking two (or more) objects based on the satisfaction of specific conditions reflected by specific object properties. However, there is more to a relationship than a simple link. A relationship is a concept with its own distinct meaning proffered by the relationship intension and extension. In classifying concepts, Bunge (1967) defined a relation as *a concept* which applies to individual and sets of objects. This is hardly unforeseen considering the significance of the task of distinguishing concepts from relationships in earlier design phases; which in itself is an indication of the vague difference between concepts and relationships.

Because a relationship, in a semantic sense, is a concept proper, it is necessary to examine whether or not such view warrants treating relationships as concepts within the same conceptual framework presented in the previous chapter.

The conceptual framework introduced in chapter 7 placed extensive emphasis on concept meaning definition portrayed by intension and extension. The main purpose of outlining the meaning of a concept was said to be the establishment of firm basis for concept classification and integration.

When considering a relationship as a concept, similar implications regarding the relationship meaning prevail. In other words, the meaning of a relationship should be established in order to arrive at sound relationship constructs. However, and because intension and extension, as defined in previous chapters, are purely semantic, the application of such concepts in a modelling framework cannot be expected to be systematic. Strict application of intension and extension concepts to relationships may even be inappropriate.

To begin with, a concept intension is meant to point out its intrinsic properties, while its extension indicates its field of application. The definitions of intension and extension are then utilized within a predefined context to provide a frame of reference for classification and integration. The definition of concept attributes and update is identified with, and linked to intension and extension sets. in addition, concept representation is very closely related to its intension. In other words, the data collected about a specific concept is a set of properties which are conceptually inherent. Therefore, and by definition, such data constitutes a sub-set of the concept intension.

The situation is different in the case of relationships. To begin with, to a considerable extent, relationship meaning is implied by indicating participating concepts. This significantly reduces the gap between the relationship and its meaning. For example, the relationship "Own" can be, relatively, moderately understood, but only to a certain extent. However, when it is stated that "Own" holds between the concepts of "Person" and "House", its meaning emerges almost immediately. This raises doubts as to the appropriateness (and not validity) of involving the additional constructs of intension and extension in the definition of a relationship; because such action would hinder rather than benefit the modelling approach at hand. But before drawing a certain conclusion to that effect, it is worth exploring aspects of relationships within the context of concept semantic definition.

In conceptual modelling, the main interest in a relationship is not in the notion carried out by the relationship itself, but is rather in the relationship facility of collecting additional and unique data about participating concepts. In other words, the creation of a concept out of existing ones, which has been used at times to describe a relationship, is not the main interest here; but it is the additional attributes which concepts gained by virtue of their participation in the relationship that is of concern to conceptual modelling. This is clearly evident in the case where the same relationship holds between a number of conceptual classes. For example, the relationship "Own" previously identified as holding between "Person" and "House" can also be defined over the classes "Company" and "Building", "Company" and "Bank-account", "Bank" and "Company", etc. What this reveals is the diminished emphasis on the meaning of *ownership*, and the importance of the attributes which the relationship adds to the modelling situation.

Finally, relationships, within a database context, acquire additional properties pertaining to the management of concept participation in a relationship. At the forefront of such properties is the cardinality of a relationship. This property explicitly indicates the rules governing the number of concept instances allowed to participate in the relationship. Only a single instance of the concept "Employee", for example, participates in the relationship "Manage" between "Employee" and "Project" when the relationship cardinality is one-to-one. There are, however, other important properties including relationship direction, degree, and dependency. According to Kent (1977), the possible number of combinations of relationship properties is enormous.

> "Even with [the provided] list of characteristics, we already have 432 forms...This number might include some symmetries, duplicates, and meaningless combinations, but after subtracting these we still have a sizeable checklist."

The point to make, here, is that the meaning of a relationship is drastically affected by the properties attached to each relating situation. The same relationship between two concepts can have different meanings based on the rules governing concept participation in such relationship. After all, such rules are properties of the relationship, and their composition, permutation, or annulment can establish, modify, or completely change the meaning of the relationship. To demonstrate this point, consider the example of the relationship "Marriage" defined over "Person" concept. This relationship can have as many meanings as properties attached to the relationship. In the case of the single property of cardinality, a one-to-one cardinality reflects *conventional* marriage; whereas a one-to-many would give it a different meaning. On the other hand, lack of constraints over a participant gender would complicate things even further.

Therefore, and after considering the issues affecting the meaning of the concept of relationship, the following conclusion can be drawn. A relationship is a concept. It has its own distinct meaning. Therefore, it has, as concepts do, an intension and an extension. However, and because the identification of a concept meaning, and in particular its intension, was utilized for the purpose of concept integration, a relationship meaning cannot appropriately be devised to serve the same purpose. This is simply because of the degree of variety in a relationship intension - portrayed earlier by the possibilities of property variance even within the same relationship class - which would accordingly result in overwhelming the design process by elaborating its execution. Because the dividend gained form additional intricacies can be undermined by potential complexity, meaning will not be utilized to define a relationship. Instead, form, or extension, will be used as the sole indicator of the meaning of a relationship.

In a sense, abandoning intension as a component of meaning in relationship definition, is not unconventional in scientific theory. For example, a sovereign paradigm in semantics exists on the assumption that form is meaning. *Extensionalism*, as it is referred to in semantics, is a branch of the science which relies on structure more than description, sense, or intension as conveyors of concept meaning.

This prospective, as it is clear from the previous chapter, was not adopted in the pursuit of an underlying mechanism for concept definition and integrating. Here, however, the situation is different.

In the previous chapter, concept definition was completely constructed from basic linguistic constructs, and with complete reliance upon user perception of reality. It is critical when linguistic constructs, i.e., concept names, are used to describe reality to rely on content rather than form to ensure effectiveness of semantic modelling. The strength of the whole approach of the first perspective of SECOM is derived from this convention. Such convention separates the proposed approach from most other modelling approaches which rely on the structure of an entity as grounds for set, class, or type inclusion, and integration; an approach typified by Elmasri et *al.* (1985) in which it is stated that:

> "Entities are classified into types according to their basic attributes."

Relationships, however, and in particular in a database modelling environment, by definition exhibit formalization shifted from that of concepts. The most relevant, under the circumstances, is that relationships are basically mathematical operations over concept classes. Their contextual definition in data modelling is weak. Interest in a relationship is in its linking property and not its meaning content. This, in addition to relationship property considerations stated earlier, makes subscribing to an extensionalist approach to relationship management, imperative, if not inevitable.

Accordingly, the remainder of this chapter will offer analysis of intrinsic conceptual definitions and properties of relationships. Such analysis will create an environment in which relationships will be utilized within the framework of the proposed approach.

8.2 Relationship definition

A relationship is an association between two concepts. Its meaning is indicated by the relationship extension. Formally, a relationship is defined as follows:

Definition 8.1 A relationship R is a set of associated concepts defined over a number of concept classes. The set R includes all individual concepts which satisfy the predicate of the relationship.

Defining a relationship over a number of concepts refers to the property of relationship degree. The degree of a relationship is the number of concept classes participating in the relationship (by definition, a relationship degree is always more than one). For example, the relationship of "Employ" between the concepts "Company" and "Employee" is of degree two, or binary. The relationship "Delivery", on the other hand, between the concepts "Supplier", "Parts", and "Warehouse" is of degree three, or trinary.

To avoid complexity, only binary relationships are considered in this work. Making provision for higher degree relationships is recognized as a matter of extending binary relationship constructs.

The specific definition of a relationship over specific concepts under specific constraints renders their management a delicate issue bound, and imposed upon by such specificity. The relative generality of the approach according to which concepts were defined, classified, and integrated in the previous chapter, does not apply to relationships. Whereas concept properties are mainly intrinsic in nature and are, thus, part of concept definition, relationship properties are less relevant to their inherent significance, and are rather assigned according to the participating concepts.

Because of the basic modelling requirement pertaining to model implementation independence, it is important to explicitly state all relationship properties as part of their definition and at the modelling stage, and not as logical constraints at implementation phases. Therefore, relationship properties are presented in the following sections, with emphases on explicit indication of such properties.

8.2.1 Relationship extension

Extension, having been established as the relationship defining notion, is going to be utilized mainly for that purpose. The extension of a relationship is identical, in definition, to that of a concept. However, similarities between the two extensions cease beyond their definition. A relationship extension is further extended to compensate intension, with additional property inclusion.

The definition of a set relation reference was provided in chapter 5 in the context of semantic theory (corollary 5.3). The same definition is re-molded and utilized, here, for the purpose of distinctly identifying relationship sets. Additional provisions and/or specifications will be formalized and introduced as part of a relationship extended extension (or extension from here on).

Definition 8.2 The extension of a relationship R is the set of paired elements of the extensions of the relationship constituents. In other words,

If $c_m \in E(C_m)$, $c_n \in E(C_n)$, and
R is a set defined as C_mRC_n, then
$c_mRc_n \in E(R)$

A relationship may be defined over a number of concept classes, in which case each class of the relationship extension is indicated separately as part of the relationship extension.

Definition 8.3 The extension of the set R is defined as the set of extensions of the relationship constituents. In other words,

If C_iRC_j , C_kRC_l, ..., C_mRC_n, then
$E(R)= \{E(C_iRC_j), E(C_kRC_l), ..., E(C_nRC_m)\}$

For example, the relationship "Owns" between the concepts "Person" and "Vehicle", and "Company" and "Building" is defined as:

E(Owns)= {(Employee, Vehicle), (Company, Building)}

The extension of a relationship will be further developed as additional concepts are introduced in the scope of this chapter.

8.2.2 Structural properties

Paramount to effective definition and implementation of relationships is the explicit and comprehensive inclusion of their structural properties. A relationship property refers to the constraints imposed by users and applications on the way concepts participate in the relationship, to ensure the correctness and cohesion of the conceptual framework.

Relationship properties are defined on a number of levels and in various forms. Because of such variety, and the need for their explicit specification, structural properties, unavoidably, contribute to the extension of the level of detail and specificity of a modelling construct. The inevitability of such explicit specification stems from the reality that any alternative to doing so would lead to the separation of relationship properties from its definition. This could lead to managing properties at the implementation level, an outcome which contradicts basic data independence requirements.

In the following, structural properties of relationships, as defined in the context of SECOM, are introduced.

8.2.2.1 Participation

The extension of a relationship is specified by the concepts participating in the relationship. As indicated earlier, a relationship can hold over a number of concepts as illustrated in the following:

$E(R)= \{(C_m, C_n)\}$

A relationship can be defined over the same concept. For example, "Company" has the relationship "Own" over itself. The extension of the relationship "Own" is then,

E(Own)= {Company, Company}

More traditionally, a relationship holds between two different concepts. For example, the relationship "Drive" is expressed as extending over the concepts "Driver" and "Vehicle". Accordingly, the relationship is defined as:

E(Drive)= {Driver, Vehicle}

The same relationship can be defined over a number of different pairs of concept classes, such as "Trucker" and "Truck", "Employee" and "Car", etc. In which case, the extension of the relationship becomes:

E(Drives)= {(Driver, Vehicle), (Trucker, Truck), (Employee, Car)}

Furthermore, more than one concept may participate in either side of the relationship. In this case the extension of the relationship becomes:

$E(R_i)= \{C_k|C_l, C_m|C_n\}$

For example, both "Engineer" and "Supervisor" can participate in the relationship "Manage" over the concept "Project". The relationship "Manage" is then illustrated as:

E(Manage)= {Engineer|Supervisor, Project}

This indicates that the relationship holds between the classes "Engineer" and "Supervisor", on the one hand, and "Project" on the other. By the same token, relationships may hold among any number of concept classes on either side of the relationship.

Moreover, a relationship may be defined over the same set of concept classes. For example, the relationship "Assist" over the concepts "Scientist", "Engineer", and "Technician" may be defined as:

E(Assist)= {Scientist|Engineer|Technician, Scientist|Engineer|Technician}

The above is a manifestation of the basis underlying the approach to specifying a relationship property using the category concept (Elmasri et *al*, 1985). In relationship categorisation, an additional construct is distinctly designated for representing a group of concepts within a category participating in a relationship. For example, the categories of "Owner" and "Vehicle" are defined for the relationship of "Ownership"

In the same way, a relationship may hold between the same set of concept classes. Where the relationship among the concepts holds under the same conditions, the definition is a direct development of the earlier recursive relationship, i.e.:

E(Assist)= {Scientist|Engineer|Technician: 1-1: Scientist|Engineer|Technician}

The definition is further elaborated if the same relationship were to be introduced with various properties. A possible definition of the latter case could be:

E(Assist)= {(Engineer|Technician: N-M: Engineer|Technician), (Technician: 1-N Scientist), (Engineer|Scientist: 1-1: Scientist)}

Though it may seem as if the above definition suggests redundancy, the reality is quite different. The repetition is integral in a context which emphasises the significance of the explicit indication of structural properties, but only at the definition stage. Such repetition is not carried through to further design stages, and thus poses no liability in the sense of possible redundancy.

8.2.2.3 Participation dependency

Finally, participation of concepts in a relationship are subject to dependency rules which specify total, partial, and functional dependences.

A concept can be defined, in a relationship, as a total participant if every instance of the concept class must participate in the relationship. The implication, here, is that every relationship instance is dependant on the existence of an instance of the concept class. Once the concept instance is removed, the relationship is then also terminated. Total participation is simply indicated in the relationship definition by the letter *T* in brackets.

between "Person" and "Corporation" on the one hand, and "Automobile" and "Truck"

Figure 8.1 An example of the category concept (Elmasri et *al*, 1985)

In SECOM, however, there is a realization that categories are in effect additional constructs which require their own management scheme. But even more critical, categories impose a restraint on participating concepts in that all category participants must share the same constraints, i.e., cardinality, and existence dependency. This, of course, limits what can be modelled using categories, unless constraints are specified between concepts and categories (and not relationships) which would, then, defeat the purpose of the category and categorisation.

The explicit indication of concept participation in a relationship, provided in this work, would not only inherently allow for managing multiple concept participation, but will also clarify multiple constraints within the same relationship construct.

8.2.2.2 Cardinality

Cardinality is the most expressed relationship property. This property indicates the constraint on the frequency of concept participation in a relationship instance. A concept may participate in a relationship on a one-to-one, one-to-many, and many-to-many basis. This property is indicated by distinct statement of cardinality in the relationship extension in the following fashion.

$E(R)= \{C_m: 1\text{-}N : C_n\}$

For example, The relationship "Teach" between "Lecturer" and "Course" may be specified as:

E(Teach)= {Lecturer: 1-1 : Course};
E(Teach)= {Lecturer: 1-N : Course}; or
E(Teach)= {Lecturer: N-M : Course}

In the first case, a single lecturer may only teach one course; and accordingly, a course may only be taught by a single lecturer. In the second case, a lecturer may teach more than one class, i.e., an N specified number of course; and as in the previous case, a course can only be taught by a single lecturer. In the third case, a lecturer may teach an N number of course, and a course may be taught be an M number of lecturers.

While involving more detail, the case of multiple concept participation requires a systematic development of earlier definitions. As previously indicated, multiple concept participation in a relationship takes a number of forms. Each possible participation configuration is accounted for below.

The first possibility is of more than one concept participating on equal terms on one side of a relationship. In this case, cardinality is straightforwardly indicated. For example, the relationship "Manage" in which an "Engineer" or a "Supervisor" may manage one project or one "Team" is defined as:

E(Manages)= {Engineer|Supervisor: 1-1: Project|Team}

This indicates a one-to-one relationship. Here, an example of the inadequacy of applying categories clearly emerges, because the relationships between the concepts on the left and each of the other concepts on the right, are considerably different.

On the other hand, participation maybe partial, meaning that any number of instances of a concept can participate in the relationship. This is indicated by the letter *P* in brackets.

For example, the relationship "Support" between "Parent" and "Dependents" would normally be defined as:

E(Supports)= {Parent: [P] 1-N [T] : Dependent}

This is an indication of the fact that the existence of a "Dependent" is dependent upon the existence of an instance of "Parent", while a "Parent" may exist independently.

The same definition illustrated by the above example is systematically extended to manage multiple concept participation.

The last dependency property is the functional constraint in which at most one instance of a concept may participate in a relationship at any given moment. This is indicated by the letter *F*, and is illustrated in the following example.

E(Lives-in)= {Person: [T] N-1 [F]: City}

This constraint is a sub-set of partial dependency, but because of its importance it is distinctly indicated.

8.3 Relationships and semantic relations

Relationships between concepts, as specified in the application environment, are just as critical in conceptual modelling as the inherent concept relations defined in the previous chapter. While the former introduces new concepts based on existing ones, the later serves important purposes such as consistency checking and redundancy maintenance.

The implication of semantic relations on relationships is critical to the effectiveness of integrity management. Therefore, semantic relations will be examined in view of relationships to illustrate the connection between the two concepts.

The semantic relations of inclusion, intersection, and convergence are addressed separately below. Semantic independence will not be discussed because it is not relevant to the issue under consideration.

8.3.1 Relationships and inclusion relation

Semantic inclusion has a clear and direct inference on a relationship and its participating concepts. The implication, here, is that a relationship between two concepts, of which one is a *definer*, infers that the relationship is also applicable to all refining concepts, i.e., the extension of the defining concept. When, for example, the concept "Employee" which defines the concepts "Engineer" and "Secretary", is involved in the relationship "Lives-in" (Figure 8.2). Then, "Lives-in" is also inherited by the extension of "Employee", i.e., "Engineer" and "Secretary".

E(Lives-in)= {Employee: [T] N>1 [F]: City}

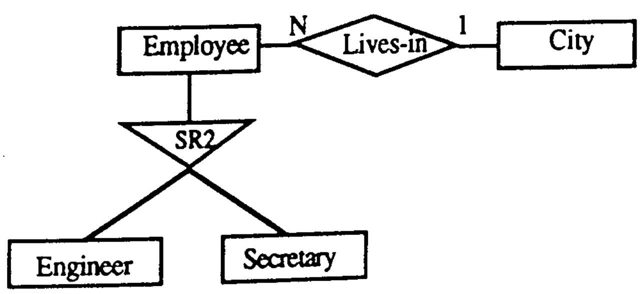


Figure 8.2 An example of a relationship over a defining concept

In other words, attributes of "Engineer" and "Secretary" are participants in the relationship, by virtue of being defined by "Employee". Therefore, an expanded definition of "Lives-in", depicted in figure 8.3, would be:

E(Lives-in)= {Engineer|Secretary: [T] N>1 [F]: City}

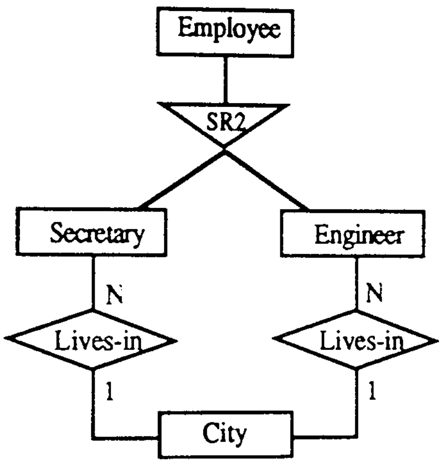


Figure 8.3 The extended version of the relationship

However, the latter definition would be done away with, simply because indicating the defining concept, in the context of semantic relations, is sufficient for the identification of the implicit extent of the relationship.

In the case of a relationship involving a refined concept, the same underlying approach is followed. Therefore, a relationship is only extended over the extension set of the participating concepts.

8.3.2 Intersection and convergence

In the case of semantic intersection and convergence (or SR3, SR31, SR11), the implication is slightly different. Here, the situation is more detailed because the semantic relations concerned involve specific attributes and not concepts.

In this case, the application of a relationship is directly linked to the attributes participating in the definition of the semantic relation. In other words, changes, as a result of the participation in the relationship, in the concept over which the convergence is defined are carried through to the converging concepts.

This case will be exemplified by a relationship involving an SR31. Each of the concepts "Manager" and "Engineer" are said to have an SR31 with "Engineering-Manager". Thus, a relationship involving "Manager" or "Engineer" is implicated in the new concept only if attributes of instances of "Manager" or "Engineer" that are part of the new concept are incurred in the relationship. In other words, only if the relationship extends over the intersection of the defining concepts, that the new concept will be involved in the relationship. What this leads to is that the implication of the relationship on the convergence is carried through to its defining concepts (figure 8.4).

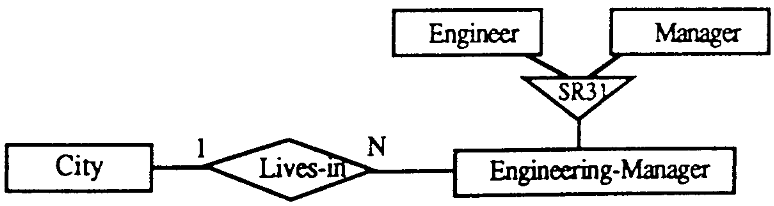


Figure 8.4 An example of a relationship over semantic convergence

8.4 Conclusion

This chapter presented the second perspective of SECOM in which relationships between concepts were introduced. The significance of relationships, as defined in this context, is elaborated by semantically strengthening the concept of a relationship. While in the conventional view, a relationship is considered a link or an association between concepts devised to establish basis for attribute relating, this work values relationships as concepts with their own distinct meaning. Such view enhances the understanding of the connection between the semantics of concepts and relationships. This same connection was then utilized to define relationships in accordance with the semantic definition of concepts.

Chapter 9

SECOM III: The organization perspective

The introduction to organizations, presented in chapter 2, outlined the implication of organization structure on information processing and flow. It has also pointed out that the implication is reciprocal, i.e., the design and implementation of information systems impacts the functioning of the organization.

Chapter 4, on the other hand, addressed the organization hierarchy, the most prominent feature of its structure. The structure hierarchy was directly linked to the quality and quantity of information required at any position in the organization.

The relationship between the organization and the database, through the information system, was outlined in chapter 6. The significance of this relationship was established through the relationship between required information and data definition and representation.

This chapter will present the necessary tools for managing the implication of the relationship between the database and the organization. More specifically, underlying organization consequences on information representation will be accounted for in order to create a modelling environment in which conceptual abstractions are constructed in view of organization structure considerations which give rise to such abstractions. Accordingly, the ensuing conceptual abstractions will contribute better information representation because of the high degree of information relevance. In this sense, the concepts introduced in this chapter depart from those of established modelling approaches.

9.1 Introduction

Databases can be defined as an accumulation of the data used throughout the organization, irrespective of the type of activity utilizing the data (e.g., management, decisions, operations). In this sense, the definition of the database is a primitive one. Data is only bound by the classification of concepts and the inherent links between such concepts.

On the other hand, organizations are entities consisting, among other things, of people utilizing the data stored in the database to carry out specific tasks. Employees are organized according to established methods of work assignment. Such methods are based on the recognition of a person's capacity to perform related duties. An integral part of such capacity is the level of conceptual abstractions a person is capable of comprehending and manipulating.

Because data are in effect linguistic expressions, and databases utilize abstracts of reality, the link between organizations and databases can be easily perceived. Usually the link between an organization and its database is established through the information system. This was the concern prior to establishing the relation between language and organization structure (i.e., through semantic theory). However, the task, here, is to utilize such link to build stronger more distinct attachments, between organizations and databases, to improve database utilization.

Utilizing the link between the structure of the organization and its database will be realized through the creation of relations between data which will reflect conceptual representations according to the level of abstraction implied by the organization structure. Such relations will be referred to as abstraction relations or ARs.

Before introducing organization relations, it is necessary to briefly re-examine the definition of a user view in the context of the proposed approach.

9.2 Structure levels and database views

In chapter seven, user views were defined as mechanisms through which users acquire their data requirements. Views were also said to consist of representations of concepts and relationships.

Here, views will be considered as properties of the structure level in which they are utilized. A view of a user at the first hierarchic level, for example, is a level 1 view, and so on. Accordingly, representations are contingent upon the views in which they participate.

The purpose of such definition, is to devise a method for constructing conceptual abstractions through user views which will reflect the level of data and information abstraction in reality. In other words, user views will be inter-related in accordance with linguistic abstractions, which were defined as a common denominator for the multiplicity of user representations and the sole identifier of data, information, and structure abstractions. This notion is illustrated in figure 9.1, in which database views at different structure levels are shown to correlate with the levels of linguistic abstractions. In the figure, linguistic abstractions presented by Korzybski (1933) (shown in figure 6.3) are utilized to demonstrate such correlation.

The actual materialization of links among database views, i.e., the composition of view abstractions, will emerge as a result of establishing relations and, accordingly, imposing constraints over the level of detail within each view in relation to that of the views above and/or below it.

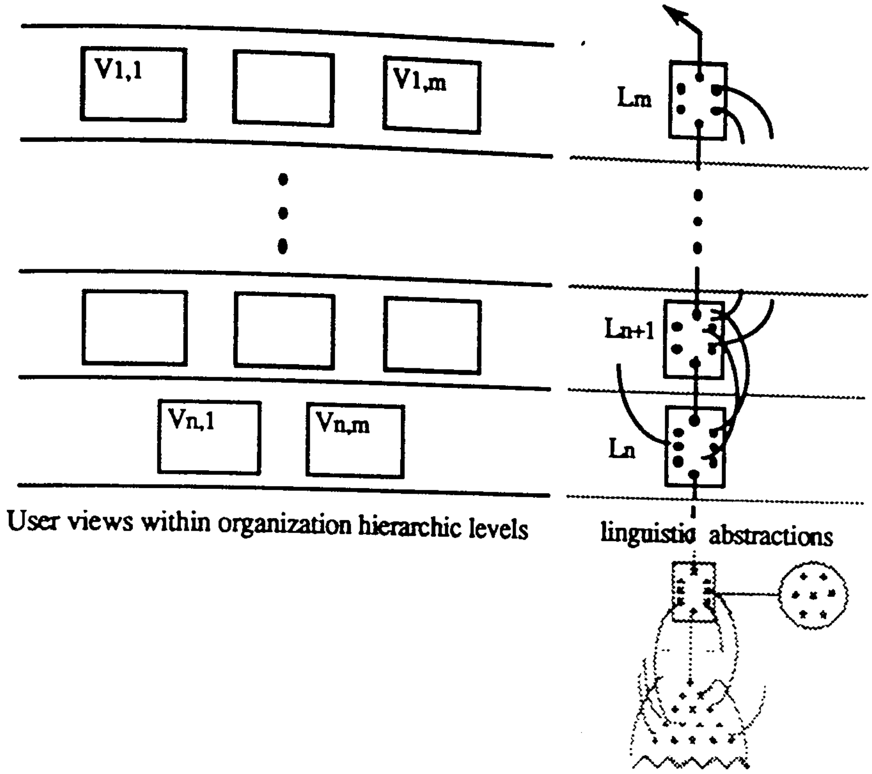


Figure 9.1 User views, structure levels, and levels of linguistic abstractions

It is essential, however, to take into considerations the fact that such relations and constraints are not exclusive. In other words, the designer should not impose constraints presented here on users requirements, nor should the case be that designers decide the content of a certain view. The aim, here, is not the definitive design of a database, but is rather enhancing and complementing the established approach. What has to be taken into consideration, is that the method presented in this chapter confirms existing organization and information system propositions. In addition, it makes use of established data abstraction techniques. With this in mind, designers have to approach this step of the process with full understanding of the issues involved. Only such understanding will enable the designer to make optimum decisions pertaining to the nature and extent of the relations and constraints to incorporate into the database environment.

It is justifiable, nonetheless, to assert that in an appropriately designed organization, in which an appropriately designed information system is utilized, the application of conceptual abstraction methods, as presented in this chapter, should emerge with no conflicting regularities. The justification of this assertion is based on the reality that the emergence of consistent information abstractions is an inevitable consequence of a well designed organization and information system.

In order to relate a view to a certain level, the notion of a view introduced in chapter 7 will be utilized here. Therefore, a user view will be referred to as:

$V_{l,n}$

where l is the view number, and n is the hierarchic level of the user utilizing the view. Each representation within a view is indexed according to its corresponding view. In other words, a representation of a concept or a relationship is indicated as:

$T_{l,n,t}$,

where l and n are as indicated earlier, and t is the representation number. Representations themselves, have already been linked to their corresponding concept classes and relationship sets. Indexing of concepts, relationships, and views has been utilized in previous work (De *et al.*, 1982; Sen, 1982; Kangassalo, 1984).

9.3 Hierarchical-abstraction relations (ARs)

The link between representation abstractions and organization levels will be realized through the establishment of relations between conceptual classes at different organizational levels.

The relations presented in this context are defined over concept abstractions. Such abstractions are either semantic or non-semantic-based, depending on the relations between the concepts within the abstractions. The aim will be to maintain representation consistency by checking for situations in which more general concepts are represented at lower organization levels and vice versa. In which case, the inconsistency is modified so that no general concept is represented at a lower level, while a more specific form of that concept is represented at a higher level.

The objective of abstraction relations is to ensure control over the level of detail and specificity of representations within the levels of user views. This control, as earlier mentioned, is implied by the hierarchic structure of the organization, manifested by the introduction of specific concept relations, and identified by the designer with adequate user participation. Six ARs are defined within four categories of abstraction relations, and they are as follows.

9.3.1 Semantic abstractions

This category of vertical structure relations is devised to control the level of concept generality represented at each hierarchic level according to representations of concepts at other hierarchic level. As its name suggests, this relation is identified over semantically defined abstractions. More specifically, this relation is defined over the relations of semantic inclusion, semantic intersection, and semantic convergence (SR2, SR3, and SR31).

9.3.1.1 Hierarchy-inclusion relation (AR11)

Hierarchy-inclusion is an abstraction relation in which a refining concept assumes the representation of one or more of its definers, so as to ensure hierarchy-levels' consistency with the levels of linguistic abstractions.

This relation is utilized when the following modelling context is encountered. A user at level n requires a representation of concept C_k, and a second user at a lower hierarchic level n+1 requires a representation of another concept C_l, which is more general a concept than C_k, i.e., C_l <SR2> C_k. The implication of this situation is that the user in the lower level has a more abstract view than that of the user at the higher level. This, clearly, contradicts both effective information utilization principles, and basic rules of linguistic abstractions. This contradiction is highlighted in figure 9.2, in which a subsection of figure 9.1 has been incorporated to illustrate the contradiction.

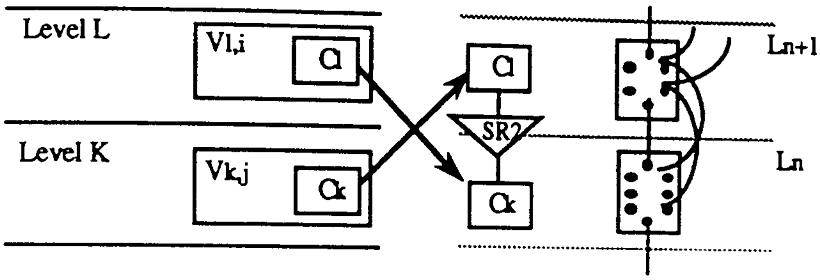


Figure 9.2 Representations contradicting abstraction rules

To resolve such inconsistency, a relation will be drawn between the representation at the higher level and the definer of the two concepts, to indicate that the higher representation is to assume an equal level of detail to that of the defining concept represented. This would result in making the representations contain similar level of detail.. This is illustrated in figure 9.3 in which the situation in figure 9.2 is corrected.

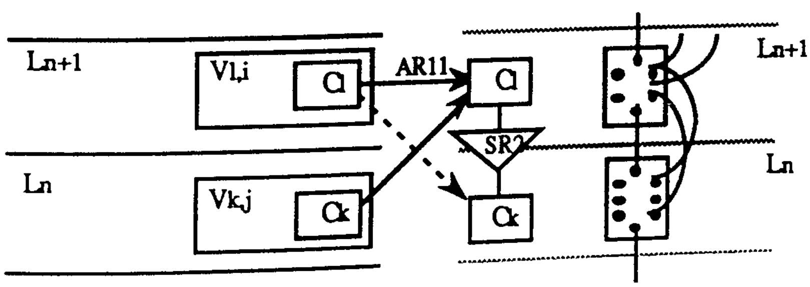
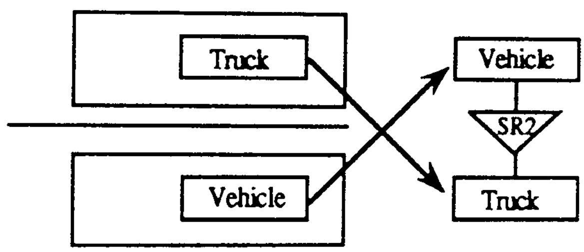
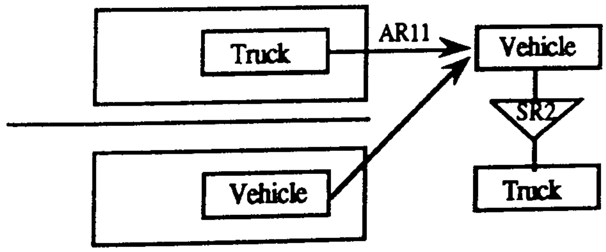


Figure 9.3 A correction of the representation contradiction presented in figure 9.2

To clarify this relation, consider the following example. The concept "Truck" is defined in the view of a user at the second hierarchical level. At the same time, the concept "Vehicle" is defined as part of the view of a user at the first level. The result is that more detail will be shown at a higher level, i.e., "Vehicle" information and "Truck" specific details. To prevent this situation from taking place, only "Vehicle" specific attributes of "Truck" will be represented for the user at the highest of the two levels. The relation is therefore drawn between the representation of "Truck" within the view in question, and the concept "Vehicle". The application of AR11 in this example will readjust the representation anomaly, as illustrated in figure 9.4 a,b.



(a)



(b)

Figure 9.4 An example of the hierarchy-inclusion relation:

(a) An abstraction-hierarchy conflict

(b) The application of AR11

What the relation will do, for the example above, is to present the higher level user with a less detailed account of "Truck".

Naturally, this relation is not bound to two concepts with a semantic inclusion relation between them. Rather, it extends over any number of concepts as long as they are within the same inclusion abstraction hierarchy. In other words, the relation is utilized whenever the rule of hierarchy-abstraction correspondence is violated.

9.3.1.2 Hierarchy-convergence relation (AR12)

This relation is similar to AR11 with the difference that the concept represented at the higher hierarchic level is a convergence of two concepts. The relation is therefore established when a view of a user at a higher level includes a representation of a concept that is defined based on the convergence of two concepts (i.e., SR11 and SR31), while a view of a user at a lower level includes one (or more) of the converging concepts. The implication here, is that since the convergence results in a more detailed, and thus more specific concept, a hierarchy-abstraction conflict results (figure 9.5).

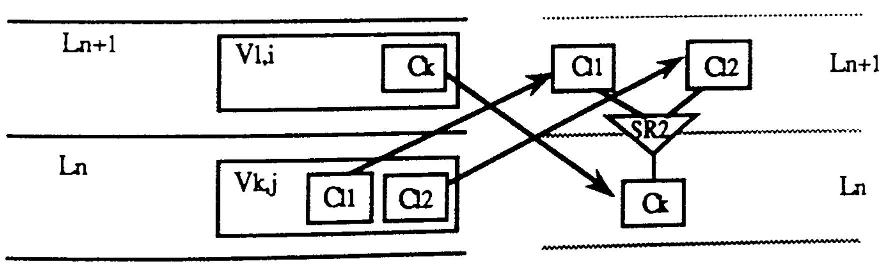


Figure 9.5 A case of hierarchy-abstraction conflict over meaning convergence

To overcome such inconsistency, one of two alternative resolutions is invoked, according to the type of convergence.

In the case of prevalent convergence, the situation is corrected by limiting the representation of the more specific concept to one equal to the representation of the common attributes of the converging concepts, which is equal to the representation of their common defining concept (figure 9.6). This will have the same effect as that of AR11. In other words, the level of detail will be limited to one less than or equal to that of the concept represented at the lower level.

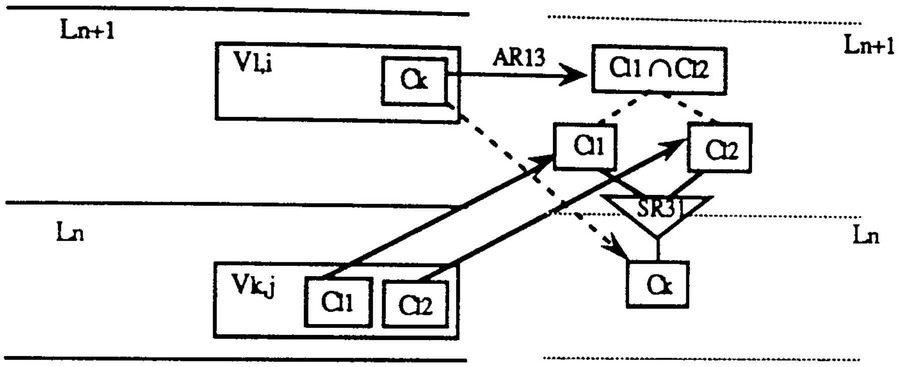
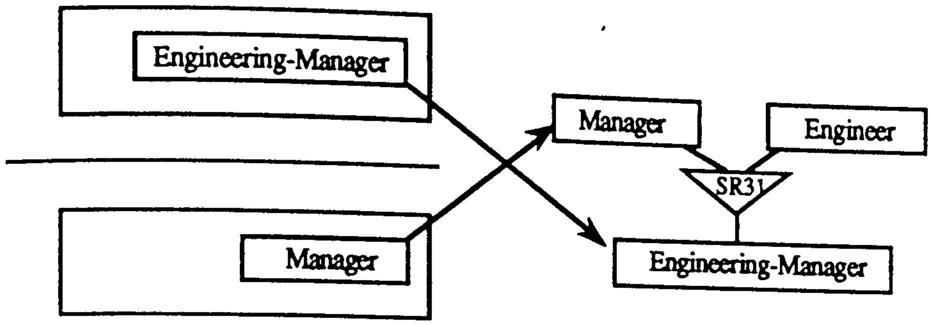
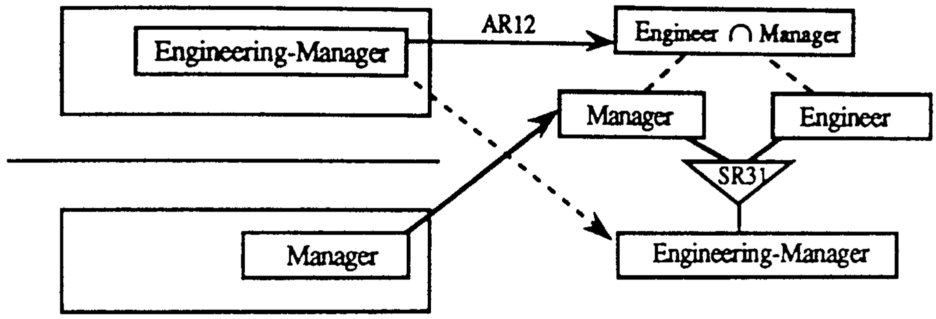


Figure 9.6 Rectifying representation inconsistency

On the other hand, if the convergence is atypical, then the above solution is not applicable. This is mainly because the common attributes among the converging concepts will not provide an acceptable representation. However, and keeping in mind that what is required is a more general form of the convergence, the solution is to assign the representation of one of the defining concepts as proxy for the convergence itself. It is then left to the designer to establish what the application requirements for the representation at the higher level are, and accordingly, designate the representation of one of the defining concepts for the higher concept. This abstraction relation is exemplified in figure 9.7a and b.



(a)



(b)

Figure 9.7 An example of hierarchy-convergence relation:

(a) Inconsistent representation in the case of an SR31 relation, and
(b) An application of AR12

9.3.2 Derived abstractions

The second category of abstraction relations is specified over user derived abstractions. Such abstractions are superficial, in the sense that relations between the concepts within such abstractions are not formulated over intrinsic semantic definitions. The abstraction relations according to this category are as follows.

9.3.2.1 Partition abstractions relation (AR21)

Concepts sometimes define other concepts through non-semantic relations. One of such relations is that of partitioning in which a concept class is split into two semantically identical concepts so that instances of one partition are distinguished from instances of the other by a difference in an attribute value. A concept can also be divided into more than two partitions, just as the partition itself can be defined over more than one attribute value.

This abstraction method has been defined elsewhere in the literature (dos Santos *et. al*, 1980; Malorg, 1986; Su, 1983). A detailed definition of this abstraction is presented by Battista & Batini (1988).

Here, the new concept will be referred to as the partition set, whereas the defining concept will be referred to as the partition definer. Therefore, a concept is identified as a partition if the difference between its meaning and that of its defining concept is established by a value of one or more of the defining concept attributes.

An AR21 is defined over partitions and their definers. Such definition implies that a partition set may not be represented at a higher level than its defining set. This is because the partition is a more detailed concept than its definer. Therefore, an AR21 is established between a higher representation of a partition and the partition definer, when the definer is included in a representation at a lower level.

This relation is a constraint on data representation that is devised to ensure consistency of organization hierarchy and information representation. Moreover, upholding such constraint would ensure the integrity and applicability of other abstraction relations.

As an example of a partition, consider the concept "Employee", and the sets "Male-employee" and "Female-employee" defined over the attribute "Gender". Here, the value of the attribute "Gender" decides the difference between each of the partition sets and the concept class "Employee". Installing an abstraction relation in this case would ensure that the details attached with each partition set is suppressed for the purpose of achieving a more general concept.

9.3.2.2 Summary relations (AR22)

While a partition set is specified according to its defining concept's attribute value, a summary concept is defined over instances of one or more concepts. Concepts defining a summary can have different meaning than the summary concept itself.

The concept of summary information and summary concepts is usually addressed in statistical databases (Hebrial, 1986, Battista & Batini, 1988). It is included in this context because of its relevance to abstraction.

In a summary concept, a value of an attribute is a number that is derived by applying a mathematical operation over a set of other concept's attribute values. In this case, the concept whose attribute value is mathematically derived is referred to as the dependent concept, and it is defined over the independent concept(s).

This relation, as in other relations in this category, implies that the independent concept is represented at a higher level than one in which the dependent concept is represented. Moreover, the value of the attribute to which the result of the mathematical function is returned cannot be updated. On the other hand, any update in the attribute(s) of the independent concept should trigger a systematic update in the dependent, or summary concept.

As a mathematical rule, no concept should be dependent on the value of another concept, which in turn is dependent on the original one. In other words, if the the concept C_k is a summary concept derived from, among other concepts, C_l; then C_l can not be a summary concept which is derived form C_k.

The abstraction relation defined over a summary set is identical to that of partitions. Namely, a summary set is constrained to a representation at a higher level than the level in which its defining concepts are represented. An example of a summary concept is one which includes totals and averages.

9.3.2.3 Relationship abstractions relation (AR23)

Relationship abstractions were presented in chapter 2 under semantic data models. The concept of a relationship abstraction is derived from an aggregation of two or more concepts and a relationship between them. In a sense, a relationship abstraction forms a new concept whose definition is a by-product of its constituent concepts and relationship. However, and because the concept developed by the abstraction is a concoction of existing concepts, independent conceptual definition of the new concept is not warranted.

Relationship abstractions are not too different from the semantic convergence of two concepts' meanings. The main differing aspect between the two is that a meaning convergence has a reference class of its own, which translates into an extension. On the other hand, a relationship abstraction has no clear reference. The implication of such difference is that a convergence is defined as a concept class independently, i.e., independent instances of the convergence can be found irrespective of application, whereas relationship abstractions are defined over the concepts participating in the relationship, in addition to the relationship itself.

In the case of representing relationship abstractions, the hierarchy-abstraction consistency has to be maintained. Therefore, when the concept participating in a relationship abstraction is represented at a higher level than one in which a representation of the abstraction itself is represented, then an AR23 relation is predicated between the concepts and the abstraction to indicate an alteration of the representation. In such alteration, the higher representation - which includes the abstraction concepts - assumes the representation of the abstraction itself.

An AR23 is exemplified as follows. Figure 9.8a shows the abstraction "Management" which is defined over the concepts "Manager" and "Project", and the relationship "Manage" between them. This indicates the requirement for representing the concepts "Manager" and "Project" at a lower level than one in which "Management" is represented. As a result, and as illustrated in figure 9.8, the representations of "Manager" and "Project" are replaced with a representation of "Management" through an AR23 between the representation of each of the concepts and the relationship abstraction.

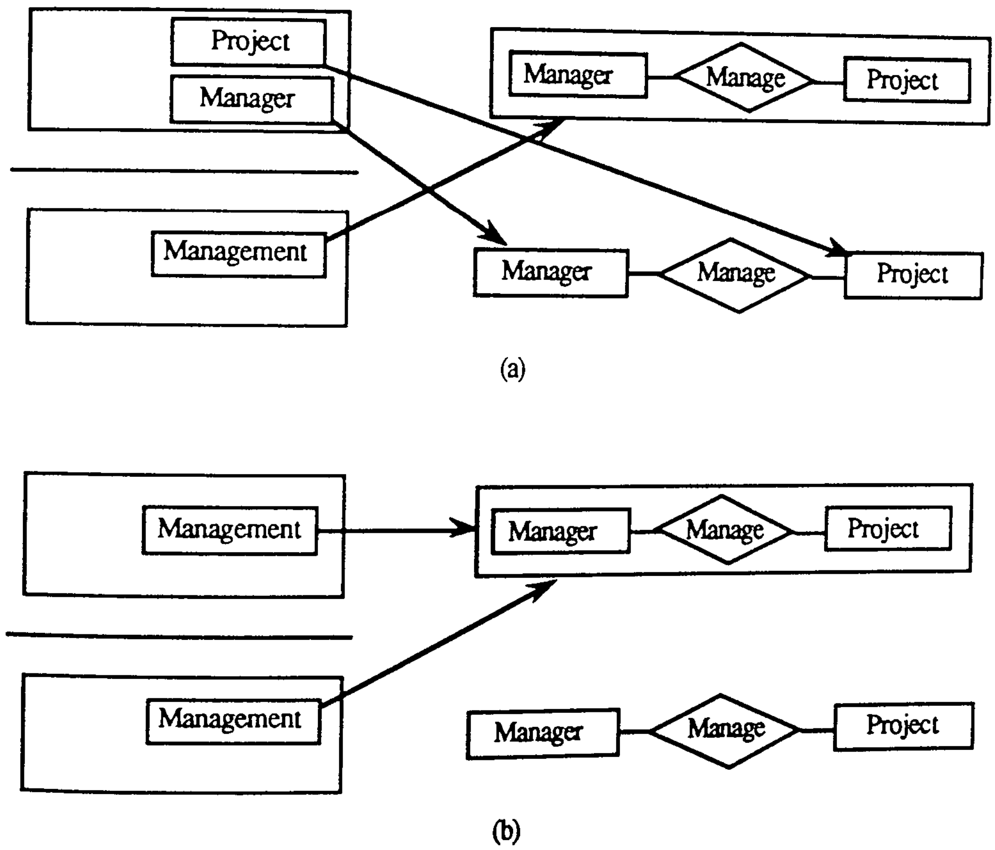


Figure 9.8 An example of a relationship abstractions relation
(a) Inconsistent representation of a relationship abstraction
(b) A correction of the above representation

9.3.3 Representation abstractions relation (AR3)

Finally, an abstraction relation is defined between representations of the same concept. This relation indicates that the same concept may not be represented in more detail at a higher level than at a lower one. This relation can be materialized in the context of the suggested approach, since there is a provision which allows and accounts for multiple concept representations.

9.4 Conclusion

In this chapter, the final dimension of SECOM was presented, in which relations were established to ensure the coherence of concept representation within the organization hierarchy. It is important to mention, however, that the application of organization abstraction relations is not a straightforward process. The designer's understanding of the application area should serve as a guide to the implementation of such relations. Strict application of abstraction relations may not always be possible. Nonetheless, and because the relations promote information relevance, their incorporation is necessary.

Chapter 10

SECOM: Application in an academic department

This chapter demonstrates an implementation of SECOM through its application in a university academic department. This demonstration will illustrate the execution of SECOM operations.

10.1 Introduction

A university academic department presents a realistic environment for illustrating the application of conceptual modelling techniques. This is mainly because a department consists of a structure that is, to a large extent, independent from that of other university departments, and therefore constitutes an organization within the university.

Because SECOM is applied at the conceptual level, it is assumed, as previously mentioned, that user requirements have been outlined in advance. For the purpose of this chapter, information provided by a current information system project in the Department of Computer Science and Applied Mathematics, at the University of Aston in Birmingham, will be utilized. The project, according to Avison & Wood-Harper (1990), has a long term aim of "fulfilling the information requirements of staff and students of the department." Further details on such projects can be found in publications on Departmental Information System (DIS) (Boshell, 1988, Foster *et al.*, 1989). The example in this chapter will, therefore, refer to the case as the DIS example.

The utilization of existing information on the department user requirements and specifications will not only provide a factual modelling situation, but will also present a realistic test bench for examining the potentials of SECOM.

Before proceeding with the analysis, it is important to mention here, that because of the multiple perspectives of SECOM, it will not be possible to provide a complete diagrammatic representation of the DIS example in a single diagram in which all concepts, relations, and relationships are depicted. Any attempt to do so would inevitably lead to one of two outcomes: a diagram that is physically too large to handle; or a diagram that is too complex to be informing. Clearly, either case is self defeating. Instead, each perspective will be represented separately. Under the circumstances, this seems a logical way of model representation since each perspective is concerned with a different aspect of conceptual modelling.

10.2 User views

Boshell (1988) defined user views based on previous work, interviews with users, and refinements of existing information specifications. Twelve user views were defined and outlined, and they are as follows:

1- Admissions officers (UCCA, non-UCCA, and research admissions officers)
2- Supervisor
3- Research student
4- Course tutor
5- Non-Academic staff
6- Lecturer
7- Student
8- Examination officer
9- Timetable officer
10- Jobs/Placements officer
11- Computer registration
12- Database administrator

In addition to the above views, further work identified the need to define new views (Foster *et al.*, 1989). Of the additional views, the "Head of department" view will be included in the discussion. The total number of views consists of a large number of concepts (around 100). Therefore, and to keep the example manageable, only a subset will be utilized for the purpose of this chapter.

The subset views were selected based on their capacity to illustrate the application of SECOM. Consideration will therefore be given to the following views (note that "Admission officers" view is in fact three distinct views).

1- Head of department
2- UCCA-admissions officer
3- Research admissions officer
4- Supervisor
5- Research student

At this stage, the operations to be performed in the context of SECOM will be presented to give an outline and a general idea of the direction to the process at hand.

10.3 DIS conceptual modelling

The conceptual modelling of the DIS example will proceed according to the following steps.

Step 1 will be to identify organization levels of users. Such identification requires examining user positions in view of the stratification methods presented in chapter 4. In the current example, it appears that the users reside over three distinct organization levels: the head of department at the highest of the three levels; supervisors and admission officers at the next level down; and research students at the lowest of the three levels.

Note that this step will usually identify at least two levels. This is because the identification of less than two levels is an indication of the small size of the organization. Application of the suggested approach in small organizations is, as earlier stated, not recommended.

Step 2 is then to link each user view to his/her respective level. For example, the "Head of department" view is linked to level 1, according to the following notation:

V1,1

In the case where a second view is defined for "Head of department", the notation will then be:

V1,2

The link of a view to the level of organization structure will provide the reference for hierarchy abstractions, which will be used to maintain the consistency in concept representation (as will be shown later on).

A user view is perceived, in this context, as having representations. Accordingly, in step 3, each representation will be assigned a number, and will be indexed according to the view in which it is required. Therefore, the representation "Funding" in the "Head of department" is assigned the index (1,1,1), which indicates that "Funding" is the first representation of view number 1, which is at the first level of the structure hierarchy.

Once all the representations are linked to their perspective views, step 4 is to define, for each entity, its meaning through its intension and extension. In addition, a representation is specified according to user specific requirements. This step is called NIET definition since it is the step in which the name, intension, extension, and representation of an entity are defined.

Step 5 is concerned with classifying and integrating concepts (and not entities). The classification and integration is performed over the meaning presented by intension and extension. This step is performed according to MBC methods explained in chapter 7.

In step 6, and after conceptual classes have been identified, semantic relations are established among concepts. Concepts are examined to determine the presence of any one the semantic relations developed in chapter 7 (i.e., SR1, SR2, SR3, SR31, and SR11).

In step 7, relationships between concepts are determined according to the definitions provided in chapter 8. The relationships are indicated diagrammatically by a straight line connecting the concepts, with the name of the relationship next to it.

In step 8, concepts and representations are examined for consistency in abstraction-hierarchies. This step is aimed at ensuring that concepts and representations are compatible with the structure levels in which they are represented. In other words, representations of more general concepts are provided at higher (or equal) levels than the representations of more specific form of such concepts. In addition, representations of the same concept, at different levels, correspond to the convention of less detail at higher levels, and vice versa. This step is accomplished through the application of the third perspective of SECOM, in which abstraction relations are defined (i.e., AR11, AR12, AR21, AR22, AR23, and AR3).

10.4 SECOM application

The first thing to do at this stage, is to designate the level of organization hierarchy in which each user view is ranked. Methods for identifying user positions were presented in chapter 4. The above views will belong to users at three varying organization levels. As such, the list of available views will be referenced accordingly. Namely:

V1,1: Head of department

V2,1: UCCA-admissions officer
V2,2: Research admissions officer
V2,3: Supervisor

V3,1; Research student

The basic requirements for such views include representations defined by users. Figures 10.1, 10.2, 10.3, and 10.4 provide information on each view, including the data requirements (the information about views in the figures are paraphrased from existing user data models identified elsewhere (Boshell, 1989, Foster *et al.* 1989)).

V1.1

Figure 10.1 Head-of-Department view information

Aston University

Illustration has been removed for copyright restrictions

Figure 10.2 Admissions Officers view information

Aston University

llustration has been removed for copyright
estrictions

Figure 10.3 Supervisor view information

Figure 10.4 Research-student view information

Views, in this context, are defined as consisting of representations rather than concepts. The representations in the views under investigation are, therefore, indexed according to their relevant views. Such representations are listed in figure 10.5, thus creating a basic view-representation reference list.

Head of Department	UCCA-admissions Officer	Research-admissions Officer	Supervisor	Research Student
V1,1	V2,1	V2,2	V2,3	V3,1
1,1,1: Funding	2,1,1: UCCA-Candidate	2,2,1: Research-Candidate	2,3,1: Supervisor	3,1,1 Supervisor
1,1,2: Funding-body	2,1,2: Non-research-student	2,2,2: Student	2,3,2: Non-research-student	3,1,2 Student
1,1,3: Lecturer	2,1,3: Course	2,2,3: Funding	2,3,3: Research-student	3,1,3 Project
1,1,4: Research-student	2,1,4: Supervisor	2,2,4: Funding-body	2,3,4: Project	3,1,4 Report
1,1,5: Contract	2,1,5: Area-of-interest	2,2,5: Supervisor	2,3,5: Research	3,1,5 Message
1,1,6: Non-academic-staff		2,2,6: Area-of-interest	2,3,6: Report	3,1,6 Publication
1,1,7: Aggregate-stud-info			2,3,7: Publication	3,1,7 Area-of-interest
1,1,8: Bulletin-board-msg			2,3,8: Area-of-interest	3,1,8: Conference
			2,3,9: Conference	3,1,9: Seminar
			2,3,10: Seminar	3,1,10: Seminar-speaker
			2,3,11: Seminar-speaker	3,1,11: Conference-msg
			2,3,12: Conference-msg	3,1,12: Internal-paper
			2,3,13: Paper	3,1,13: External-paper
			2,3,14: Funding	

Figure 10.5 Representations within available views

10.4.1 NIET definition

For each representation within each of the views at hand, users are then required to define, exactly, their intension of each representation name listed. This is equivalent to asking the question "what is meant by...?" The intension of each representation is then included in a set containing the properties of that entity. The intended purpose of defining intension is to arrive at the meaning of the representation at hand. For example, the intension of the representation "Research-student" can be determined by placing the question: "what is meant by a research-student?"

The extension of a representation is then indicated in the same way as in intension. The question, in this case, is in concurrence with "what (or who) can be a ...?" For example, the extension of the representation "Research-student" can be determined by asking: "who can be a research-student?"

The definition of a representation's name, intension, extension, and attributes provide the necessary concept definition, or NIET definition. For the DIS example, NIET concept definitions are listed under their corresponding views in figures 10.6, 10.7, 10.8, 10.9, and 10.10. Such definitions will be referenced throughout the modelling process.

V1,1	HEAD OF DEPARTMENT			
		Meaning (M)		
Ind	Name (N)	Intension (I)	Extension (E)	Representatation (T)
1,1,1	Funding	A financial obligation	(Research-fund, Non-research-fund)	title-of-funding, amount-of-funding, type-of-funding, start-date, end-date, funding-organization
1,1,2	Funding-body	Funding provider	Funding-body	address, funding-organization, funding-contact-name
1,1,3	Lecturer	(Person), employed by the department to lecture courses to students	(Lecturer, supervisor)	id-code, name, qualifications, role
1,1,4	Research-student	(Student), working on a degree through research	(Research-student, lecturer)	application-no, id-code, nationality, eec-member,
1,1,5	Contract	An official work agreement	Contract	id-code, date-of-appointment, length-of-contract, contract-start-date, contract-end-date
1,1,6	Non-academic-staff	(Person), non participant in an academic activity	(Adminstrator, Secretary)	id-code, job-description
1,1,7	Bulletin-board-msg	(Message), for display on a bulletin board	Bulletin-board-msg	bulletin-type, bulletin-entry-date, bulletin-deletion-date, bulletin-entry-no

Figure 10.6 NIET definitions for Head-of-deprtment view

$V_{1,1}$	HEAD OF DEPARTMENT			2
		Meaning (M)		
Ind	Name (N)	Intension (I)	Extension (E)	Representatation (T)
1,1,8	Term-details	Information on students and projects on a particular term	Term-details	term-date, no-of-students, no-of-research-students, projects-completed
1,1,9	Publisher	A publishing organization	Publisher	pub-name, pub-address, contact
1,1,10	Seminar	A one-time lecture on a specific area	Seminar	seminar-title, seminar-date, seminar- room, start-time, abstract
1,1,11	Conference	An extended meeting of specialist on a subject area	Conference	conf-organiser, conf-date, conf-location
1,1,12	Academic-staff-res-std	(Research-student), participating in teachning activities	(Research-student), (Academic-staff)	res-std-id-code, res-std-name, Job-description

Figure 10.6 Cont.

V2,1	UCCA-ADMISSIONS OFFICER			
Ind	Name (N)	Meaning (M) Intension (I)	Meaning (M) Extension (E)	Representatation (T)
2,1,1	UCCA-Candidate	(Candidate), applying to the department for a degree through UCCA	UCCA-Candidate	ucca-no, cand-name, cand-address, cand-tel-no,course-name, university-offer, candidate-response
2,1,2	Non-research-student	(Student), a person working for a departmental degree through coursework	(MSc-student, BSc-student)	application-no, id-code, nationality, eec-member, last-educational-establishment, start-date, expected-end-date
2,1,3	Course	As defined by the university	Course	course-code, course-title, course-profile, admiss-requierments
2,1,4	Supervisor	(Academic-staff) supervising research-students	(Supervisor, Lecturer, internal -supervisor, external-supervisor)	supervisor-id, supervisor-pos-in-dept, supervisor-position-in-res-group
2,1,5	Area-of-interest	A field of knowledge	Area-of-interest	id-code, general-interest

Figure 10.7 NIET definitions for UCCA-admissions officer view

V2,2	RESEARCH-ADMISSIONS OFFICER			
		Meaning (M)		
Ind	Name (N)	Intension (I)	Extension (E)	Representatation (T)
2,2,1	Research-Candidate	(Candidate), applying for a degree from the department	Research-Candidate	application-no, appl-date, qualifications, country, interviewed, interview-date, offer-made, response
2,2,2	Student	(Person), working for a departmental degree	(Student, Research-student, Non-research-student)	application-no, id-code, last-educational-establishment, eec-member, nationality, start-date, expected-end-date
2,2,3	Funding	Financial provisions for research purposes	Funding	id-code, amount, start-date, funding-organization, end-date, funding-description
2,2,4	Funding-body	Provider of financial support	Funding-body	address, funding-organization, funding-contact-name
2,2,5	Supervisor	(Academic-staff), supervising research-students	Supervisor	supervisor-id, supervisor-pos-in-dept, supervisor-position-in-res-group
2,2,6	Area-of-interest	A field of knowledge	Area-of-interest	id-code, general-interest

Figure 10.8 NIET definitions Research-admissions officer view

V2,3	SUPERVISOR			
		Meaning (M)		
Ind	Name (N)	Intension (I)	Extension (E)	Representatation (T)
2,3,1	Supervisor	(Academic-staff), supervising research students	Supervisor	supervisor-id, super-name, super-title, super-room-no, supervisor-position-in-res-group, area-of-interest
2,3,2	Non-research-student	(Student), a person working for a departmental degree through coursework	(Msc-student, Bsc-student)	student-id, application-no, last-ed-establishment, o-levels-cs, course, student-status, tutor, year-of-entry, year-of-departure
2,3,3	Research-student	(Student), working on a degree through research	(Research-student, Lecturer)	application-no, id-code, res-std-name, research-area, nationality,
2,3,4	Project	Work defined as a project by the department	(Project, Research)	project-no, project-title, abstract
2,3,5	Research	(Project), work defined as research by the department	Research	research-id, title, author, supervisor, abstract
2,3,6	Report	Text on a project	(Project-report, Research-report)	author-id, date-of-report, report-title, external-moderator, result
2,3,7	Publication	Text on a subject	(Book, Paper)	author, pub-subject, pub-title, pub-keywords, pub-date

Figure 10.9 NIET definitions for Supervisor view

V2,3	SUPERVISOR			2
		Meaning (M)		
Ind	Name (N)	Intension (I)	Extension (E)	Representatation (T)
2,3,8	Area-of-interest	A field of knowledge	Area-of-interest	id-code, general-interest
2,3,9	Conference	An extended meeting of specialists on a subject area	Conference	conf-organiser, conf-date, conf-location
2,3,10	Seminar	A one-time lecture on a specific area	Seminar	seminar-title, seminar-date, seminar- room, start-time, abstract
2,3,11	Seminar-speaker	(Person), delivering a speech at a seminar	(Seminar, Person)	seminar-title, speaker-name, subject
2,3,12	Conference-msg	(Message), on a conference	(Conference, Message)	entry-no, date, location, contact-name, abstract
2,3,13	Paper	A manuscript on a specific topic	(Paper, Intern -papaer, External-paper)	paper-no, paper-date, author-id, paper-title, abstract
2,3,14	Funding	Financial provisions for research purposes	Funding	id-code, amount, start-date, funding-organization, end-date, funding-description

Figure 10.9 Cont.

V3,1	RESEARCH STUDENT			
		Meaning (M)		
Ind	Name (N)	Intension (I)	Extension (E)	Representatation (T)
3,1,1	Supervisor	(Academic-staff) supervising research-students	(Supervisor, internal -supervisor, external-supervisor)	name, supervisor-position-in-department, upervisor-position-in-res-group, area-of-interest
3,1,2	Student	(Person), studying for a departmental degree	(Student, Research-student Non-research-student)	student-id, name, address, area-of-interest, start-date, expected-end-date
3,1,3	Project	Work defined as a project by the department	(Project, Research)	project-no, project-title, abstract
3,1,4	Report	Text on a project	(Project-report, Research-report)	author-id, report-no, report-title, assessor-id, external-moderator, report-date, report-result
3,1,5	Message	A notification for display on a medium	(Messsage, Conference-msg, bulletin-board-msg	msg-no, msg-date, text
3,1,6	Publication	Text on a subject	(Book, Paper)	id-code, author, title, no-of-pages, abstract
3,1,7	Area-of-interest	A field of knowledge	Area-of-interest	id-code, general-interest

Figure 10.10 NIET definitions for Research-student view

V3,1	RESEARCH STUDENT			2
		Meaning (M)		
Ind	Name (N)	Intension (I)	Extension (E)	Representatation (T)
3,1,8	Conference	An extended meeting of specialists on a subject area	Conference	conf-organiser, conf-date, conf-location
3,1,9	Seminar	A one-time lecture on a specific area	Seminar	seminar-title, seminar-date, seminar- room, start-time, abstract
3,1,10	Seminar-spaker	(Person), delivering a speech at a seminar	(Seminar, Person)	seminar-title, speaker-name, subject
3,1,11	Conference-msg	(Message), in a conference	(Conference, Message)	entry-no, date, location, contact-name, abstract
3,1,12	Internal-paper	(Paper), published internally	Internal-paper	intrenal-paper-no, internal-paper-date, author-id, internal-paper-title, abstract
3,1,13	External-paper	(Paper), published in an external publication	External-paper	external-paper-date, author-id, external-paper-title, internal-paper-no, journal, vol, abstract

Figure 10.10 Cont.

10.4.2 MBC classification

At this stage, concepts are specific enough to warrant their classification according to the meaning implied by their intensions and extensions. MBC will therefore proceed according to the case classification process outlined in chapter 7, i.e., by comparing each representation with the remaining others. Such comparison would result in one of sixteen possibilities, or cases, which in turn are divided into four distinct groups. Instead of excessively extending the analysis, only examples of selected cases will be provided to illustrate methods of MBC.

The first MBC group is one in which representations and names are in conflict. This group includes cases 3, 5, 7, 10, 12, and 14. As an example of this group, consider the representations (2,2,2: Student) and (2,1,2: Non-res-student). The two representations have different names, different intensions and extensions (thus different meaning), but similar representation. The MBC outcome of this comparison is therefore, equal to (0,0,0,1). The conflict in this case is a name-representation conflict of case 7. The conflict is then reconsidered to establish the fact that the representations are of different concepts.

The second MBC group is that of meaning equivalence. This group results in creating categories of concepts with similar meaning irrespective of concept name or representation. Cases in this group include identical NIETs, i.e., case 0; and multiple concept representations, i.e., cases 1, 8, and 9. In the example provided, representations (2,2,6: Area-of-interest) and (2,3,8: Area-of-interest) exemplify case 1 of this MBC group.

The third MBC group is inter-class representation conflict, and it refers to similar representations of different concepts. This group includes cases 2, 4, and 6; in which two representations are similar, but their intensions and/or extensions differ. This group is exemplified by representations (2,2,3: Funding) and (2,3,14: Funding), where the representations and their names are identical, but the represented concepts are different (i.e., case 2 in which intensions vary).

The fourth group includes all representations that do not fall in any of the previous categories. This group includes cases 11,13, and 15, in which different representations have different names and stand for different concepts. Representations (1,1,6: Contract) and (2,3,7: Publication) are an example of this group (i.e., case 16).

10.4.3 Concept integration

Once concepts are classified according to meaning, further modelling operations can be performed. Concept integration is the main process in conceptual modelling, and it is comprised of techniques defined within the three perspectives of the proposed approach.

10.4.3.1 The semantic perspective

As a result of NIET concept definitions and MBC classification, a list of concept classes and representations for each class is composed. The next step would then be to draw the semantic relations between such classes. In the following, the DIS example will be examined to determine existing semantic relations. It is necessary to state at this point, that the examination of concepts for the purpose of determining semantic relations is better achieved in a similar fashion to that of MBC, i.e., by examining a single concept against the remaining ones, and so on. However, and for briefness reasons, an alternative approach is followed, namely by targeting related concepts.

Semantic independence (SR1)

Semantic independence is a relation which indicates lack of semantic connection between concepts. It is preferable to identify concepts that are not related at first so as to rule out the possibility of establishing semantic inclusion, and convergence between such concepts; thus reducing the number of comparisons which would have to be performed. Therefore, managing complexity is one of the main two objectives of this relation. The other being the identification of possible SR11.

Semantic inclusion (SR2)

Semantic inclusion is a relation between two concepts in which one concept defines the other. In the DIS example, semantic inclusion is accounted for by examining a concept intension. More specifically, a representation whose intension includes another concept name is an indication of the presence of the relation of semantic inclusion. Out of NIET definitions, a number of SRs are ascertained, and they are illustrated in figures 10.11, 10.12, 10.13, and 10.14.

Note that an SR2 has an advantage over an ISA in that the identification of an SR2 relation relies on user defined concept meanings, thus avoiding the possibility of inaccurate abstractions. This advantage may not be obvious when ascertaining that a "Student" ISA "Person". However, this is not always the case. In a manufacturing industry, for example, it is not so obvious, from a designer point of view, that the contraption "P123" ISA "PSQ". Moreover, the definition of intension, as exemplified in this chapter, provides sound methods for identifying SR2s, and all semantic relations for that matter.

Semantic intersection (SR3)

This semantic relation is established between overlapping concepts. Beside overlap indication, an SR3 provides the direction to concept update by indicating the direction of the overlap. The identification of this relation is provided by concept extension. A concept extending over concepts which it does not define, indicates a semantic intersection between the concept and each of the concepts within its extension.

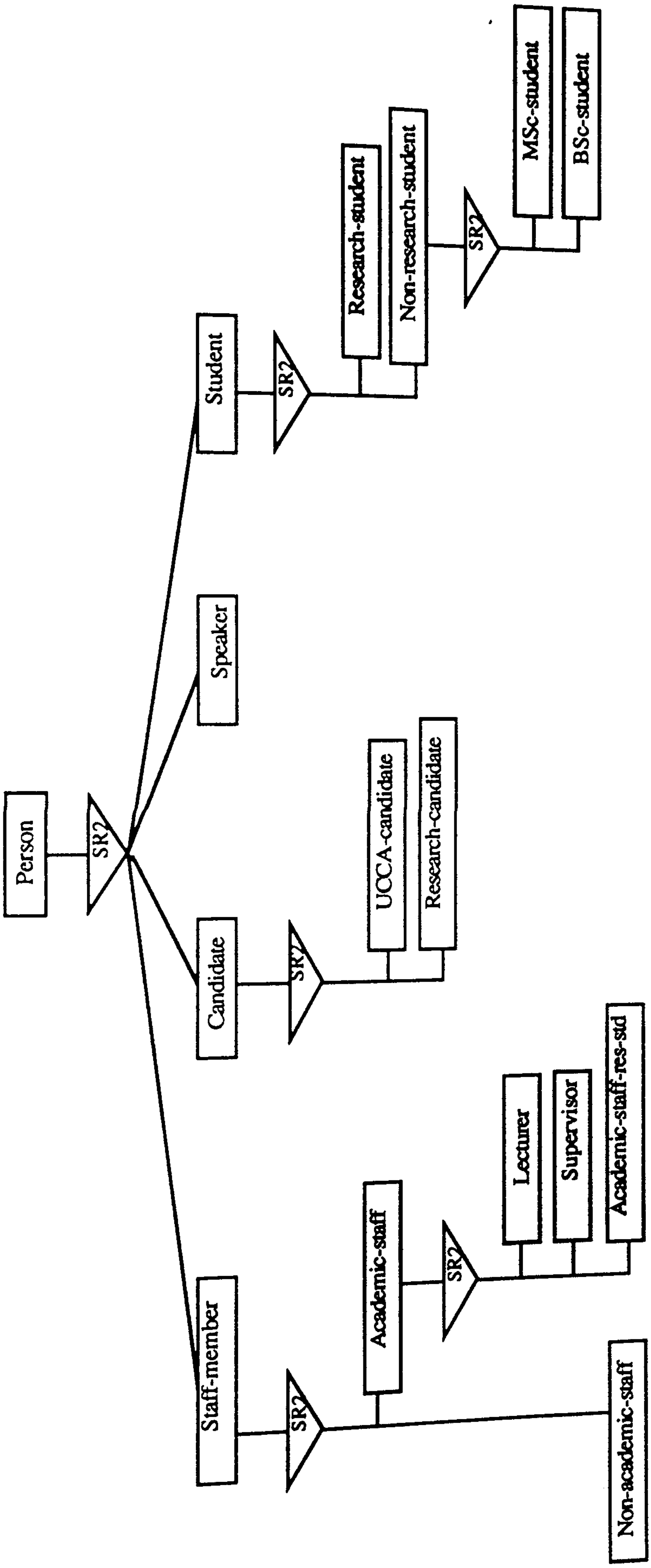


Figure 10.11 "Person" inclusion abstraction

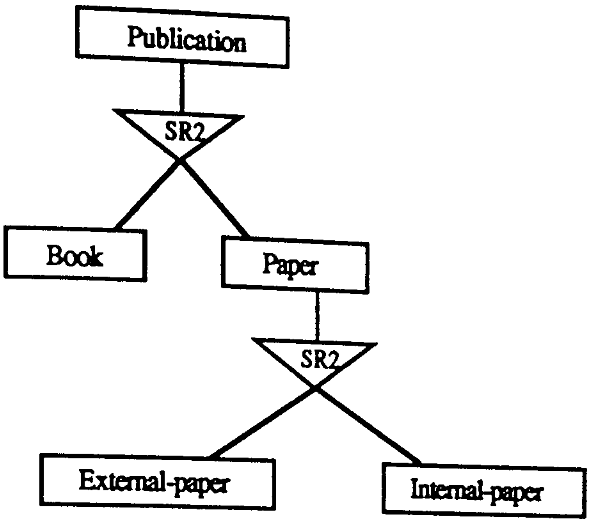


Figure 10.12 "Publication" inclusion abstraction

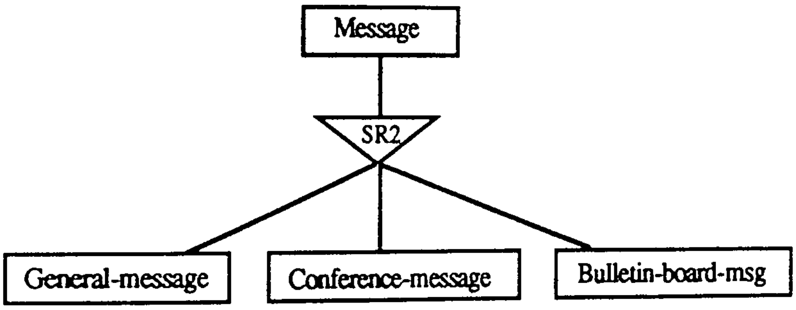


Figure 10.13 "Message" inclusion abstraction

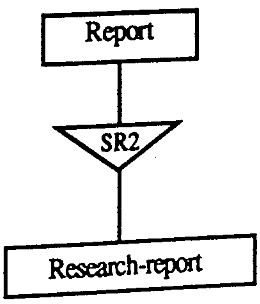


Figure 10.14 "Report" inclusion abstraction

Therefore, in the DIS example, the concepts "Lecturer" and "Research-student" intersect in the direction of "Lecturer", i.e. a "Lecturer" can be (and not is a) "Research-student". The implication is that update in "Research-student" should prompt a check on "Lecturer". The other case of semantic intersection is the concepts "Lecturer" <<SR3> "Supervisor".

Prevalent convergence (SR31)

This semantic relation holds between two prevalently converging concepts. For the DIS example, the concept "Res-std-acad-staff" expresses an SR31 between "Research-student" and "Academic-staff" (figure 10.14). This relation implies an inheritance of attributes from the defining concepts to the new concept. Note, here, that the convergence was defined independently instead of being a simple semantic intersection of the two defining concepts, because there are further operations defined on the convergence itself, e.g., a relationship with another concept.

Semantic convergence is identified by the presence of two concepts in the intersection of a third. SR31s for the DIS example are illustrated in figure 10.15.

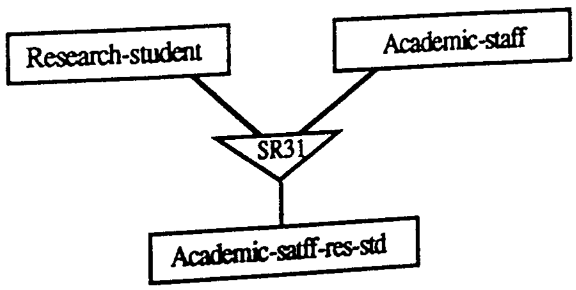


Figure 10.15 An example of an SR31

Atypical convergence (SR11)

This relation is the same as SR31 except that the defining concepts are not semantically related. There are no concepts, in the example, which satisfy the definition of this relation. This is most likely because of the relative level of homogeneity of concepts within an academic department.

10.4.3.2 The application perspective

After defining semantic relations, the second category of relations is defined according to application specifications. This is a familiar modelling task in which concepts are related according to the ERM approach. However, the extensions provided in chapter 8 introduced additional relationship constructs which are both beneficial and critical to the integration of this step with other design phases.

An initial diagrammatic representation of the DIS application perspective is illustrated in figure 10.16. In the figure, for reasons of simplicity, relationships are indicated by simple lines connecting concepts. As it is clear in the diagrammatic illustration, the only relations depicted are the relationships between concepts as defined by the application perspective. In addition, and to simplify further, the names and properties of the relationships are not included.

What is more important than the diagrammatic representation of the DIS is the schematic specification of relationships between concepts. Such specification is an added advantage of SECOM implementation. This aspect will be demonstrated by analysing specific relationships.

Figure 10.16 shows a relationship between the concepts "Supervisor" and "Student", which will be assumed to have the following properties. A supervisor is limited to

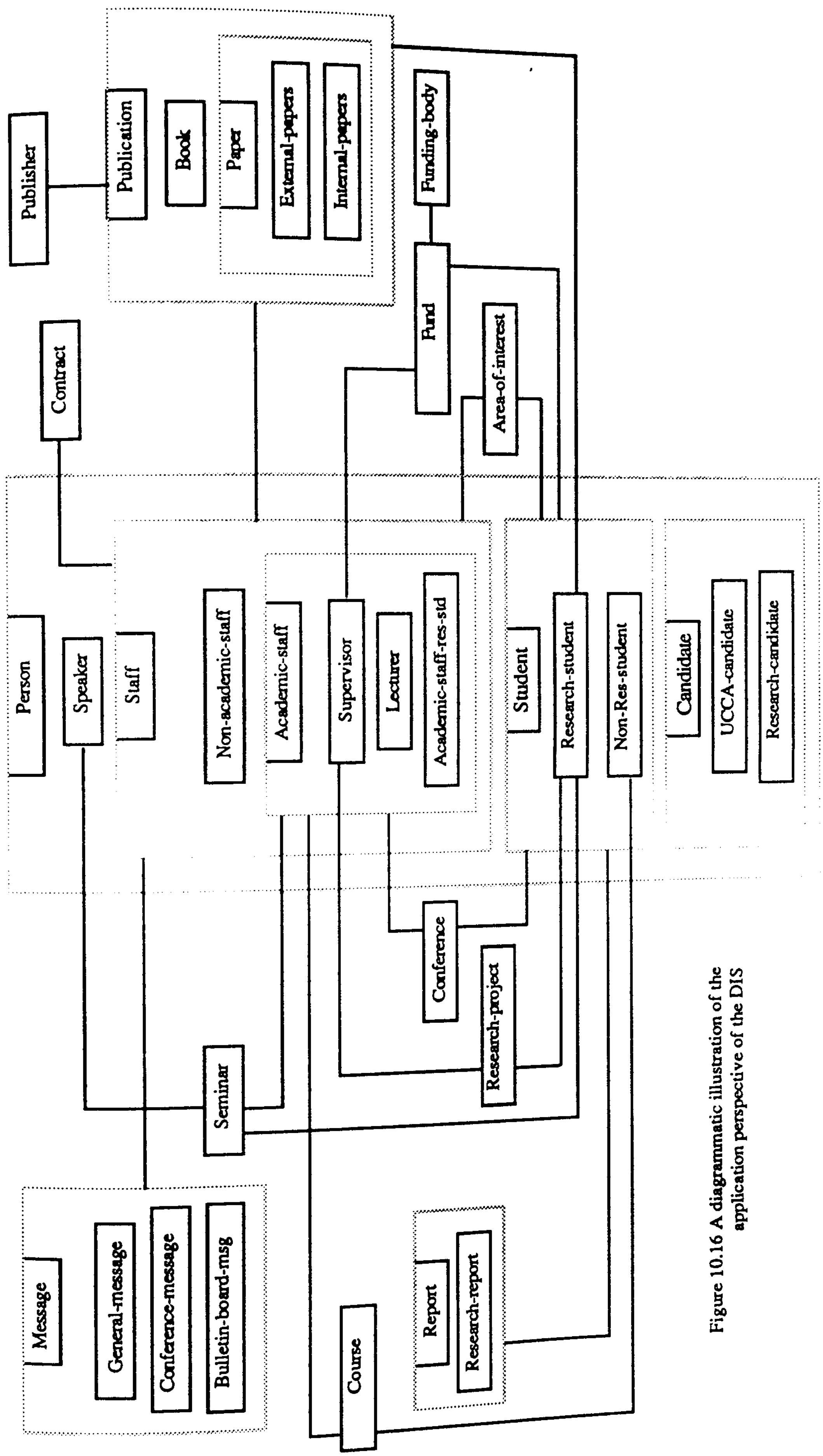


Figure 10.16 A diagrammatic illustration of the application perspective of the DIS

supervising between 0 and 5 students. Because every student has to have a supervisor, "Student" participation is functional. As such, the relationship "Supervision" is explicitly indicated by its extension in the following fashion:

E(Supervision)= {Supervisor: [P] (0-5) - (1) [F] :Student}

Another, and more elaborate relationship is that of "Places" between "Research-student" and "Supervisor" on the one hand, and "Message" on the other. This relationship can take any number of definitions according to the properties of the relationship. For example,

E1(Places)= {SupervisorlResearch-student: [P] (0-N) - (0-N) [P]: Message}, or

E2(Places)= {(Supervisor: [P] (0-N) - (0-N) [P]: Message), Research-student: [P] (1) - (0-N) [P]: Message)}

In E1, a supervisor or a research-student is permitted to place any number of messages at any period in time; while in E2 the relationship is different from that of E1 since a research-student can only place only one message at any time period, thus the alteration in the relationship definition.

10.4.3.3 The organization perspective

This perspective of the SECOM deals with abstraction relations between concepts rising out of organization structure constraints over information representation. Each of such relations is identified for the DIS example.

Inclusion abstractions (AR11)

The definition of this abstraction relation is strictly utilized for controlling the consistency of abstraction representation among semantic inclusion relations at different organizational levels. According to effective organization and information systems design, less detailed and general concepts are represented at a higher organization level; while more detailed, specific concepts are represented at lower levels .

In the DIS example, there are two situations in which more general concepts are represented at lower levels than the levels in which their refiners are. The first is the representation of (1,1,4: Research-student) at the first level, which is a more specific than "Student" in the second or third levels (i.e., 2,2,2 and 3,1,2). The second is the representation of (1,1,8: Message) at the first level again, and the representation of (2,3,12: Conference-msg) at the second level.

The two contexts illustrate an inconsistency in abstraction representation. In other words, the "Head-of-Department" should rarely be interested in the specification of the concept "Student", and the subsequent additional specification details, when "Student" itself is of interest to users at a lower level. The only logical explanation to this situation, is that the "Head-of-Department" is only interested in a "Research-student" information which is not related to the detail provided by the specification of the concept itself, but to a type of "Student" which in this case happens to be "Research-student". The same explanation applies to the other case.

Therefore, an AR11 is drawn between the concept "Student" and representation (1,1,4: Research-student) to indicate the representation is bound by the attributes of the more general concept. On an implementation level, the relation implies that "Research-student" representation is limited to "Student" attributes of the concept "Research-student"; after all, the concept "Research-student" is the concept "Student" plus additional "Research-student" specific attributes. Therefore, as far as "Student" is concerned, an AR11 is without any implication, because the relation does not imply any changes in "Student", and "Student" was only pointed out because it is the definer which instigated the relation.

The second representation situation, i.e., "Message" and "Conference-msg", is dealt with is the same way as above.

Convergence abstractions (AR12)

The other abstraction relation defined over a semantic relation is an AR12, and it is defined over concept convergence. In the example, one case of semantic convergence falls under the definition of an AR12, and it is (1,1,12: Academic-staff-Res-std) which is the convergence of "Research-student" and "Academic-staff". The convergence is represented at a higher level in which one of its definers is represented (i.e., "Research-student"), which results in representation inconsistency.

Therefore, the detail of representation (1,1,12) is limited to that of the common attributes of the concepts "Academic-staff" and "Research-student". The common attributes would amount to a representation of "Person" who is both a "Research-student" and an "Academic-staff". This would result in disregarding details attached to the original concept, including the details which prompted the need to define "Academic-staff-Res-std" in the first place.

The abstraction relation in this case is drawn between "Academic-staff-Res-std" on the one hand, and "Person" on the other. Note, again, that the relation implies that the representation is made less detailed, so that hierarchy-abstraction consistency is maintained.

Partition abstractions (AR21)

This relation is defined over subsets of concept classes where the such subsets are defined based on the value of one or more attribute. There are no concepts defined according to this abstraction relation, and therefore, A21 is not applicable.

AR22: Summary abstractions

This abstraction relation is defined over summary concepts. The implication of this relation is that the dependent concept, i.e., which the summary concept is represented at an equal or higher level than one in which the dependent concept is represented.

In the example, an AR22 is drawn between the concept "Term-details" and each of "Funding", "Students", and "Projects". The relation indicates the dependency of the concept "Term-details" and each of the concepts contributing to the determination of its attributes. The update implied by the relation is from the independent concept to the dependent one.

AR23: Relationship abstractions

A relationship abstraction is an abstract concept of two concepts and a relationship. Each of the concepts participating in the abstraction is, by definition, a more specific concept than the abstraction itself. Therefore, a representation of a relationship abstraction which contradicts the hierarchy-abstraction correspondence calls for an AR23 application.

In the DIS example, the concept "Seminar-speaker" is an abstraction of the relationship "Speak" between the concepts "Speaker" and "Seminar". The abstraction is represented at the third hierarchical level, i.e. (3,1,10: Seminar-speaker), and at the same time, "Seminar" which is a constituent of the abstraction is represented at a higher level(s), i.e., (1,1,10 and 2,3,10:Seminar).

Therefore, the representations of "Seminar" at the higher levels are substituted with a representation of "Seminar-speaker". In other words, an AR23 is drawn from the concept "Seminar" to the abstraction itself, to indicate the abstraction of the representation. However, and because the supervisor view includes both "Seminar" and "Seminar-speaker", the representation of "Seminar" is removed from the view, leaving "Seminar-speaker" to provide abstract information about both "Seminar" and "Speaker".

Representation abstractions (AR3)

The final abstraction relation imposes a constraint on the level of information representation. For example, the representation of "Research-student" in the first level, i.e., (1,1,4) is limited to similar to or less than the number of attributes representing the same concept at a lower level, i.e., (2,3,3). This, again, is in accordance with the hierarchy-abstraction convention of not representing more detail at a higher level than a lower one.

10.5 Conclusion

This chapter presented an example of an application of SECOM. Though limited by the context in which it is presented, the application reflected the effectiveness and comprehensiveness of the proposed approach.

The large extent of semantic inclusion is evident in the way the approach leaves no room for conflict over concept meaning. The fact that concepts can be viewed differently by different users has no effect on the integrity of the database. This is a consequence of understanding the impact of semantic theory on database modelling.

In addition, the application of the suggested approach illustrated the benefit of the examining the structure of organizations, and utilizing organization concepts for the purpose of improving information relevance.

It is clear that the application of the modelling approach, presented in this chapter, does not carry through the database modelling to an implementation level. This, however, does not stand in the way of evaluating the effectiveness of SECOM; especially when considering that methods of concept definition, representation, and integration are usually discussed - in the literature - in isolation of implementation specifications. The translation of the modelling constructs - developed in suggested approach - into implementation specific constructs is seen to be immediate.

It is also evident from the example provided that the translation of SECOM processes into an integrated and automated tool is a matter of direct translation into an appropriate environment for such tool, e.g., a programming environment.

Chapter 11

Conclusions and recommendations

In this thesis a new approach to modelling organization databases has been presented. The approach emphasized the importance of examining all issues of relevance to database design and utilization. Such issues include conceptual modelling, and semantic and organization theories.

Semantic theory was explored to establish basic concept definitions. Such definitions provided a system of concept definition, classification, and integration. The advantages of relying upon sound semantic foundations enabled the separation of concept names and representations from concept meaning. By depending on concept content - instead of form - more intrinsic abstractions can be formed. Such abstractions are more reliable than conventional data abstractions because they relate concepts which lack similarity in form, i.e., lack common attributes. As a result, concepts can be represented in many forms according to users' view of such concepts.

The advantages of relying upon sound semantic theory for conceptual modelling include the following: ensuring database integrity as a result of improved semantic relations; facilitating concept multiple representations as a consequence to separating form from content; improved database implementation which is a direct result of removing representation constraints from the implementation to the modelling phase of the design; and the promotion of data independence.

The second perspective presented a method for concept relationships based on their use in the application environment. The method emphasized the importance of the explicit indication of relationship constraints. The main advantage of this aspect of the modelling approach is that it improves database consistency because it relies on semantic relationship definition. Therefore, a relationship includes within its semantic definition the constraints and field of the relationship application.

The third perspective introduced a new modelling dimension in which organization structure issues were incorporated. In this perspective, concept representations were evaluated in relation to the level of organization structure in which such representations are required. In addition, relations were defined to control the level of generality and extent of detail represented at various hierarchical levels. By maintaining such control, information relevance is further enhanced since concept meaning and representations are hierarchically arranged in accordance with the nature of information abstractions implied by the organization hierarchy.

The strength of the proposed approach lies in its holistic nature. Its investigation of the relevant modelling concepts enabled the presentation of a comprehensive solution to database modelling problems. SECOM addressed the interrelationships between database design problems. The examination of semantic and organization theories led to establishing organization oriented conceptual abstractions.

It is evident by examining the steps in the modelling approach, that the processes involved were created with a view to the prospect of developing an automated conceptual modelling tool. As such, the development of an automated tool is a matter of translating the processes involved into a programming environment.

The contribution of this research to conceptual modelling in specific, and computer utilization in general, is manifested in the development of new concepts and tools for database design. The concepts presented form solid grounds for further database development.

In addition, the research contributed in the way of bringing database concepts into line with organization concepts; thus narrowing the gap between computer systems and their users.

It is clear, however, that the approach presented in this thesis is not detailed enough to enable its immediate implementation. There are a number of reasons for its being so, and the most important of which is that bringing the approach to the level of detail necessary for its implementation requires extending the scope of analysis to areas beyond the limits of this research (e.g., database physical design). However, it is felt that the example of its application, provided in chapter 10, gives a strong indication of the viability and applicability of the suggested approach.

The new direction to examining conceptual modelling, suggested and followed in this thesis, is seen as most appropriate for developing database design methodologies. The examination of the database environment, including all issues of relevance to the storage and utilization of data, is the only way to securing effective database design.

Future database design developments can benefit from and build on the concepts presented in this thesis. There is a need, for example, to further the examination into the relationship between language and organization, and the evaluation - in terms of quality and quantity - of the impact of the nature of work on data in general and information representation in general.

References

Abiteboul, S. & Hull, R. (1984), "Towards a formal semantic database model", *Proceedings of the 3RD Scandanavian Research Seminar on Information Modelling and Semantics*, University of Tempero: Finland, pp.149-175.

Abrial, J. R. (1974), "Data semantics", in Klimbie J. W. & Koffeman, K. L. (eds), *Data Base Management*, North-Holland.

Allan, K. (1986), *Linguistic meaning (vol 1)*, Routledge & Kegan Paul.

ANSI/X3/SPARC study group on database management systems (1975), "Interim report", *FDT Bulletin of ACM SIGMOD* 7(2).

Anthony, R. N. (1965), *Planning and control systems: A framework for analysis*, Graduate School of Business Administration, Harvard University.

Avison, D. E. & Wood-Harper, A. T. (1990), *Multiview: an exploration in information systems development*, Information Systems Series.

Bachman, C. W. (1969), "Data structure diagrams", *Data Base* **1**(2), pp.4-10.

Bachman, C. W. & DAYA, M. (1977), "The role concept in data models", *Proceedings of the 3rd International Conference on Very Large Data Bases*, Tokyo, pp.464-476.

Batini, C. (edt) (1988) "Proceedings of the Seventh International Conference on Entity-Relationship Approach", *Proceedings of the 7th International Conference on Entity-Relationship Approach,* Chicago: USA, IEEE Computer Science Press.

Battista, G. D. & Batini, C. (1988), "Design of statistical databases: A methodology for the conceptual step", *Information Systems* **13**(4), pp.407-422.

Bealer, G. (1982), *Quality and concept*, Oxford: Clarendon Press.

Beer, S. (1979), *The heart of the enterprise*, Wiley.

Boshel, A. (1988), *Department of Computer Science and Applied Mathematics Information Systems Report*, MSc. Report, Department of Computer Science and Applied Mathematics, Birmingham, UK.

Brown, W. (1971), *Organizations*, Heinemann.

Bunge, M (1967), *Scientific research I: The search for a system*, Spriner-Verlag.

Bunge, M. (1974a), *Treatise on basic philosophy, Volume I, Semantics I: Sense and Reference*, D. Reidel Publishing Company.

Bunge, M. (1974b), *Treatise on basic philosophy, Volume II, Semantics II: Interpretation and Truth*, D. Reidel Publishing Company.

Caraswell, J. jr. & Navathe, S. (1987), "SA-ER: A methodology that links structured analysis and entity-relationship modelling for database design", in March, S. (edt), *Proceedings of the 6th International Conference on Entity-Relationship Approach*, Rome: Italy USA, pp.381-97.

Ceri, C. (edt) (1983), *Methodology and tools for database design*, North-Holland.

Ceri, S. & Navathe, S. B. (1988), "Conceptual and logical database design", Tutorial paper, *International Conference on Very Large Databases,* Long Beach: USA.

Chen, P. P. (1976), "The Entity-Relationship Model- Towards a unified view of data", *ACM Transactions on Data Base Systems* **1**(1), pp.9-36.

Chen, P. P. (1977), "The Entity-Relationship Model - A basis for the enterprise view of data", *Proceedings of the AFIPS Conference*, AFIPS Press, Arlington: USA, pp.77-84.

Chen, P. P. (edt) (1980) "Entity-Relationship Approach to Systems Analysis and Design", *Proceedings of the 1st International Conference on Entity-Relationship Approach*, L.A.: USA, North-Holland.

Chen, P. P. (edt) (1981) "Entity-Relationship Approach to Information Modelling and Analysis", *Proceedings of the 2nd International Conference on Entity-Relationship Approach,* Washington: USA, Elsevier Science.

Chen P.P. (1983), "English sentence structure and Entity-Relationship diagrams", *Information Science (USA)* **29**(2-3), 127-49.

Cherry, C. (1957), *On human communication*, Wiley.

Child, J. (1984), *Organization, A guide to problems and practice* (2nd edition), Harper & Row.

Cibora, C. U. (1981), "Information systems and transactions architecture", *International Journal of Policy and Information Systems* **5**(4), pp.305-324.

Clemons, E. K. (1985), "Data models and the ANSI/SPARC architecture", in Yao, S. B. (edt), *Principles of database design (Vol I)*, Prentice-Hall.

CODASYL (1971), *CODASYL Data base Task Group Report REPORT*, New York: ACM.

Codd, E. F. (1970), "A relational model for large shared data banks", *Communications of the ACM* **13**(6), pp.377-87.

Codd, E. F. (1979), "Extending the database relational model to capture more meaning", *ACM Transactions on Data Base Systems* **4**(4), pp.397-434.

Coseriu, E. (1970), *Sprache, Strukturen und Funktionen*, XII Aufsatze, Tubigen.

Date, C. J. (1983), *Database: A primer*, Addison-Wesley.

Davis, R. C. (1951), *The fundamentals of top management*, Harper.

Davis, C., Jajodia, S., Ng, P. & Yeh, R. (eds) (1983) "Entity-Relationship Approach to Software Engineering", *Proceedings of the 3rd International Conference on Entity-Relationship Approach*, Anaheim: USA, North-Holland.

De, P., Haseman, W. D. & Ho, S. Y. (1981), "Four-schema approach : An extended model for database architecture", *Information Systems (GB)* **6**(2), pp.117-24.

De, P., Sen, A. & Gudes, E. (1982), "A new model for data base abstraction", *Information Systems* **7**(1), pp.1-12.

dos Santos, C. S. & Neuhold, E.J. & Furtado, A. L. (1980), "A data type approach to the entity-relationship model", In Chen, P. P. (edt), *Entity-Relationship approach to systems analysis and design*, North-Holland, pp. 103-119.

Drucker, P. F. (1973), *Management: tools-responsibilities-Practices*, Harper & Raw.

Elliott, D. (1980), "The organization as a system", in Salaman, G. & Thompson, K. (eds), *Control and ideology in organizations*, Open University Press, pp. 85-102.

Elmasri, R. (1980), *On the design, use, and integration of data models*, Ph.D. dissertation, Stanford University: Palo Alto, USA.

Elmasri, R., Weeldreyer, J. & Hevner, A. (1985), "The category concept: An extension to the entity-relationship model", *Data and Knowledge Engineering* **1**, pp.75-116.

Elmasri, R. & Navathe, S. B. (1989), *Fundamentals of database systems*, Benjamin/Cummings.

Feldman, P. & Miller, D. (1986), "Entity model clustering: Structuring a data model by abstraction", *The Computer Journal* 29(4), pp.348-60.

Fleming, C. C. & von Halle, B. (1989), *Handbook of relational database design*, Addison-Wesley.

Follnus, D. P., Madnick, S. E. & Shultzmann, H. B. (1974), "Virtual information in database systems", *FDT SIGFIDET* 6.

Ford, R. C., Armandi, B. R. & Heaton, C. P. (1988), *Organization theory, an integrative approach*, Hrpper & Row.

George, F. H. (1964), *Semantics*, The English University Press Ltd.

Gerth, H. H. & Mills, C. W. (eds&trns) (1946), *Max Weber: Essays in sociology*, Oxford University Press.

Goldstein, K. & Scheerer, G. (1939), *The organism*, New York: American Book Co.

Gordon, R. A. (1968), *Business leadership in large corporation*, University of California Press.

Gross, B. M. (1968), *Organizations and their managing*, Free Press.

Hammer, M. M. & Mcleod, D. J. (1978), "The semantic data model: A modelling mechanism for database applications", *Proceedings of the ACM SIGMOD International Conference on the Management of Data*, Austin, Texas.

Hammer, M. M. & Mcleod, D. J. (1981), "Database Description with SDM: A semantic data Model", *ACM Transactions on Database Systems* 6:3.

Harrison, E. F. (1978), *Management and organizations*, Houghton Miffin.

Harvey, O. J., Hunt, D. E. & Schroder, H. M. (1961), *Conceptual systems and personality organization*, Wiley.

Hautamaki, A. (1984), "Tutorial on philosophical approaches related to conceptual modelling", *Proceedings of the Third Scandinavian Research Seminar on Information Modelling and Data Base Management*, Tampere: Finland, pp.15-33.

Hawryszkiewycz, I. T. (1984), *Database analysis and design*, Science Research Associates.

Hebrial, G. (1986), "A model of summaries for very large databases", in Cubitt, R. *et al.* (eds), *Proceedings of the Third International Workshop on Statistical and Scientific Database Management*, Luxemborg: Grand Duchy of Luxemborg, pp.143-151.

Howard, D. V. (1983), *Cognitive psychology*, Macmillan.

IBM Business Systems Planning (1984), *Information systems planning guide*, International Business Machines.

Jansson, C. G., (1985), "A framework for representation", in Senadas, A. *et al.*, *Information Systems: Theoretical and formal aspects*, North-Holland.

Jaques, E. (1965), "Speculation concerning level of capacity", in Brown, W. & Jaques, E. (eds), *Glacier Project Papers*, Heinemann.

Jaques, E. (1976), *A general theory of bureaucracy*, Hienemann.

Jaques, E. (1978), "Levels of abstraction in mental activity", In Jaques E., Gibson R. O., & Isaac, D.J. (eds), *Levels of abstraction in Logic and human action*, Hienemann.

Jenkins, A. M. (1985), "Research methods and MIS research", in Mumford, E. *et al.* (eds), Research methods in information systems, North-Holland.

Kahn B. K. (1976), "A method for describing information required by the database design process" Proceedings of the 1976 ACM SIGMOD International Conference on Management of Data, June 1976, ACM, New York.

Kahn, B. K. (1979), *A structured logical database design methodology (Vol I & II)*, Ph.D. dissertation, The University of Michigan, Michigan, USA.

Kangassalo, H. (1982), "On the concept of concept in a conceptual schema", Report A67, Department of Mathematical Sciences, University of Tampere, Tampere, Finland.

Kangassalo, H. (1984),"On embedded views in a definitional conceptual schema", In Kangassalo, H. (edt), *Third Scandinavian Research Seminar on Information Modelling and Data Base management*, University of Tampere: Finland.

Katz, J. (1972), *Semantic theory*, Harper and Row.

Katz, D. & Kahn, R. L. (1966), *The social psychology of organizations*, Wiley.

Kauppi, R. (1967), "Einfuhrung in die theorie der begriffssysteme", Ser. A, Vol 15, Acta Universitatis Tamperensis, Tampereen yliopisto, Tampere, Finland.

Kent, W. (1977), "Entities and relationships in information", in Nijssen, G. M. (edt), *Architecture and models in data base management systems,* North-Holland, pp.67-91.

Kent, W. (1979), "Limitation of record-Oriented information models", *ACM Transactions on Data Base Systems* 4(1), pp.107-131.

Kerschberg, L. & Pacheco, J. E. S. (1975), "A functional database model", *Technical Report*, Pontificia Catholic University, Rio de Janeiro, Brazil.

Korzybski, A. (1933), *Science and Sanity*, Science Press.

Laudon, K. C. & Laudon, J. P. (1988), *Management information systems*, Macmilan publishing company.

Lee, D. T. (1985), "A cohesive methodology of integrating technical and organizational issues for information systems and database design", *Policy and Information* 9(1), pp.13-29.

Lenzerini, M. (1985), "SERM: semantic entity-relationship model", in Liu, J. (edt), *Proceedings of the 4th International Conference on Entity-Relationship Approach,* Chicago: USA, IEEE Computer Science Press, pp.270-278.

Lewis, P. V. (1980), *Organizational communication: The essence of effective management* (2nd edition), Wiley.

Litterer, J. A. (1965), *The analysis of organizations*, Wiley.

Liu, J. (edt) (1985) "Proceedings of the Fourth International Conference on Entity-Relationship Approach", *Proceedings of the 4th International Conference on Entity-Relationship Approach,* Chicago: USA, IEEE Computer Science Press.

Lundgren, Earl F. (1974), *Organizational management: systems and process,* Harper & Raw.

Lyons, J. (1977), *Semantics*, Vol 1, Cambridge University Press.

Malmorg, E. (1986), "On the semantics of aggregated data", *Proceeding of the Third International Workshop on Statistical and Scientific Database Management.* Luxemborg: Grand Duchy of Luxemborg, pp.152-158.

March, S. (edt) (1987) "Proceedings of the Sixth International Conference on Entity-Relationship Approach", *Proceedings of the 6th International Conference on Entity-Relationship Approach,* Rome: Italy USA.

Martin, J. (1982), *Strategic data-planning methodologies*, Prentice-Hall.

Massie, Joseph L. & King, William R. (1971), *Essentials of management* (2nd edition), Prentice-Hall.

Mattison, R. M. (1987), "The organizational and information context of large business system integration project", *SIGMOD Record* **16** (2), pp.20-36.

McFadden, F. R. (1985), *Data base management*, Benjamin/Cummings.

Meltzer, H. S. (1969), *Database concepts and architecture for database systems*, IBM report to SHARE Information Systems Research Project.

Merton, R. K. (1957), *Social theory and social structure* (revised edition), Gelncoe.

Meurer, T. (1980), "Solving the mystery of data base design", *Computer world* **17**, pp.48-51.

Mill, J. (1970), *A System of Logic* (8th ed), Longman.

Molina, F. W. (1979), "A practical database design methodology", *Data Base* **11**(1), pp.3-11.

Mylopoulos, J. & Wong, H. K. T. (1980), "Some futures of the TAXIS data model", *Proceedings of the 6th International Conference on Very Large Data Bases*, Montereal, Canada, pp.399-410.

Navathe, S. B. & Schkolnick, M. (1978), "View representation in logical database design", *ACM SIGMOD International Conference on the Management of Data*, Austin: USA, pp.144-56.

Navathe, S. B. & Gadgil, S. G. (1982), "A methodology for view integration in logical database design", *Proceedings of the 8th Conference on Very Large Data Bases*, Mexico City, Mexico, pp.142-164.

Ng, P. A. (1981), "Further analysis of the entity-relationship approach to Database Design", *IEEE Transactions on Software Engineering* **SE-7**(1), pp.85-99.

Nord, W. R. (1976), *Concepts and controversy in organizational behaviour* (2nd edition), Pacific Palisades, California: Goodyear.

Ogden, C. K. & Richards, I. A. (1938), *The meaning of meaning*, Harcourt Brace.

Pace, R. W. (1983), *Organizational communication, foundations for human resource development,* Prentice-Hall.

Paterson, T. T. (1972), *Job Evaluation*, London: Business Books.

Pfiffner, J. M. & Sherwood, F. P. (1960), *Administrative organization*, Prentice-Hall.

Rapoport, A. (1965), "What is semantics?", in Hayden, D. E. & Alworth, E. P. (eds), *Classics in semantics*, London: Vision Press.

Robbins, S. P. (1983), *Organization theory: The structure and design of organizations,* Prentice-Hall.

Rosch, E. H. (1978), "Principles of categorisation", in Rosch, E, H. & Lloyd, B. B. (eds), *Cognition and categorisation*, Lawrence Erlbaum.

Rosch, E. H. (1973), "On the internal structure of conceptual and semantic categories", in Moore, T. E. (edt), *Cognitive development and the acquisition of language,* Academic Press.

Rosch, E. & Mervis, C. B. (1975), "Family resemblance: Studies in the internal structure of categories", *Cognitive Psychology* **7**, pp.573-605.

Roussopoulos, N. & Yeh, R. T. (1984), "An adaptive methodology for database design", *Computer* **17**(5) pp.64-80.

Rowbottom, R. & Billis, D. (1977), "The stratification of work and organizational design", *Human Relations* **30** (1), pp.53-76.

Rowbottom, R. & Billis, D. (1987), *Organization design: The work-levels approach,* Gower.

Rowbottom, R. & Billis, D. (1989), "Levels of work: New application to management in large organizations", *Journal of Applied Systems Analysis* **16**, pp.19-34.

Scanlan, B. K & Keys, J. B. (1979), *Management and organizational behaviour* (2nd edition), Prentice-Hall.

Scarrott, G. G. (1985), "Information, the blood of organisation", *The Computer Journal* **28**(3), pp.203-205.

Scheuermann, P., Schiffner, G. & Weber, H. (1980), "Abstraction capabilities and invariant properties modelling with the Entity-Relationship approach", in Chen P. P. (edt), *Entity-Relationship Approach to Systems Analysis and Design*, North-Holland.

Schiffner, G. & Scheuermann, P. (1979), "Multiple views and abstractions with an Extended-Entity-Relationship model", *Computer Languages* **4**, pp.139-154.

Schneider, A. E., Donaghy, W. C. & Newman, P. J. (1975), *Organizational communication*, McGraw-Hill.

Schreider, A. Jr. (1975), *Equality, resemblance and order*, Mir Publishers: Moscow.

Scott, D. (1967), "Existence and subscription in formal logic", in Schoenman, R. (edt), *Philosopher of the century: Essays in honour of Bertrand Russell*, London: Allen & Unwin.

Sen, A. (1979), *On the theory of logical data base design for a decision support system*, Ph.D. dissertation, The Pennsylvania State University: USA.

Sen, A. (1982), "Logical data base design: A management-oriented approach", *Information & Management* **5**, pp.77-85.

Senko, M. E., Altman, E. B., Astrahan M. M. & Fehder P. L. (1973), "Data structures and accessing in database systems", *IBM Systems Journal* **12**, pp.30-93.

Shahabuddin, S. (1987), "MIS: Do we still need it?", *International Journal of Policy and Information Systems* **11**(2), pp.1-8.

Smith, J. M. & Smith, D. C. P. (1977a), "Database abstractions: Aggregation and generalization", *ACM Transactions on Data Base Systems* **2**(2), pp.105-133.

Smith, J. M. & Smith, D. C. P. (1977b), "Database abstractions: Aggregation", *communications of the ACM* **20**(6), pp.405-413.

Smith, J. M. & Smith, D. C. P. (1978), "Principles of database design", *New York University Symposium on Database Design*, NY., USA, pp.35-50.

Spaccapietra, S. (edt) (1986) "Proceedings of the Fifth International Conference on Entity-Relationship Approach", *Proceedings of the 5th International Conference on Entity-Relationship Approach*, Dijon: France.

Su, S. Y. W. (1983), "SAM*: A semantic association model for corporate and scientific statistical databases", *Information Science* **29**, pp.151-199.

Sundgren, B. (1978), "Database design in theory and practice- Towards an integrated methodology", *Proceedings of the 4th International Conference on Very Large Data Bases*, Berlin, Germany, pp.3-18.

Taylor, F. W. (1947), *Scientific management,* Harper and Row.

Thompson, J. D. (1967), *Organizations in action*, McGraw-Hill.

Tucherman, L., Furtado A. & Casanova, M. A. (1983), "A pragmatic approach to structured database design", *Proceedings of the 9th International Conference on Very Large Data Bases*, Italy, pp.219-231.

Ullman, J. D. (1984), *Principles of database systems,* Computer Science Press: Potomac: USA.

Urwich, L. (1943), *The elements of administration*, London: Pitman.

Wang, S. & Shixuan, S. (1986), "Normal-Entity-Relationship Model: A new method for enterprise schema design", *IEEE International Conference on Data Engineering 86*, LA., USA, pp.108-116.

Weber, H. (1977), "D-GRAPH: A conceptual model for data bases", in Morlet, E. & Ribbens, D. (eds), *International Computing Symposium*, North-Holland, pp.497-503.

Weber, H. (1979), "Modularity in data base systems design", *Proceedings of Joint IBM/University of Newcastle upon Tyne Seminar.*

Wiederhold, G. (1977), *Database design*, McGraw-Hill.

Wittgenstein, L. (1953), *Philosophical investigations*, Blackwell.

Yadav, S. B. (1983), "Determining an organization's information requirements: A state of the art survey", *Data Base* **14**(3), pp.3-20.

Yao, S. B., Navathe, S. B. & Weldon, J. L. (1978), "An integrated approach to database design", in Yao *et al.* (eds), *Database design techniques I: requirements and logical structures*, North-Holland.