DATA BASE

A QUARTERLY PUBLICATION OF SIGBDP VOLUME 15 NUMBER 4 SUMMER 1984

A FRAMEWORK OF REFERENCE FOR DATABASE DESIGN

By Maristella Agosti and Roger G. Johnson

DATA BASE

VOLUME 15 NUMBER 4 SUMMER 1984

A quarterly publication of the
Special Interest Group on Business Data Processing
of the Association for Computing Machinery

Editorial material should be sent to the Executive Editor; subscriptions; address changes and other business communications to DATA BASE, *c/o ACM, 11 West 42nd St., New York, New York 10036*

A FRAMEWORK OF REFERENCE FOR DATABASE DESIGN

By Maristella Agosti and Roger G. Johnson

This work introduces a framework of reference for database design.

Database (DB) design is presented as a part of information system (IS) design, because the DB design process is concerned with the construction of a database as a subsystem of an IS. Then the DB design process is related to the general system development process and all existing connections are stated.

The actions which are necessary to implement a database are divided in two categories: data system design and management design. The utility of a reference model for the data system design is shown.

1. ENVIRONMENT OF DATABASE DESIGN

The environment of database design can be defined in different ways depending on which of the aspects are particularly relevant to the discussion. It is important to look at it in two perspectives:

(i) The information system design process of which the process of designing a database can be seen as a subset. The totality of objects and conditions which affect the behavior of the database system are in the information system that fully surrounds it.

(ii) The technological environment and framework of the database to be constructed. All the techniques and tools that must be available to develop, implement, and maintain a database. In this perspective the attention is on the functions the database management system (DBMS) has to be able to perform with the available computer equipment and on their performances.

1.1. First Perspective: Information

The term information system has a wide range of interpretation. An information system (IS) is seen here as an *open system*, namely a system that interacts continuously and effectively with its environment [Open Systems Group, 1981]. Main characteristics of an open system are that it is in a dynamic relationship with its environment and always in a state of change. As the environment changes the components of the open system react to adapt to the new surroundings. It has a high level of complexity; human factors and activities constitute a large part of it; it receives various inputs, transforms these inputs, and exports outputs. To operate the transformations the components of an open system need to interact, thus the system is also open internally because the interactions between components affect the system as a whole, [Kast et al, 1981] and [von Bertalanffy, 1981].

A graphical representation of the role an IS plays in an enterprise is introduced here to clarify the background to the subsequent discussion. This representation partly derives from one given in [Maggiolini, 1981].

The information system, shown in Figure 1, is seen as a system which plays a central role in the enterprise. The decision-taking body of the enterprise uses the information system to form the management decisions; then the IS makes a decision-taking body interact with a production/ transformation system. The production/ transformation system transforms resources, 'inputs,' into final products, 'outputs.' It can be a production or a commercial transformation system. The decision-taking body manages the production/ transformation system. Sensors are used to detect the reactions to the products of the enterprise in the real world. The information system changes over the time depending on the information inserted in it by the sensors and because of the needs of the decision-taking body and of the overall enterprise.

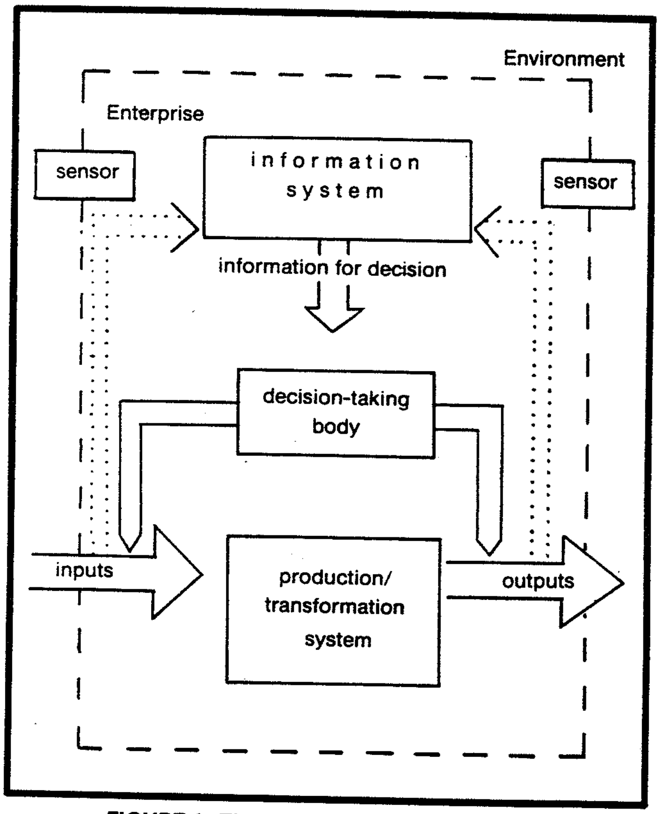

FIGURE 1. The Role of an Information System.

Maristella Agosti is with the Faculty of Statistics of University of Padova, Italy, and Roger G. Johnson is with Birkbeck College, University of London, United Kingdom. *Maristella Agosti wishes to thank the School of Mathematics Statistics and Computing of Thames Polytechnic, London, where her work was developed, and Paul Feldman for his useful comments on the paper.*

An information system study must be undertaken to decide on the feasibility of having one or more of its components computer-based or not.

If the opportunity has been shown, several subsystems of a whole information system can usually be automated. They may be completely independent of one another (i.e. completely independent components of the system) or they may be partially overlapping components. However, the information system is the environment in which each of these subsystems operates. Systems analysis usually identifies these subsystems, but different approaches to systems analysis give different outputs. The results of systems analysis are relevant here because they define the boundaries between the whole information system and each of its subsystems. Both the subsystems that are going to be automated or not must be identified.

A subsystem or group of interrelated subsystems of an information system which can be implemented by software is called an *application*.

1.2. Second Perspective: Functions

The range of different functions that are performed on the data within the IS are illustrated in Figure 2. This approach is shared by the ISO/TC97/SC5/WG5 which illustrates it in [ISO/WG5, 1983]. The objective of the database design process is to build a database which is managed through a DBMS. The DBMS can be seen as the part of an IS providing the functions required to store and retrieve data from the computers within an enterprise.

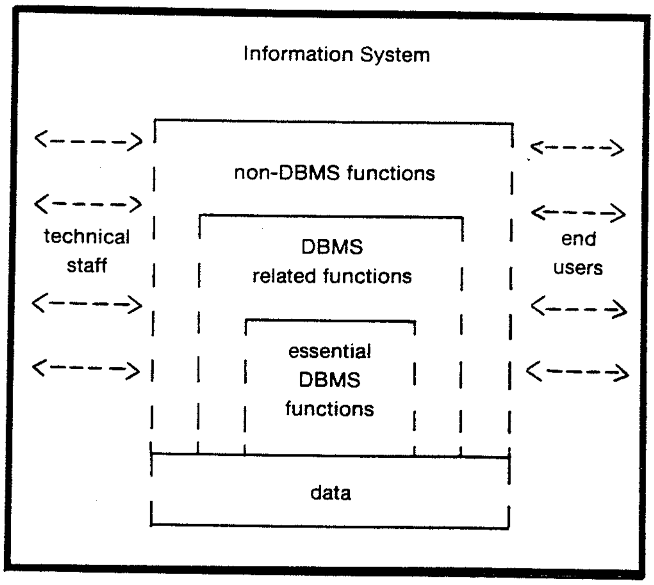

FIGURE 2. A DBMS as a Part of an Information System.

The database is a structured collection of data fundamental to an enterprise and within its information system. The data is used by different users to support the diverse operational requirements of an enterprise. To meet these requirements the database stores the data in an integrated manner and the representation is independent from the peculiarities of specific applications. Programs extract and format the data according to each application needs.

The technical staff, in Figure 2, are those users whose major concern is the analysis, design and development of the IS. The end users are the people whose major concern is the enterprise. They interact with the IS by using it as a tool to perform their work. The IS has different interfaces for these two categories of users.

Figure 2 illustrates the different sets of functions which have been identified in [ISO/WG5, 1983]: (i) the non-DBMS functions; (ii) DBMS related functions and (iii) essential DBMS functions.

(i) the non-DBMS functions. Some sets of functions which are performed within an IS are non-DBMS functions; some of them are data communications, operating systems, programming language compilers, and application programs. They constitute the greatest part of functions that are performed within the IS.

(ii) the DBMS related functions are those which facilitate the DBMS operations but are not essential to its functioning. Some of these facilities are those for database design and those for program and data conversion.

(iii) the essential DBMS functions are those which have to be supported for the operation of the DBMS. Some of these are the definition and manipulation of logical structures and substructures, physical storage structures, integrity rules, access rules, logical and physical concurrency control, logical data access, integrity control, logical and physical performance statistics collection, logical and physical audit data collection, physical data access.

Some other functions are not classified because they perform functions that are identifiable as non-DBMS and DBMS functions at the same time. An example can be view mapping.

2. DATABASE DESIGN

2.1. Database Design as a Component of Information System Design

The database design process is an integral component of the information system design process.

A range of different software components are used to implement applications and one of these is a DBMS. If the decision has been reached to develop an application using database technology, the database design process can be started. Different related applications can be arranged to share the same database. Thus the words application and applications are used interchangeably in the following discussion:

The planning process of the database environment is not pertinent to this discussion, but it is necessary to recall here, as it has been shown in different sources, for example [GAO, 1979], that the success in the introduction and use of the database technology is highly dependent on accurate planning. The planning should include comprehensive studies of user requirements, feasibility, and cost-benefit analysis; two valuable references for these topics are [Couger et al, 1982] and [Maggiolini, 1981].

The database design process is concerned with the construction of a database as a subsystem of an information system. For an understanding of the techniques and methods that are at present used as an aid to the design, it is useful to recall the system life cycle concept. The usefulness of this concept as a framework to use in the development of a system has become controversial [McCracken et al, 1982]. It is introduced here because most of present system development techniques can be related to it. Instead of *system life cycle*, it can be referred to as *system development process* to avoid suggesting that, once completed, the process has to be restarted only from Phase I, as this does not always happen in practice.

For the purpose of this discussion seven phases are distinguished in the systemm development process as in [Couger, 1982]:

Phase I	Documentation of the existing system (manual or computer-based)
Phase II	Analysis of the system to establish requirements for an improved system
Phase III	Design of the computerized system, which starts with a functional specification phase
Phase IV	Programming and procedure development
Phase V	System test and implementation
Phase VI	Operation
Phase VII	Maintenance and modification.

The actions to be taken for Phases I and II of the system development process are often grouped under the term *systems analysis.*

Database design ideally extends across Phase III and the initial part of Phase IV, but in practice there is often an overlapping with the final part of Phase II. This fact of covering only a part of Phase II and of Phase IV is a disadvantage because different database design methodologies and techniques cover different parts of a process which is not uniquely identified. For example the starting point of the process is not identified in the same place by all designers. The database design process is represented at a greater level of detail in Figure 3 and discussed in Sections 2.4 and 2.6.

In [Couger, 1982] *system development* covers the span of Phases I through III; in the context of information systems some authors use the term information systems design methodology to cover the same span. It appears from the previous discussion that an information systems design methodology covers a larger span than a database design methodology. Furthermore, many IS design methodologies do not result in the use of a database.

Software engineering is used by some authors to refer to Phase I through V, but not much attention is paid to Phase I.

The division of the system development process into the seven phases presented here is not generally used. Slightly different groupings of the activities to be undertaken during the development process are identified by authors in different areas, but it is interesting to note that the sequence . . .

- systems analysis
- systems design
- system implementation and test
- system operation and maintenance

is commonly accepted as the sequence to be followed in all the systems development projects. Examples are in [Ghezzi et al, 1982] and in [Freeman et al, 1983] for the software development process and in [Jenkins, 1981] where the development of a general systems engineering project is fully dealt with.

At present a Comparative Review of Information Systems Design Methodologies (CRIS) is being undertaken. It was suggested by the Working Group (WG) 8.1 of the International Federation for Information Processing (IFIP) and was launched by the IFIP General Assembly late in 1980 [Olle, 1982]. Two CRIS working conferences have since been held and the results of these are in [Olle et al, 1982] and [Olle et al, 1983].

2.2. *Database Design Methodologies, Techniques, and Methods*

The database design methodologies, techniques, and methods which are practiced today are part of the development techniques for computer-based systems. Their development is directly affected by the developments taking place in related areas such as programming languages and hardware technology.

It is not necessary to analyse the evolution of the system development techniques because a recent comprehensive treatment of this subject is in [Couger et al, 1982]. However, it is relevant to this discussion to identify a framework in which to insert the database design techniques and methodologies to analyse the present practice.

A database design methodology is an integrated collection of methods and techniques, which supports the complete database design process. The meaning that is ascribed here to a methodology is similar to the notion of it which is introduced in [Freeman et al, 1983] in the context of software development methodologies.

A technique in this design context provides a systematic way of doing a part of the design. When a technique is applied correctly, it will lead to a foreseeable result. A technique does not fulfill the requirements of integration and completeness that are required of a methodology. The technique can lack on formalism.

A method is an organized set of ideas that are used in doing a specific activity. It is a formal way of doing the design of only an activity.

2.3. *Database Design and Data System Design*

Database design is a process that transforms and organizes unstructured information and processing requirements concerning the application, through different intermediate representations, to a complex representation which defines schemas and functional specifications. Various documents which record the intermediate representations and the meanings of all the pertinent objects are produced during this process.

A schema is an abstract model of the subset of the information system that is being designed. At different levels of abstraction different schemas are prepared during database design.

A database is a collection of data structured in a particular way as related to a schema. The term is used to refer both to a particular instance of a collection of data and to a series of instances which are somehow related [Tsichritzis et al, 1982]. An instance of a database is used in the enterprise for different purposes and by diverse categories of users. Some categories of users are allowed to change an instance of a database through the application of some operations. The series of instances of a database, as they are transformed by operations, is usually referred to as one database.

The main objectives aimed at during the design process are:

- the definition of the schemas which correspond to different levels of abstraction within the application;
- the identification of existing constraints on the data; the implementation of tools which ensure compliance with the constraints;
- the generation of the related database;
- the implementation of functional specifications.

The schemas, the constraints on the data, the database and the functional specifications constitute a system of which they are all interdependent and interrelated components. This system is referred to as a *data system* in the rest of this work.

The collection of actions which is necessary to implement a data system is named *data system design.*

The attention of the designer of the data system is concentrated on the data and on the functions that have to be performed on the data. Before this data system is made available to the users many other activities, that are not included in the previous list of database design objectives, have to be performed. This second group of activities is often also referred to as database design, for clarity in this discussion it is referred to as *management design.*

The data stored in a database system is essentially used to make decisions at the different levels of the enterprise. Different categories of users have different targets in their work, they tend to see the data under different perspectives. The stored data concern the IS objects and their attributes: the meaning the users tend to ascribe to the data can change in the different departments of the enterprise and on the use they make of it.

For these reasons a fundamental exercise does exist in parallel with the data system design process, being the construction of a depository of all the meanings ascribed by the designer to the data kept in the database.

The data descriptions are called *metadata;* the depository of them is called a *data dictionary* (DD) and the system that manages it a *data dictionary system* (DDS). A DDS helps to support the data system design principles which are advocated in the present work.

If it is shown in the cost/benefit analysis of the application prepared before design work is commenced that it is advisable for the enterprise to organize the data through the database technology, then it is economical for the enterprise to invest much effort in the design of the database so that is can be used for diverse applications; this should reduce its overall data processing costs.

Errors made during the design process affect the application's entire life and any decisions that are made at different levels based on the data. This process therefore is of great importance for the enterprise and it is necessary to pay much attention to it. This fact explains why so much work is under development in the area.

2.4. *Data System Design Process*

The *data system design process* is usually divided into components which produce the intermediate representations. It is important to note that the division into components is conventional because this process cannot be subdivided into different natural parts as the components which are identified are dependent upon the present knowledge of the process and the present technology. Divisions are useful to handle the process and to permit the exchange of information and intermediate results between people involved in the design and between different participating sites if the IS is resident in a distributed environment.

It is not the purpose of this section to deal with a detailed description of the different components of the data system design process or the successive representations that these components produce. A scheme of these components is proposed in Figure 3 and a short explanation of the contents of each component is reported as a justification of its presence in the process. Since the application is a subsystem of the information system and the data system design is a subsystem of the application, these inclusions are represented in Figure 3.

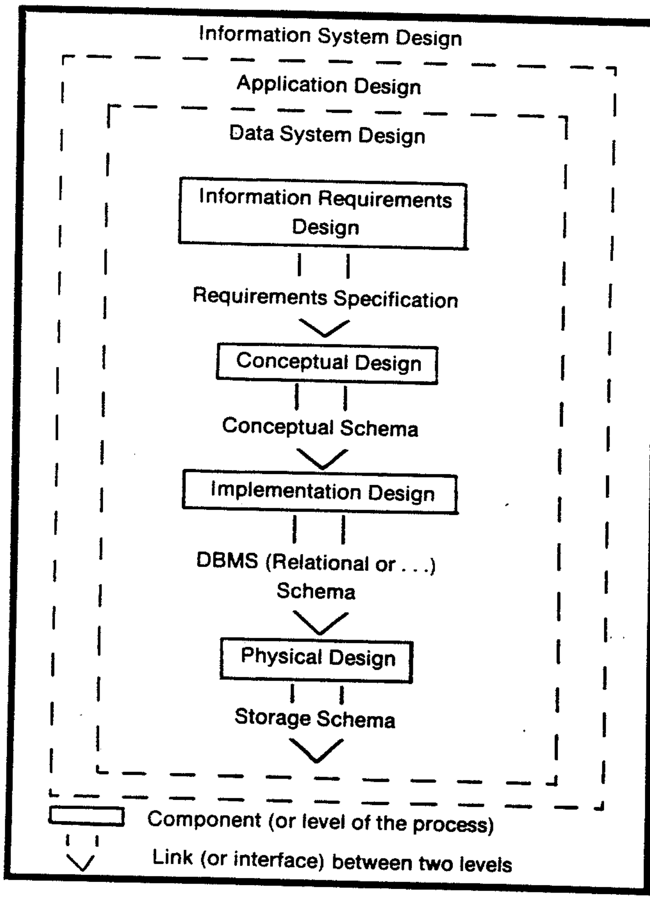

FIGURE 3. Data System Design

Most of the authors agree on a division of the process into four or five components [Lum et al, 1979], [Teorey et al, 1982] and [Yao et al, 1982b].

A division of the process into four components is:

1) Information requirements design,
2) Conceptual design,
3) Implementation design,
4) Physical design.

Some authors divide the conceptual design component into two components, instead of two main subcomponents; in this case the design process has five components.

The logical database design term has not been used, because it has so many different meanings in the literature and its use can only cause confusion. Also conceptual design can be a misleading term because it is applied to different practices. The conceptual design phase is meant to embrace the integration of all the concepts which are necessary to support the various application views of data; the conceptual design produces a conceptual view of data in which the particularities of specific views of data are resolved [Smith et al, 1982].

2.5. *Aspects of Data Models for Database Design*

It is necessary to introduce here the concept of data model because it plays a major role in the database design process.

A *data model* is a way of representing data and its interrelationships; a data model can be used to describe the data of the pertinent part of the information system, because it can be used as a tool to capture the meaning of data as related to the complete meaning of the information system.

A data model is defined [Codd, 1981] by three main sets of characteristics:

a) *data structures types:* the structures of data, the data structures types, which are supported by the model; the relational data model, for example, supports the data structure relation as defined by Codd in [Codd, 1970].
b) *operations:* the operations or inferencing rules which can be applied to occurrences of the possible data structures types which the data model supports. Using the relational data model as an example, a set of operations have been introduced in [Codd, 1970] for it; a characteristic of these operations is that the operands are relations and each application of one of the operations derives a relation from the operands.
c) *integrity constraints and rules:* the constraints and rules which have to be respected in the representation of the data to keep the database in a situation of integrity and consistency. That is, the data model supports a representation of data that is correct and does not maintain conflicting information. Besides the data in the database is accurate. These rules are sometimes expressed as insert-update-delete rules.

Data models can be divided in processable and non-processable, or semantic, data models. By *semantic data model* is not meant a specific data model occurrence but, as in [Parent, 1981], the term identifies all the data models that have been constructed to explicitly capture with their constructs the semantics of the data. The semantic data models derive their name from the fact that they are used during the conceptual design to capture the meaning, or semantics, of the application. It is still an open area of research, because no semantic data model yet defined has sufficient power of expression to balance the two important requirements of completeness and simplicity. To be successful a semantic data model will need to support the description of each pertinent aspect of the application reality and at the same time utilize only a limited number of constructs for its representation.

While the *processable data models* are the first data models that have been introduced and used, they are more machine oriented; the classical ones are the hierarchical, the network, and the relational data models. They are used in the implementation design component and they are called processable because the data representations, or schema, which reflect the data representation through one of these data models, are processable by the computer.

The abstract model that represents the data pertinent to the totality of the applications is called *conceptual schema (CS).* The *CS* is often elaborated using a semantic data model and it is elaborated during the conceptual design; the *CS* is implementation independent and it is uncommon that the *CS* is fully automatically processable.

The *implementation schema* is elaborated during the implementation design, it is implementation dependent, because it is expressed with the constructs that are present in the data model which is supported by the adopted database management system; a direct effect is that the implementation schema is always automatically processable.

When the database is centralized, the conceptual and the implementation schemas represent the same portion of the reality of the enterprise.

A *view* is an abstract model of a portion of the conceptual schema [Ullman, 1980]. That is the model of a subset of the part of the IS that is under design; this subset is the portion of the reality that a group of users needs to use most often for data management.

2.6. *Components of the Data System Design Process*

1. Information requirements design. This is the first component of the application design but it is not always considered as the first component of the data system design process, because it is the interface between the analysis process and the design process. It represents the process of mapping analysis into design. This component is part of the boundary between the application and the data system design process. Many design methods and methodologies consider the requirements specification as a prerequisite. It leads to the specification of the design requirements for the pertinent subset of the information system to meet the information requirements of the enterprise. This component has started to be called 'requirement engineering' (see for example [Yadav, 1983]) underlining the distinction with the process of determining and recording the information requirements of the enterprise which precedes it.

2. Conceptual design. This component leads to the construction of the conceptual schema that comprises a unique central description of the various information contents that may be in the database [ISO WG3, 1982]. The conceptual schema is a DBMS-independent information structure which is obtained through the consolidation of the user information requirements specifications. Both the static (or passive) and the dynamic (or active) characteristics of the data are taken into consideration.

3. Implementation design. The conceptual model is mapped into the structure of the selected database management system (e.g. relational, network or hierarchical data model) and the possible transactions are analyzed. The file design starts together with the access paths study.

4. Physical design. The storage schema is constructed. The physical space for records and indexes and the physical media that are to be utilized are defined. Many interactions exist between this component and management design.

2.7. Present Situation

No general agreement has been reached on an outline of the data system and the database design process. No cooperative project is being undertaken in the database area which can be compared with the CRIS exercise and which could lead to a generally agreed reference model.

New methodologies for database design are still being proposed; an example of a complete new methodology is the DATAID-1 integrated methodology, which has been developed within a general computing science project financed by the Italian Research Council [Ceri, 1983]. But a tendency is emerging in the area, as it is in the general area of development techniques for computer-based systems [Couger et al, 1982], towards the integration and improvement of existing design tools with a reduction of the total number of techniques which are adopted. At present experience has been accumulated in the adoption of tools for specific tasks, but not enough for the cooperation of the tools to be used in the complete design process.

Consequently this discussion aims to gain a much deeper knowledge of the design process. Immediate utilizations results of this work could be:

(i) support the automation of the design process;
(ii) use of these results as a starting point for the construction of a reference model to be used in the choice of the most suitable methodology, from among the available ones, for developing a specific application.

No generally applicable methodology exists which can be considered appropriate for all kinds of applications. The utility of such a database design process reference model in supporting the designer in the evaluation of alternative methodologies is shown in Figure 3.

Development of reference models has been considered a first priority for standardization in computing science by different international bodies. ISO/TC97/SC5/WG5 is concentrating its efforts in the development of a reference model for database management systems [ISO/WG5, 1983]; the Open Systems Interconnection (OSI) is an example of a reference model that has been recently adopted [OSI, 1982].

A multi-phase manner could be chosen as a way of proceeding in the development of such a reference model:

(i) development of a survey of the available methodologies, possibly through a questionnaire, as it has been done by the United States Department of Defense for Ada methodologies, [Freeman et al, 1983] and [Porcella et al, 1983];
(ii) analysis of the results;
(iii) proposal of the reference model.

A prerequisite of such an exercise is the definition of the set of requirements which every design methodology needs to possess in order to be identified as a methodology. This set of requirements is proposed in [Agosti, 1983].

3. UTILITY OF A REFERENCE MODEL FOR THE DATA SYSTEM DESIGN

Several data system design methodologies and techniques have been developed and proposed in literature. These methodologies and techniques are not directly comparable for many reasons, some of them cover different subsets of the data system design process, many of them have been established with a qualitative approach and different terms are used to identify objects of the discussion. The data system design area is one in which more skill than professionalism is still applied. The importance of the difference between skill, or craft, and professionalism in computing has been discussed by Hoare in [Hoare, 1981].

One important consequence of these facts is that the choice of one methodology for developing an application is very difficult, because it always depends on the personal expertise and knowledge of the person or group of people who have to develop the design. Furthermore, it does not obey any objective procedure. To make progress in the construction of an objective procedure it is necessary to have a systematic technique for comparing the different existing design methodologies.

If two different methodologies, say A and B, of data system design are not directly comparable, then it is necessary to have an intermediate "tool," say T, to compare with each methodology.

The result of the comparison between T and A has to be the understanding of the main features of the methodology A, like the main characteristics on which it is based, phases of the data system design process covered and possible types of utilization of A. These main features have to be expressed in a standardized and quantified way producing a final statement A′ of the methodology A. The comparison has to be reproducible between T and B, or any other methodology.

Then the procedure is repeated for B and the statement B′ will be produced.

The final statements A′ and B′ are homogeneously represented, they can now be compared and the most suitable for the specific application can be chosen.

What are the characteristics T has to have? It has to be divided in as many levels as the possible phases of the data system design process. These levels should not be redundant. The interactions between two adjacent levels have to be minimized and as much as possible have to be quantified. The terminology used has to be as general as possible and it has to be related to the relevant work in standardization, for example [ISO/WG3, 1982]; different terms used in literature have to be grouped according to similarities in their usage. T then can be called a reference model because of these characteristics.

One of the aims of the definition of T has to be its use as a tool for identifying and expressing all the differences in the database design process between the centralized and distributed context.

4. CONCLUSIONS

A general framework for data system and database design has been presented in this paper.

It is being used as a basic block for a general set of requirements for a data system design methodology [Agosti, 1983]. It can be used also as a reference to the different areas which are connected to the design of a database.

References

[Agosti, 1983] M. Agosti. *Strategy for a Data System Design Methodology*. Rep. No.TP-CS-MA-8302, School of Mathematics Statistics and Computing, Thames Polytechnic, London, England, Dec 83, pp.43.

[Ceri, 1983] S. Ceri (Ed). *Methodology and Tools for Data Base Design*. North-Holland, Amsterdam, The Netherlands, 1983, pp.x+255.

[Codd, 1970] E.F. Codd. A Relational Model of Data for Large Shared Data Banks. *Comm. of the ACM*, Vol.13, No.6, Jun 70, pp.377-387.

[Codd, 1981] E.F. Codd. Data models in database management. *SIGMOD RECORD*, Vol.11, No.2, Feb 81, pp.112-114.

[Couger, 1982] J.D. Couger. Evolution of System Development Techniques. In: [Couger et al, 1982], pp.6-13.

[Couger et al, 1982] J.D. Couger, M.A. Colter, R.W. Knapp (Eds). *Advanced System Development/Feasibility Techniques*. Wiley, New York, NY, 1982, pp.xiii+506.

[Freeman et al, 1983] P. Freeman, A.I. Wasserman. Ada Methodologies: Concepts and Requirements. *SOFTWARE ENGIN. NOTES*, Vol.8, No.1, Jan 83, pp.33-50.

[GAO, 1979] United States General Accounting Office. *Data Base Management Systems — Without Careful Planning There Can Be Problems*. Report To The Congress of the United States by the Comptroller General. Jan 79, pp.ix+48.

[Ghezzi et al, 1982] C. Ghezzi, M. Jazayeri. *Programming Language Concepts*. Wiley, New York, NY, 1982, pp.xvi+327.

[Hoare, 1981] C.A.R. Hoare. Professionalism. *Computer Bulletin*, II/29, Sept 81, pp.2-4 & 27.

[ISO/WG3, 1982] ISO/TC97/SC5/WG3. *Concepts and Terminology for the Conceptual Schema and the Information Base*. J.J. van Griethuysen (Ed). Mar 82.

[ISO/WG5, 1983] ISO/TC97/SC5/WG5. *An introduction to a data base management system reference model for standardization*. Rep. No.112, Sept 83.

[Jenkins, 1981] G.M. Jenkins. The systems approach. In: [Open Systems Group, 1981], pp.142-168.

[Kast et al, 1981] F.E. Kast, J.E. Rosenzweig. The modern view: a systems approach. In: [Open Systems Group, 1981], pp.44-58.

[Lum et al, 1979] V. Lum, S. Ghosh, M. Schkolnick, D. Jefferson, S. Su, J. Fry, T. Teorey, B. Yao. *1978 New Orleans Data Base Design Workshop Report*. IBM Res. Rep. RJ2554, IBM Res. Lab., San Jose, CA, Jul 79, pp.117.

[McCracken et al, 1982] D.D. McCracken, M.A. Jackson. Life Cycle Concept Considered Harmful. *SOFTWARE ENGIN. NOTES*, Vol.7, No.2, Apr 82, pp.29-32.

[Maggiolini, 1981] P. Maggiolini. *Costi e benefici di un sistema informativo*. Etas Libri, Sonzogno (MI), Italy, 1981, pp.252.

[Olle, 1982] T.W. Olle. Comparative Review of Information Systems Design Methodologies — Stage 1: Taking Stock. In: [Olle et al, 1982], pp.1-14.

[Olle et al, 1982] T.W. Olle, H.G. Sol, A.A. Verrijn-Stuart (Eds). *Information Systems Design Methodologies: A Comparative Review*. North-Holland, Amsterdam, The Netherlands, 1982, pp.x+648.

[Olle et al 1983] T.W. Olle, H.G. Sol, C.J. Tully (Eds). *Information Systems Design Methodologies*. Participants Edition. North-Holland, Amsterdam, The Netherlands, 1983, pp.ix+226.

[Open Systems Group, 1981] Open Systems Group. *Systems Behaviour*. 3rd Ed. Harper and Row, London, 1981, pp.332.

[OSI, 1982] ISO/TC97. *Information processing systems —Open systems interconnection — Basic reference model*. Draft Int. Standard ISO/DIS 7498, Int. Organization for Standardization, Apr 82, pp.78.

[Parent, 1981] C. Parent. *Project SCOOP — Etude de models semantiques de donnees*. Rapport RT.SCOOP 81.5, Systemes Informatiques, Institut de Programmation, Universite P. et M. Curie, Paris, France, May 81, pp.31.

[Porcella et al, 1983] M. Porcella, P. Freeman, A.I. Wasserman. Ada Methodologies Questionnaire Summary. *SOFTWARE ENGIN. NOTES*, Vol.8, No.1, Jan 83, pp.51-98.

[Smith et al, 1982] J.M. Smith, D.C.P. Smith. Principles of Database Conceptual Design. In: [Yao et al, 1982a], pp.114-146.

[Teorey et al, 1982] T.J. Teorey, J.P. Fry. *Design of Database Structures*. Prentice-Hall, Englewood Cliffs, NJ, 1982, pp.xv+492.

[Tsichritzis et al, 1982] D.C. Tsichritzis, F.H. Lochovsky. *Data Models*. Prentice-Hall, Englewood Cliffs, NJ, 1982, pp.xiv+381.

[Ullman, 1980] J.D. Ullman. *Principles of Database Systems*. Pitman, London, England, 1981, pp.379.

[von Bertalanffy, 1981] L. von Bertalanffy. General system theory — critical review. In: [Open Systems Group, 1981], pp.59-79.

[Yadav, 1983] S.B. Yadav. Determining an organization's Information Requirements: A State of the Art Survey. *DATA BASE*, Vol.14, No.3, Spring 83, pp.3-20.

[Yao et al, 1982b] S.B. Yao, S.B. Navathe, J.L. Weldon, An Integrated Approach to Database design. In: [Yao et al, 1982a], pp.1-30.