December 1993

Criteria for Evaluating Data Modeling Concepts of Data Analysis

Winnie Cheung  
*IBM China / Hong Kong Cooperation*

Masatoshi Nishio  
*Nagoya University of Commerce and Business Administration*

Follow this and additional works at: [http://aisel.aisnet.org/pacis1993](http://aisel.aisnet.org/pacis1993)

**Recommended Citation**  
[http://aisel.aisnet.org/pacis1993/55](http://aisel.aisnet.org/pacis1993/55)

This material is brought to you by the Pacific Asia Conference on Information Systems (PACIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in PACIS 1993 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.
CRITERIA FOR EVALUATING DATA MODELING CONCEPTS
OF DATA ANALYSIS

WINNIE YUK-WAN CHUING, IBM CHINA/HONG KONG CORPORATION

MASATOMI NISHIO, NAGoya UNIVERSITY OF COMMERCE AND BUSINESS
Administration, MieSHin-cho, Aichi, 470-01 JAPAN

ABSTRACT

To account for the effectiveness and efficiency of different data models, this paper proposes criteria for evaluating modeling concepts of data analysis. It is believed that a logical data analysis should support: structural validity, functionality, maintainability, comparability, and simplicity. Understandability, also, is a prerequisite for effective modeling methodologies and effective implementation.

To illustrate the application of these criteria, a data model is used as the tool for logical data analysis. A plausible approach is based on a simple, natural, and valid interface between an end-user and data. Also, the standardized graphics and symbols of the data model are presented.

In conclusion, this article has contributed towards the logical data structure of an enterprise effectively and efficiently, and in an understandable and communicable manner. These are believed to be the essential requirements of the logical data analysis.

INTRODUCTION

As an increasing number of organizations implement systems employing database technology, the focus of information systems methodologies tend to be oriented towards data. Another change in the information environment is the emergence of information resource management (IRM) concepts. With the IRM concepts, attention is now to be conceptually separated from the computer or other physical system that contains the data, and the necessity of managing information using the same management techniques that are used for other resources becomes apparent.

As a tool for data analysis, the concept of data model has been more important. A considerable number of data models have been proposed for the logical design of databases. Nevertheless, there is still a lack of evaluation criteria as well as a question of what should be modeled, and what is a good data analysis modeling concept.

A list of criteria is proposed in this paper for evaluating the modeling concepts of data analysis. A data model, a popular data modeling concept in Japan, is used as an example for illustration of an application of these criteria. It is hoped that the proposed criteria could provide an effective basis for evaluating different modeling concepts of data analysis, and guidance for designing new data analysis modeling concepts.

DATABASE APPROACH, DATA ANALYSIS AND DATA MODELS

Database Approach

When more and more organizations implement systems employing database technology, a new attitude has been developing towards data processing. This new attitude has been termed the database approach. This approach emphasizes a data-oriented view, where data is shared between data areas. It is constructed with the concept of information management, in order to create a total information system (Sapirch, 1987).

The database approach is a database model, which is a conceptual model that is used in the design of databases that support many different applications and users. The database approach emphasizes a separation of physical and logical design, and the database model provides a systematic method of better understanding of the data structures of the enterprise and the data resources in communication between people and procedures (Pangalos, 1987).

Data Analysis

A fundamental part of the database approach is data analysis, which is concerned with identifying the data resources of an organization and providing a sound basis for conceptual database design.

Traditional systems analysis (figure 1) is regarded as consisting of two projects: data analysis and functional analysis.

With the database approach and the advent of database technology, the focus of information systems methodologies tends to be centered on data-oriented perspective. Data-oriented perspective places emphasis on a complete and thorough analysis of data and its relationships. As a result, a new positioning and approach to data analysis systems development is prepared. It is shown in figure 3.

Data Models

The fundamental purpose of data analysis is to identify the data resources of an organization. However, data resources do not exist in isolation but are associated with one another. Maps need to be drawn in which data resources are associated with each other, and what types of associations these are. Such maps give an overall representation of the data resources that is identified and needed to run an organization. They are referred to as data models.

The original model of data was the architecture proposed by John Von Neumann (figure 3). Beyond Von Neumann's model, the data models developed have mainly taken the form of ad hoc solutions to the problem of how to represent and manipulate data in a machine based on the Von Neumann architecture. These models (notably the hierarchical and network models) do not have many useful formal characteristics and are only of interest when considering the question of how a database might be implemented, as opposed to the far more important question of how it might be logically represented or described.

A considerable number of data models have been designed on an end user's formal basis, for example, relational data model (figure 4), entity-relationship model (figure 5), and the logical data model. These data models have made important contributions to the ways that we can think about data and data analysis.

Relational model (Chen 1976) and entity-relationship model (Chen 1976) have been widely used as the basis of a number of data analysis projects (Prous 1979) (Davenport 1980). However, as a tool for logical data analysis, they are not without shortcomings. For example, the absence of a diagrammatic technique for representing the logical data structure of an entity-relationship model is a picture format presents an important handicap of the relational model (Tsukahara 1989). At the same time, the entity-relationship model is a conceptual description of the data that does not determine which relations are necessary and which
relations can or should be omitted.

To reduce the inherent complexity of data analysis and to increase precision, a new generation of data models, called semantic data models (SDMs), was introduced. SDM concepts (Hewer and McLeod 1989) are intended to be more appropriate for data analysis and for specifying real or imagined properties because of their abilities to represent classes of real-world semantics. A number of semantic data models have been proposed. One such model is the TD data model discussed by Tsuchik.

**TD DATA MODEL**

The TD data model (Tsuchik 1979) is a type of semantic data model used to represent real-world data independent of physical representation. The title TD is composed of the first initials of the family names of the inventors: Dr. Masahiko Tsuchik and Dr. Miyabe Motaka.

The TD data model is built around the ideas of entities and relationships (Appendix 1). It has no description as known in the conceptual database diagram. According to the layout rules (Appendix 1), entities and relationships between entities can get almost the same product for the same analysis of an application.

The data analysis approach to TD data model breaks the problem of adding meaning to a database into three stages. First, identify a set of semantic concepts that describe relevant information and meaning. Second, represent the semantic concepts identified in stage 1 in terms of a set of corresponding symbolic objects. Third, devise a set of integrity rules that govern the description of the semantic concepts in terms of the symbolic objects.

For the sake of illustration of an application of logical data analysis with the TD data model, a hypothetical situation concerning a ball-bearings manufacturing company, the TD, is created. The size of data to be considered is that of a typical order processing environment (Chen 1976). Other examples of applications can be found in (Chen and Mishra 1982), (Agrawal and Naik 1992), (Hwang and Naik 1993) and (Yu 1993).

**TD: A Politepical Case Example**

TD is situated in Tokyo, Japan. There are only two units of business in TD, namely: manufacturing of ball-bearings and sales of ball-bearings. Structurally, it has a headquarters office and four branches. Inside the headquarters, there are some supporting departments: sales, manufacturing, personal and accounting. But the description of this paper will mainly centre on the order processing system.

Ball-bearings of TD are sold by TD such as shown in Figure 4. The order processing system is represented in Figure 9. Data analysis of order processing system by TD data model approach starts with the selection of top-output documents for analysis. The documents are selected after interviewing the related users for their requirements. For this case, six documents are chosen and it is believed that the items they hold constitute the order processing system of TD (Figure 9). To carry out data analysis by TD data model, facility, firstly, it is important for us to number the data items in each selected form (Figure 4). It shows on an example the purpose of numbering the data items is to identify the major entities and then to allocate those redundant items. After numbering the data items, we have to define the meaning of each data (see example is shown in Figure 10). To have a complete understanding of data meaning, the data inventory is significant in this stage. At this stage, the major resource entities and the type of relationships between these are defined. They are then represented in the partial conceptual database diagram (figure 11 shows an example). Here, doubtful points or problems during the analysis are also reviewed for later discussion or negotiation. The partial conceptual database diagrams are then merged together to produce an integrated conceptual database diagram of order processing system of TD as shown in Figures 12 and 13. These integrated diagrams (result of logical data analysis with TD data model) enable us to understand the order processing system of TD in its full complexity. At the same time, substantiation of the entities and connections of data resources lead to a creation of a sound, stable and flexibles logical database structure.

**Evaluation of the TD Data Model As a Tool for Logical Data Analysis**

**Criteria for a Model-Tool of Logical Data Analysis**

To create an enterprise-wide integrated database environment, one should not only understand the business environment in which the data model is to be used. It is not uncommon to have companies with tens of thousands of data elements, mostly scattered and poorly defined. The sheer volume of non-descriptive descriptions to be handled, not to mention issues related to the meaning of data and their relationships, makes virtually impossible to approach the overall problem at once, in a top-down fashion. Evidence of this can be very clearly seen from the failure of a number of long term expensive projects that performed data modeling in a case-by-case and piecemeal way.

On the other hand, the follow-up approach, which has been used for many years, and defines the data uniquely for each application, usually leads to a non-integrated environment such as described previously.

The discipline of data analysis is primarily concerned with the knowledge and technologies required to analyze, define, group and relate data in a way that will promote sharing and reusability as much as possible.

Data analysis with the purpose of integration is a highly skilled task that requires a profound knowledge not only of a set of technologies, but also of the underlying data, as it relates to the operations of the enterprise. Equally important is the way in which this knowledge is acquired, stored and communicated to people with backgrounds as varied as applications programmers and top level managers.

In leading an enterprise into an integrated database environment, data management must create data abstractions from the requirements level to the lowest and most basic form. Such abstractions must model data in a way that reflects the fundamental nature of the enterprise. They must be easily understandable throughout the enterprise and, in addition, lead to data structures that will minimize redundancy, optimize performance and facilitate evolution.

To accomplish all this, what is needed is an ontological data model that encompasses the data knowledge needs of users, systems developers, database designers and performance specialists, and is closely integrated with a system design methodology. It must also lead itself to evolutionary development while maintaining a clear overall view. More than enterprises already in existence, new ones will be created with data model specifications that must be preserved; this data model must be able to cope with existing data architectures and still integrate them with new developments.

Although a considerable number of data models have been proposed, there are still no widely accepted criteria for such an optimal data modeling tool of logical data analysis. This paper proposes such criteria. The criteria provide a comparative basis for evaluating modeling concepts of data analysis. The criteria are not subjective because they are summarized and reevaluated.
A data analysis modeling tool is used during the conceptual design to capture the meaning, or semantics, of the application. To be successful, it 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.

Small number of simple concepts aids understanding, design, and analysis even by untrained people (e.g., by users or by non-systems professionals). Simplicity is supported by direct and unique modeling concepts. Significant properties of the application should be representable directly, that is, using a small number of modeling concepts. There should also be relatively few (e.g., one or two ways of representing each property) because modeling concepts should not overlap in their ability to represent similar properties.

Modeling concepts should also be common to the problem domain at hand and familiar to users and designers. In other words, they should be close to the users’ vocabulary and view of application.

Diagram and drawings are the most efficient and practical tools for communicating and understanding global data structure. A good data model should lend itself to graphic representation for documentation. This helps to verify a specification with respect to a corresponding implementation and validate the database with the schema; and to derive all implicit constraints.

Finally, an optimal data model should also support effective modeling methodology and efficient implementation. * methodological denotes a project’s sequence of execution or network. It is the “makeup” for a project (Boyle and Boyle 1988). As a result, data model should support effective modeling methodology such that organization and direction for the data analysis project can be provided and data analysis can be implemented efficiently.

Data analysis is used for identifying the data resources of an organization, so it is important that the data modeling tool being employed can actually reflect and represent the fundamentals of an organization. The criteria proposed here can be used as a basis for evaluating the different modeling tool of logical data analysis. It is believed that an effective and efficient data modeling tool can satisfy all or most of these criteria.

In this paper, TH data model is used as the modeling technique for data analysis. To account for the effectiveness and efficiency, modeling concepts of TH data model are evaluated against the criteria discussed above.

Evaluating TH Data Model with the Criteria

As implied from the case example, TH data model is a valuable tool for logical data analysis and it satisfies the criteria given in the above section.

Structural validity

TH data model uses current input-output documents as materials for data analysis (form approach), so it is able to grasp and represent detailed user information requirements and business rules in the conceptual domain without personal bias.
Figure 12: Integrated conceptual database diagram of order processing system of TBK (Resources)
Figure 13: Integrated conceptual database diagram of order processing system of TBK (Events)
The classification of entity types into the three basic classes of resources, events, and summaries provides a means with which to access static and dynamic properties. TH data models also pay attention to the representation of subtypes and substructures. Subtypes are now recognized as an important construct in a data model. However, the same may not be true for substructures. If an attribute is defined as belonging to only one entity type in the TH data model, an attribute of the subtype group or substructure may be defined only when the EA decides to assign a specific attribute for managers or female employees. The data model has not been discussed very much. As accepted commonly the same entity has the same key values. But slight differences may occur, depending on their data occurrence times or their data percepting aspects sometimes have the same key values. For example, the same key values may be given to an order event and its shipping event. The same entity number is often used to identify both an employee and his/hers.

Substructures are shown in conceptual database diagrams as boxes created with dotted lines in order to show more precisely the data relationships with other boxes.

Extensibility

New entity types can be defined and entity membership rules can be altered. Properties of entity, attribute, and association are distinct, hence, they may be considered independently. Due to concept independence, constraints can be added or deleted.

Integrity

The TH data model gives an identification criteria to each repository component as entity types, attributes, domains, files, and databases. In other words, it provides a standard formal basis for specifying properties independently, explicitly, and implicitly, which can lead to an integrated environment.

Sharability

The results of data analysis with TH data modeling methodology is translatable, useful, and scalable. A data model design should be sharable, that is, it can be accessible by multiple applications and users to meet varying access requirements.

Nonredundancy

Rules of normalization are applied to the key and attributes of each entity type in order to detect any hidden relationships, and to ensure that the model represents the most fundamental understanding of the environment. This association between an entity and a key or attributes values permits details about members to be ignored when considering an entity set. As a result, the principle of some facts in one place be implemented.

Semantic Relativity

Dependent on the context, an object can be viewed as an entity, as an attribute, and also as a key value.

Simplicity

TH data model is constructed with a minimum set of primitive concepts and objects. The entity, attribute, and association of the modeling concepts are simple, natural, and direct; there are few ways of describing in a more abstract way. It can be conducted by any person because data analysis with the TH data model methodology does not require any programming or DBMS knowledge.

Understandability

The classification of entity types into the three classes of resources, events, and summaries is very effective for understanding data semantics. Through this practice, we get to know the common characteristics of the life insurance business and the construction business from the standpoint of data structure. We also view this classification as a criterion in locating entity types in conceptual database diagrams. In other words, this gives an effective foundation for layout standardization.

Picturability

The business life cycle of a sales system for mass production is also depicted as a process. This model may be composed of events such as purchase order, warehouse, order taking, shipping, and shipping payment. By arranging these event entity types from left to right in the above order, we can represent the business life cycle naturally on a data modeling diagram. We can also understand clearly the relation between the events and data by means of the diagram. The quality of a diagram is effectively improved by understanding and enhancing the diagramming standards called the Conceptual Database (CD) Diagram or TH Diagram. But it could better be shown in the business process, according to its layout rules for entity types or boxes, anyone will get almost the same analysis of an application. We usually evaluate the above diagram in terms of the above methods and the fast, he does on the location of boxes for entity types in an application program.

Consistency, verification and validation can also be based formally on the conceptual database diagrams.
Appendix D: Basic Concepts and Modeling Structure of TM data model

Basic Concepts

The basic concepts of the TM data model are simple and they are defined in ways so as to achieve the purpose of information resource management.

Entity e(i)

An entity e(i) is something that we perceive and want to manage. It may be tangible or intangible. Each entity should have a name. Examples of entities include employees, products, payments, and so on.

Attribute x(j)

An entity is assumed to have at least one attribute value x(j) for a concrete entity e(i). However, we would not have perceived it for management. The number of attribute values that an entity has depends on the management objective rather than its nature. Different users may want different numbers of attribute values of an entity. In this case, the data administrator (DA) must define all attributes that the end-users want.

Key value skl(i)

At least one key value skl(i) has to be assigned to an entity as a unique identifier. This is regarded as a special attribute value.

Entity record

An association between an entity and its list of key or attribute values. An entity record may be written as:

\[ (e(3), \{x(11), x(12), x(13), \ldots, x(n)\}) \]

where \(e(3)\) is used to denote the identifier.

Entity type En

We may have similar entities that can be managed in a similar manner. They should have the same set of attributes. This similar perception will come from abstract operations by human beings. Entities (entity occurrence) are mapped to an entity type En at the same time. This mapping or entity-typing is written as:

\[ e(1) \leftarrow En \]

Attribute Ann in an entity record type

If we recognize that two entities e(2) and e(4) are in the same entity type En, we should already have recognized that the two attribute values of e(2) and e(4) are for the same attribute Ann. Thus, association between an entity type and key or attribute value of an entity record type may be written as:

\[ En: \{Ann\} \times \{Ann, \ldots, Ann, \ldots\} \]

This may correspond to the third normal form of the relational model.

The mitters that DA should manage as information resources are those entity type based attributes such as Ann as opposed to crude attributes of e.

Many DBs have been used to store crude attributes.

REX

Suppose we have the entity types:

\[ En: \{Ann\} \times \{Ann\} \times \{Ann\} \times \{Ann\} \times \{Ann\} \times \{Ann\} \times \{Ann\} \]

and the key values of En or Ann is used for values of Ann. Then Ann is called REX or referencing key which may correspond to a foreign key in the relational model.

Since the same Ann value can occur for Ann many times, one-to-many relationships between Ann and Ann are represented by Ann and Ann. This is called a KEY-KEY association. Join operations are defined as being performed only via KEY-KEY associations, although these are permitted via REA-KEY associations if they have the same domain in some REA sets.

In which temporary or redundant attributes of Ann and Ann are included between internal parentheses as shown above. This type of record form, known as first normal form, can be observed in output and physical files.

Composite attribute

An attribute including a key value is called a composite attribute. When a key is a composite, the related REXs should also be composites.

Domain

Values for an attribute other than REXs may be integers, real numbers, or text describing such aspects as length, weight, and address. The form they take is the same for an attribute. Thus we may consider that they come from a value set of the same form, that is domain. Values of an attribute should be checked upon entry to determine whether they satisfy the domain specifications.

There seems to be no definite criteria for determining the size of a domain. It should be determined by the DA from the standpoint of data administration.

Subtype Em, En

Suppose we have two entities e(1), e(2) as shown below:

\[ e(1): \{x(1), x(2), x(3), x(4), x(5), x(6), x(7), x(8), x(9)\} \]

End-user (EU) may recognize that they are in the same entity.
1993 Pan Pacific Conference on Information Systems

Figure A.1: Conceptual files, concept record and data items

Types of conceptual files
Representative types of conceptual files are as follows:
1) Resource files (such file). For example, parts file (fl) etc. Generally, on real-time removal is required. So it is a rather stable type of conceptual files.
Standard data, standard value etc., are recorded. They are usually of very long record longevity and with high common use property.
2) Inventory type file (mark B). Similar to resource files, files that store inventory data such as inventory files is very important concept. When the item is used, record the amount of use. Real time removal is frequent so data value is quite unstable. The record itself is kept for long duration. It is more peculiar to application program.
3) Event file (mark C). For example the order file (c). Real time removal (record formation) is frequent with unstable data value. Record is generally kept for a shorter period. They are also called the transaction file and are usually peculiar to application.
4) Summary file (mark D). For example the actual order file, year, year-month, year-week, etc., are used as an element of ASET. Generally, summary files are produced by analysis of batch from event files. Contents of summary file is processed data (in kind of redundant data in a year). Compare with event file, it is kept for a long period, and more peculiar to application.

In many situations, when, where, who, when are maintained in summary file.
5) Cross-sectional file (mark E). It is to record the state of some particular during time (snap shot) in resource or inventory files. The format of key is similar to that of summary file. It is peculiar to application.

Data analysis of conceptual database design is the process which we transform our understandings to record form. The classification of conceptual file is used to something this transformation process.

How to make the key A key is the data item(s) which distinguish a recognized entity from a conceptual file. It must have an unique name (code, number and etc.) Examples of key are part s, context etc. For resource type file, the key is usually obtained from the generated code of that resource type, for example, product number of product type, department code of department.
For inventory type file, we usually use the resource number, or the group of resource numbers as the key type (accounting, parts, inventory).

For event type file, event codes are generated one by one following the sequence of happenings and they are generally automatically assigned by computer. Example [order, order detail line].

Keys of summary file and cross-sectional file are of the same structure and they are usually made up of reference keys and dates.

From external records (RMS) to conceptual records (CRS)
External records are records with redundant data. They are the same as input and output of actual record.

Example of external records, "order", is shown in figure A.2.

In the past, computer files were stored (totally unchanged) as the external records were seen by users. As a result, redundant data were always included.
With the progress of data analysis, redundant data are eliminated and data can be stored in similar forms but of
1993 Pan Pacific Conference on Information Systems

Figure A.3: Order file with redundant data

Conceptual nature.

Normalization

As shown from the above figure A.3, the products' names and their unit prices might be redundant if the product's number is given. So the figure is made of external records which contain redundant data. It cannot form for conceptual file because it breaks the rule of "one fact to one place".

To overcome such redundancy problem, TH data model records the products' names and their unit prices in another conceptual file. This method is generally known as the "normalization" process.

In TH data modeling methodology, a file can be normalized in the following way:

<table>
<thead>
<tr>
<th>Conceptual</th>
<th>detailed</th>
<th>order, product code, order, date of delivery, order amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normalized</td>
<td></td>
<td>product name, order, product code, date of delivery, order amount</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Normalization requires three actions to be performed on the attributes of an entity type. First, repeating groups are removed. Next, attributes are removed which are not directly dependent on the identifying attributes. This results in attributes being in third normal form (INF). If "name" had been identified as one of the entity types, the normalization process may show (depending upon the particular environment) that there are relationships "hidden" inside the entity type, i.e. several men may reside at the same address or a man may have several addresses. Normalization would then produce a new entity type "property". Normalization ensures that each attribute type has only a single value for each occurrence of the entity type concerned, i.e., the repeating groups of attribute types are shown as entity types in their own right. Also, it ensures that the value of each attribute is really independent. This means that the value is not affected by any change in relationship between the entity and any other entity.

Appendix A 21: Rules of Conceptual Database Diagrams

Diagrams or drawings are the best method for human beings to understand complex entity relationships. In drawings, we can easily find out what is included or constrained. At the same time, it is more efficient for users and database designers to communicate through diagrams and drawings.

The TH data model uses the drawing standards known as Conceptual Data Base (CDB) Diagram or TH Diagram (Yoshikawa 1991). According to its layout rules for entity types or boxes, anyone can get almost the same product for the same analysis of an application.

A box is used to describe an entity type as shown in the example (Figure A.4) below:

Figure A.4 As an entity type

[Diagram and textual explanation of entity types and relationships]