

Human Memory Structures for Data Dictionaries
Found in Database Designers/Analysts

by

John Smelcer

and

Marilyn Mantei

May 1984

Graduate School of Business Administration
The University of Michigan
Ann Arbor, Michigan
(313)763-5936

This paper is not for quotation, reference, or duplication. It has been submitted to the Fifth Annual Conference on Information Systems. Comments are welcome.

Abstract

The paper presents research which investigates the cognitive skills that go into making an expert database designer. We are focusing on the task of conceptual database design, that is, the task of taking a description of database items and creating a graphical diagram of how these items are to be related in the database; for example, an entity-relationship diagram.

To study the cognitive skills that experts possess, we captured order-of-recall data for database designers and non-designers from a memorization/recall task which used a data dictionary. We used this data to derive the human memory structures formed for the data dictionary by database designers and non-designers. Across designers memory structures were similar and similar to their corresponding database designs. Across non-designers memory structures were similar, but dissimilar from the memory structures of designers. This structure of memory in database designers reflects an important skill - the ability to recognize and organize familiar components to a conceptual database design. A mental model of the conceptual database design task is proposed which explains the similarity of memory structures and database designs.

1.0 Introduction

Database design is a complex and difficult task. Recent efforts to aid the designer have been in the form of computerized tools and workbenches (Teorey & Cobb, 1982; Teichroew & Jerkey, 1977; Database Design Inc., 1981). Unlike the computerized aids for programmers, these design aids for database designers have not gained widespread acceptance and use. An accurate model of the mental processes of database designers is needed in order to tailor design tools to aid designers in their mental task of database design.

One important mental skill of experts in all disciplines is organized human memory. We use a technique by McKeithen, Reitman, Rueter, and Hirtle (1981) to measure the memory organization of expert database designers and apply it to the memory organization formed after completion of a conceptual database design task. Using two groups of subjects, database designers and non-designers, we captured the database designs of designers and the memory organizations of both designers and non-designers. We found that the database designs of designers were nearly identical, that the structures of memory for designers were all similar, and that the designs were similar to the memory structures. For the non-designers their memory structures were similar, but distinguishable from those of the designers. Upon closer examination of the memory organizations we found that the designers used a "natural" view of the data to organize the database elements, while non-designers used a "usage" view of data.

The rest of the paper elaborates these ideas. The next section discusses what constitutes the type of memory organizations experts seem to possess, lays the groundwork for the experimental design task we gave our subjects, and explains the similarity comparisons we performed on the memory structures. A third section describes the recall experiment we ran and presents the results of this experiment. The fourth and final section discusses these results and what they imply about the task of database design.

2.0 Related Work: Memory Organization, Programming Expertise, and Conceptual Database Design

Before presenting our research, it is important to define the key concepts and lay out a theory for the cognitive processes of database design.

2.1 Definitions

The key psychological concept used in this paper is human memory organization. It refers to the mental organization of memorized information, and has two components. The first component is called the 'chunk' (Miller, 1957). A chunk refers to the grouping of items in memory into a single stored unit. For example consider the seven letters I B M R C A E T. Recall of these letters is made easier by grouping them into familiar, previously learned units, for example IBM RCA ET. Instead of seven letters to recall there are now only three "chunks."

Experts in all disciplines have built extensive chunks of information. Chess masters recognize pawn chains and castled-

kings not as individual pieces but as familiar patterns (Chase & Simon, 1973). Go masters recognize groupings of stones as attack and defend configurations, and not as individual pieces and their placement on the playing board (Reitman, 1976). Computer programmers recognize an IF-THEN-ELSE as one chunk and not as individual keywords and variables (McKeithen et al, 1981). These familiar patterns - "chunks" - are usually organized into higher-level structures.

This second component of human memory organization we will refer to as memory structure. Once memory chunks have been identified the structure within and among those chunks can also be derived. Consider the following list of words:

nails, paper clips, staples
 books, memos, magazines
 bookcase, stool, toothpick
 shirt, slacks, socks

Each group of three items might constitute a chunk once the list is memorized. Yet a higher-level memory organization might also be placed on these chunks:

Common Objects

<u>Metal</u>	<u>Paper</u>	<u>Wood</u>	<u>Cloth</u>
Nails	Books	Bookcase	Shirt
Paper Clips	Memos	Stool	Slacks
Staples	Magazines	Toothpick	Socks

This hierarchical organization of the chunks represents the memory structure of an individual for these items.

2.2 Expertise

Recent studies of expert behavior (Chase & Simon, 1973a, 1973b; Reitman, 1976; McKeithen et al, 1981) provide insights into the behavior of highly skilled professionals. These studies focused on the differences between expert and novice performance

and uncovered an essential feature of skilled behavior - organized human memory. We discuss one such study and adopt its techniques to our work on database designers.

McKeithen, Reitman, Rueter, and Hirtle (1981) found memory organizations for key programming concepts among expert programmers to be remarkably similar and distinct from the organizations of intermediate and beginning programmers. Employing the Reitman-Rueter technique (described below), they had their subjects memorize and then recall 21 ALGOL W reserved words. The revealed hierarchical memory structures were then related to skill level - expert, intermediate, and beginner - and compared for similarity.

Experts' memory organizations were based on programming knowledge, beginners' based on common-language associations, and intermediates' on a combination of programming knowledge and common-language associations. They counted the number of identical chunks shared by two memory organizations and carried out a pair-wise comparison of all memory organizations. When the members of each skill-level group were compared to other members of the same group, experts were very similar, intermediates less similar, and beginners the least similar to each other. We expect such similarity in memory structures to show up among expert database designers and that this similarity will provide cues to what the underlying structure is.

2.3 Memory Structures

The work of McKeithen et al (1981) demonstrated the power of the Reitman-Rueter technique both to identify memory chunks and

to show a hierarchical structure of those chunks from a simple task - the memorization and recall of a list of words. We now want to explain that technique and the results of its algorithm in greater detail. It produces a hierarchical organization, or tree, of items from a series of recalls. The subject first memorizes a list of items, then he or she is asked to recall those items, either in any order the subject desires, called non-cued recall, and or beginning with a particular item, called cued recall. This series of recalls is then subjected to an algorithm which looks for groups of items which are recalled together on all recalls, i.e., chunks.

"From this set of cued and noncued recall strings, the algorithm efficiently finds the set of all chunks and represents this set as an ordered tree. In particular, the algorithm recursively examines the strings 'top down' for chunks. The set of all such chunks forms a lattice which is then converted to a tree, with directionality indicated where appropriate.

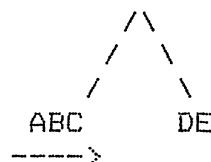
"An important detail of this technique involves appropriate analysis of the cued trials. Since the cue item may be part of a chunk whose traversal is disrupted by the cueing process (directional chunks are particularly vulnerable), only that part of a cued trial that is assumed undisrupted should be analyzed. The disrupted and undisrupted segments of recall strings are identified in an initial step of the algorithm. First, the highest-level disjoint chunks - formed by the subtrees of the root of the tree induced by all recall orders, without regard for cueing - are identified. Second, in each string the

effects of cueing are assumed to be limited to the highest-level chunk that contains the cue item. As a result, the part of each cued trial that involves traversal of the cued subtree is not used in the search for structure; only the latter parts, those involving natural traversal of the uncued subtrees, are used to build a second tree whose subtrees have the detailed structure induced from the uncued traversals. It follows that only uncued trials may be examined for the directionality of the root (Reitman and Rueter, 1980, p.561)."

By examining only the order of recall of memorized words/phrases in both non-cued and cued recalls, the algorithm in the Reitman-Rueter technique infers both the chunks and the structure of those chunks in the subject's memory.

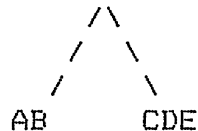
2.4 Comparing Memory Structures

Now that we have explained how memory structures are derived from recalls of memorized items, we will discuss how to compare those memory structures. McKeithen et al (1981) compared two memory structures by counting the number of common chunks they shared. In this context a chunk is defined as items that are always recalled together. So, for example, the following tree has four non-trivial chunks. (The trivial chunks are the five elementary items A,B,C,D,E and the entire list ABCDE.)



They are AB, BC, ABC, and DE. Note that directional subtrees,

e.g. ABC, are a special case and contain more chunks than non-directional subtrees. Because A, B, and C are always recalled together and always in that order, the chunks that comprise ABC are, according to the theory the possible permutations that could occur together, i.e. AB, BC, and ABC (McKeithen et al, 1981, p.312). If we wish to compare this tree to the following tree,



which has only the two non-trivial chunks AB and CDE, then the following formula defines their similarity:

$$S = \frac{\ln(\# \text{ of chunks in common} + 1)}{\ln(\text{total} \# \text{ of chunks} + 1)}$$

For this example the similarity equation above takes the value

$$S = \frac{\ln(2)}{\ln(6)} = \frac{0.693}{1.792} = 0.387 \quad (0 \leq S \leq 1)$$

This provides us with a technique for comparing two memory structures, but we still need a way to compare two database designs and a way to compare a database design to a memory structure. Unfortunately no such metrics exist. Comparing two nets, which database designs usually are, is an np-complete problem, as is the task of comparing a database design with a memory structure. Here will avoid the whole problem by relying on visual heuristics.

2.5 Conceptual Database Design

For the reader unfamiliar with database design we will now provide a brief description of the phase called conceptual database design. In the conceptual design stage the designer organizes user requirements and represents them in a graphical notation. According to Teorey and Fry(1982,p.6):

The conceptual structure, or schema, consists of basic data elements of the real world (persons, things) called ENTITIES; other data elements which describe entities, called ATTRIBUTES; and associations between occurrences of data elements, called RELATIONSHIPS.

We will refer to a conceptual structure as a conceptual database design. It results from requirements provided by the user, often extracted from interviews with users and examinations of the forms and reports in use.

From the perspective of psychology we are interested in the memory structures which are associated with the design process. Our objective in this study is to show designer - non-designer differences in memory structures. Thus, the design methodology employed by our subjects is not of great importance.

The specific conceptual design methodology used by our subjects is called bubble charting. It was developed by Martin(1976) and is a combination of Bachman data structure diagrams (Bachman, 1969) and Codd's relations (Codd, 1970). Figure 1 is a simple bubble chart of an order-inventory database design.

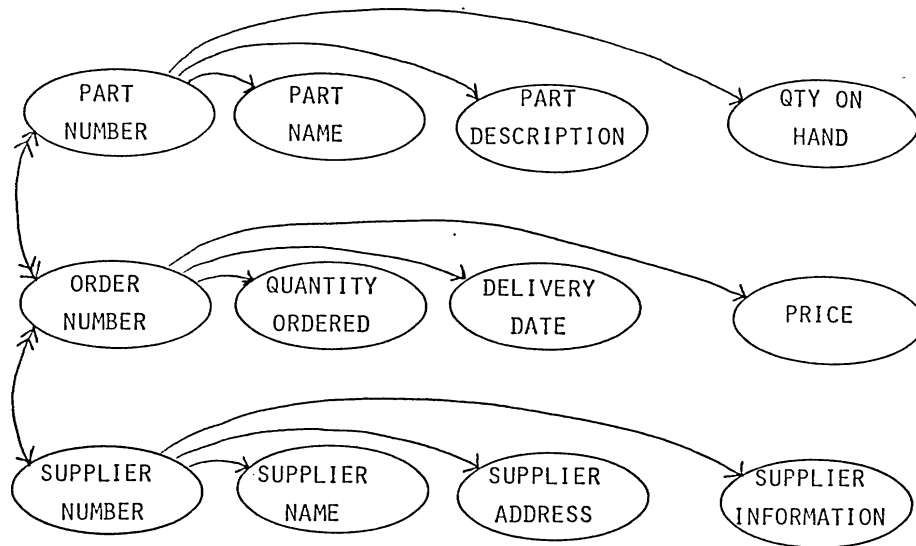


FIGURE 4
SAMPLE BUBBLE CHART

It consists of twelve data elements each contained within a 'bubble.' Those data elements are organized into three entities - PART, ORDER, and SUPPLIER. The unique identifiers for these entities are PART NUMBER, ORDER NUMBER, and SUPPLIER NUMBER, which appear as the left-most bubble in each 'row.' The remainder of the data elements which describe each entity extend to the right of the unique identifier. Relationships between data elements are represented by 'arrows,' which can have single- or double-headed arrows at one or both ends. A single-headed arrow denotes that one data element uniquely identifies another, e.g. PART NUMBER uniquely identifies PART NAME. A double-headed arrow indicates that one data element identifies many occurrences of another, e.g. each SUPPLIER NUMBER has many ORDER NUMBERS associated with it. Thus, each supplier can have many orders.

Based on this discussion of bubble charts we can see that they contain the three basic elements of conceptual schemas according to Teorey and Fry's description - ENTITIES (the groups or 'rows' of bubbles), ATTRIBUTES (the bubbles), and RELATIONSHIPS (arrows).

2.6 Theory of Memory Structures in Database Designers

In order to better understand how memory structures are related to conceptual database designs, a mental model of the design process undertaken by experts is developed next.

Like the expert programmers of McKeithen et al (1981) the database designer is highly skilled at recognizing cues and organizing information. From interviews with the user the designer structures a conceptual database design. How? This is equivalent to asking the questions, "What mental steps guide the designer in structuring a database design? What memory structure is associated with a particular database design?" The following simple model of database design will guide our answers:

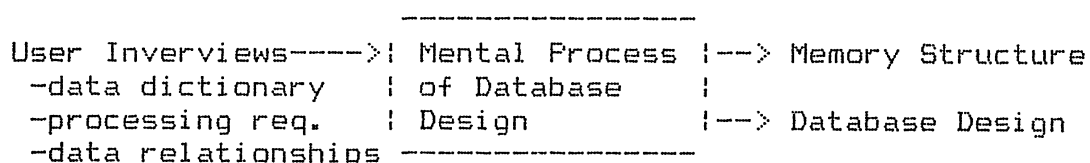


Figure 2
Mental Model of Database Design

In words, the designer processes information from the user interview and produces a database design. Simultaneously, however, some memory structure is also produced. The box in the center of Figure 2 represents the mental process of database design. Once the mental process is understood, that

understanding should then explain the two outputs - a memory structure and a database design. The mental steps probably follow the steps of the database design methodology.

The bubble charting methodology consists of the following eight steps:

- 1) select the user view
- 2) list data items - a data dictionary
- 3) identify entities in the data dictionary
- 4) select unique identifiers for the entities
- 5) draw relationships between the unique identifiers
- 6) add data attributes to the unique identifiers
- 7) add necessary unique identifiers
- 8) verify third normal form (DDI, 1984)

Steps 3 through 6 are of primary interest because, once completed, they produce an initial design. If we can understand them in detail, we can see how they contribute to the memory structure associated with a database design. Also, these were the four steps that the designers in this study performed; a data dictionary, Step 2) was supplied. For the four steps of 1)identifying entities, 2)selecting unique identifiers, 3)creating relationships, and 4)associating attributes, a model of how the designer performs these tasks will now be proposed, based protocol analyses of designers.

1. Identifying Entities: An entity can be identified from information provided by the user or from recognition of clues in the data dictionary or both.

2. Unique Identifiers: Once entities have been identified, the designer must next select unique identifiers for each one. This can often be done without assistance from the user, and the words NUMBER, IDENTIFIER, and ID play a key role. If any of these words appear with the name of an entity in the data

dictionary/list of data items, then that phrase is probably the unique identifier for the entity. For example if PART is an entity, and if PART NUMBER is a data item, then it is probably the unique id.

3. Creating Relationships: Once unique identifiers are selected the designer must rely on the user to provide sufficient information to determine the relationships among the unique identifiers. Simply put, the designer asks the user, "How are X and Y related?"

4. Associating Attributes: Finally, the remaining data items are associated with their respective unique identifiers as attributes.

We have now answered our first question: the mental steps of identifying entities, selecting unique identifiers, creating relationships, and associating attributes organize information to create a database design. Our second question, "How will these steps affect the designer's memory structure?" can now be answered. As Figure 2 (reproduced below) implies we assume that

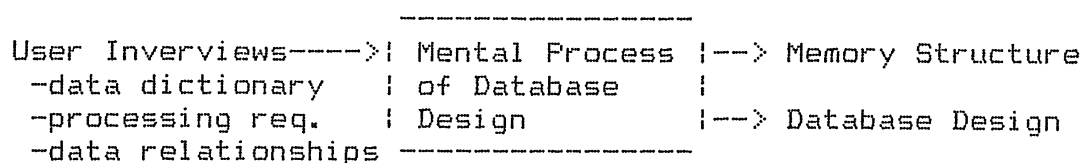


Figure 2
Mental Model of Database Design

the steps used to organize information for the database design will also organize the designer's memory. In other words these four database design steps will have four corresponding effects on the designer's memory structure:

1) entities correspond to chunks - the unique id and attributes which form each entity will also form each memory chunk;

2) unique identifiers will appear first in each chunk - the unique identifier for an entity will be the first data item in the corresponding chunk;

3) relationships among entities will be reflected as the relationships among chunks - if two entities are related then the two chunks corresponding to those two entities will be closely related in the memory organization; and

4) ordering within entities corresponds to ordering within chunks - the order of unique identifier, attribute-1, attribute-2, ... in a particular entity will be the same in the corresponding chunk.

We have proposed a mental model for four steps central to conceptual database design and have predicted that those steps will be reflected in the similarity between the memory structure and the database design of a skilled database designer.

This section on related work has introduced the key concepts of chunks and memory structures in human memory, briefly reviewed the work of McKeithen et al (1981) on memory organization in computer programmers for ALGOL W keywords, the work of Reitman and Rueter (1980) on deriving memory structures from recalls, and the theory behind comparing two memory structures for similarity. We expect to find results similar to those of McKeithen's (1981) programmers. Database designers should show similar memory structures, structures similar to their database designs; non-designers should be dissimilar in their memory structures, but

more similar to the designers than to each other.

III. Design - The Data Dictionary Recall Experiment

Now that the reader has a theoretical background in expert behavior, human memory organization, and conceptual database design we will now explain the purpose of this research. That will give the reader a framework for understanding the results and their application and relevance to the discipline of database design.

This experiment was designed to capture the semantic memory structures of database designers and non-designers. For the designers we wanted to record the database design created from the same list of phrases used to capture their memory structure. In order to isolate one of the factors affecting the design of a database, these subjects were asked to design a database only from a data dictionary without the customary user interviews or associated forms. Thus, for designers we extracted both the organization of their memory structure and the database design created from a data dictionary. For non-designers we captured only the memory structure and the subject's pre-memorization organization. This enabled us to perform three types of analyses

- 1) a comparison of memory structures within and between groups
- 2) a comparison of memory structures to database designs for the designers and
- 3) a comparison of pre-memorization organization to memory structures for the non-designers.

The above goals led us to make certain experimental design

decisions. Primarily we decided to capture first the database design for the designers and then their memory organization. This would tell us what human memory structure was cued by the task of database design. The other design decision involved the selection of items for the data dictionary. This list of items had to be both easily memorized by non-designers and designers, as well as be a valid data dictionary for creating a database design. The list was extracted from an example in the database design literature (Martin, 1977) carefully chosen to be of reasonable length for memorization.

In order to capture the memory structure resulting from the task of database design, we selected a data dictionary from a database design example in the literature and had the designers in this study create a database design from it.

IV. Methodology - Experimental Procedures

The subjects were partitioned into two groups - designers and non-designers. The designers were all experienced in bubble charting as a conceptual database design technique. The non-designers were all MBAs, currently graduate students at the University of Michigan. For the designers there were two tasks: 1) design a bubble chart from a data dictionary and 2) memorize and recall that same data dictionary. The non-designers 1) organized the data dictionary and 2) memorized and then recalled the data dictionary. The database designs, recalls, and pre-memorization organizations were all recorded.

The designers' first task was to create a database design from the data dictionary, during which time they were asked to

think aloud. The design process was recorded on video tape. Following the Reitman-Rueter technique (Reitman and Rueter, 1980), once the subject had memorized the list of words he/she underwent both cued and non-cued recalls. A cued recall was elicited by the following instruction, "Now recall all the words in any order you wish but begin with the item PART NUMBER, and those that go with it," while a non-cued recall was elicited by the instruction, "Now recall all the words in any order you wish." Each subject performed approximately ten recalls for a data dictionary containing fifteen items.

The non-designers were given a list of words, actually a data dictionary for a database design on 3X5 cards in random order. Each non-designer was asked to first order the cards to facilitate memorization, sort of a novice approach to database design. Their second task - memorization and recall, was identical to that of the designer.

V. Analytic Techniques and Results

The analytic techniques used here were all focused on the verbal recalls of the memorized items. First, the cued and non-cued recalls were subjected to an algorithm for building a hierarchical memory structure or tree. Second, each tree was analyzed for degree of structure. Finally, all possible pairs of trees were compared for similarity. The details of these three techniques are discussed below. However, we also used some visual heuristics in order to perform additional analysis of the database designs and their similarities to each other and to the memory trees.

The results and analysis of this experiment can be summarized by the following figure:

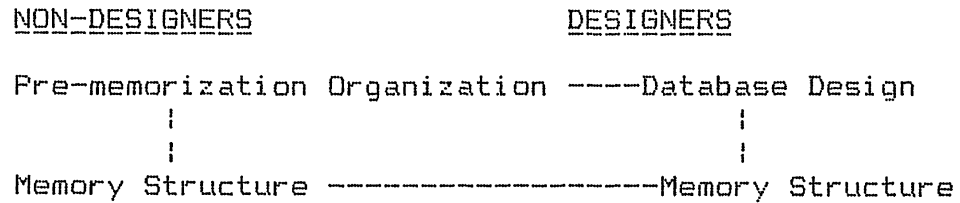


Figure 2 - Comparisons Within and Among Groups of Results

Pre-Memorization Organization is the organization that the non-designers said that they used for the data dictionary prior to memorization. The Database Design is the database design created by the designers, and the Memory Structure is the semantic memory structure derived from the recalls by the Reitman-Rueter technique.

The data collected by this study included, for all subjects, memory structures, derived from recalls of the memorized data dictionary, for the designers, database designs created from the data dictionary, and for the non-designers, pre-memorization organizations of the data dictionary. These three sets of data are compared within and among each other in the following sections.

5.1 Within and Between Memory Structures: Designers and Non-Designers

When memory structures are compared three pairings can be made: 1) among designers memory structures are very similar (.74) 2) among non-designers memory structures are similar but less so than for designers (.66) and 3) designers and non-designers show the least similarity in memory structure (.59) (see Table 1).

	NON-DESIGNERS	DESIGNERS
NON-DESIGNERS	.66	.59
DESIGNERS		.74

Table 1 - Similarity Within and Between Memory Structures of Non-Designers and Designers

The two measures of organization in memory structures - height and possible recall orders (PRO) showed no significant difference between non-designers (6.3 and 5.3) and designers (6.0 and 4.5). Designers are apparently no more organized in their memory structures than non-designers.

5.2 Within Database Designs

The designs were all very similar, with most having four entities, the same unique identifier for each entity (see Table 2); and roughly the same relationships among entities and the same attributes for each entity.

Entities

Designer #1	Supplier, Order, Line-Item, Part
Designer #2	Supplier, Order, Line-Item, Part
Designer #3	Supplier, Order, Line-Item, Part, Quotation
Designer #4	Supplier, Order, Line-Item, Part, Quotation, Placement, and Supplier-Line-Item

Unique Identifiers (for the four common entities)

Designer #1	Supplier Number, Order Number, Part Number + Quantity Ordered, Part Number
Designer #2	Supplier Number, Order Number, Part Number + Order Number, Part Number
Designer #3	Supplier Number, Order Number, Part Number + Order Number, Part Number
Designer #4	Supplier Number, Order Number, Part Number + Order Number, Part Number

Table 2 - Comparisons of Database Designs:

Entities and Unique Identifiers

5.3 Between Database Designs and Memory Structures

The interesting result concerned comparisons of the database designs to the memory structures of the designers. When entities and sub-trees are compared we find that of the 20 entities in the four designs, 17 corresponding sub-trees or chunks can be found in the memory structures (see Table 3).

<u>Designer</u>	<u>Entity</u>	# of elements in entity	# of elements in sub-tree
1	Supplier	4	4
	Order	5	4
	Line-Item	1	1
	Part	5	4
2	Supplier	4	4
	Order	3	3
	Line-Item	4	3
	Part	5	5
3	Supplier	4	4
	Order	5	5
	Line-Item	2	1
	Part	4	3
	Quotation	2	1
4	Supplier	4	4
	Order	5	5
	Line-Item	1	-
	Part	5	5
	Quotation	1	-
	Placement	1	-
	Supplier-Line-Item	2	1

Table 3 - Comparing Designs and Memory Structures:
Entities and Sub-Trees, Number of Elements

The second basis for comparing database designs with memory structures, namely unique identifier and first item recalled in a chunk, yielded results which support the proposed model. In most

cases where the entity had a corresponding chunk, the first item recalled in the chunk was the unique identifier (see Table 4). Omitting those unique identifiers which were made of a concatenation of two or more items, nine unique identifiers appeared first in the chunk and the other three appeared either first or second. (note: An item can appear first or second in a chunk if it is sometimes recalled first and sometimes recalled second.)

Two expected results were not supported by the data; 1)relationships among entities were not reflected as the relationships among chunks and 2)ordering within entities did not correspond to ordering within chunks.

<u>DESIGNER</u>	<u>UNIQUE IDENTIFIER</u>	<u>ORDER OF APPEARANCE OF UNIQUE ID IN CHUNK</u>
1	SUPPLIER NUMBER	1st or 2nd
	ORDER NUMBER	1st or 2nd
	PART NUMBER	1st
2	SUPPLIER NUMBER	1st
	ORDER NUMBER	1st
	PART NUMBER	1st
3	SUPPLIER NUMBER	1st
	ORDER NUMBER	1st
	PART NUMBER	1st
4	SUPPLIER NUMBER	1st or 2nd
	ORDER NUMBER	1st
	PART NUMBER	1st

Table 4 - Comparing Database Designs with Memory Structures: Order of Appearance of Unique Identifier in Corresponding Chunk

5.4 Memory Structures and Pre-Memorization Organizations - Non-Designers

For the non-designers their pre-memorization organizations and memory structures followed a process view of data with roughly the same 'groups' (see Table 5). The pre-memorization

organizations tended to follow the steps in the ordering process, whether manually ordering from a catalog or processing data in a computer program.

<u>NON-DESIGNER</u>	<u>MEMORY STRUCTURE CHUNKS</u>	<u>PRE-MEMORIZATION ORGANIZATION</u>
1	ORDER SUPPLIER PART QUANTITY DELIVERY COST	ORDER PART SUPPLIER QUANTITY DELIVERY COST
2	ORDER SUPPLIER PART QUANTITY+DELIVERY COST	ORDER SUPPLIER PART QUANTITY+DELIVERY COST
3	PART SUPPLIER ORDER DELIVERY (unclear chunks)	QUANTITY COST PART SUPPLIER ORDER DELIVERY
4	QUANTITY SUPPLIER DELIVERY COST	ORDER PART QUANTITY SUPPLIER DELIVERY COST

Table 5 - Comparing Memory Structures and Pre-Memorization Organizations for Non-Designers

This extensive analysis of the data from this study has included nearly every conceivable combination of comparisons of the three groups of data - database designs, memory structures, and pre-memorization organizations within each data group and to other data groups. With this background we can now discuss the implications of these results to the field of database design and to further research on expertise.

VI. DISCUSSION OF RESULTS

Two main results can be seen from the above data analysis - 1) designers and non-designers differ in their views of data and 2) the procedures used to uncover similarity between memory structures and database designs yielded results which are useful for validating models of the mental processes of database design.

The two views of data were taken by the two groups of subjects. In short designers take a natural view of data, grouping data associated with real-world objects together, independent of processing; non-designers take a usage view of data, grouping data together as it is used during manual or data processing. This corresponds with Hoffer's (1982) findings, where he found that non-designers, actually business school students, frequently take a data-flow or processing view of data (1982). This lends support to the understanding of expertise in other disciplines. Experts not only know more but have that knowledge better organized (Larkin, McDermott, Simon and Simon, 1980).

This organization of knowledge that experts display may be both the result of a problem-solving skill and a recognition capability that designers bring to the database design process. The memory structures captured here were of the former variety, i.e. the result of the design process was, among other things, a memory structure. However, it may be that designers bring to the problem environment a rich encyclopedia of prototypical database designs, condensed from their years of experience. Such a model of database designers might look like Figure 3.

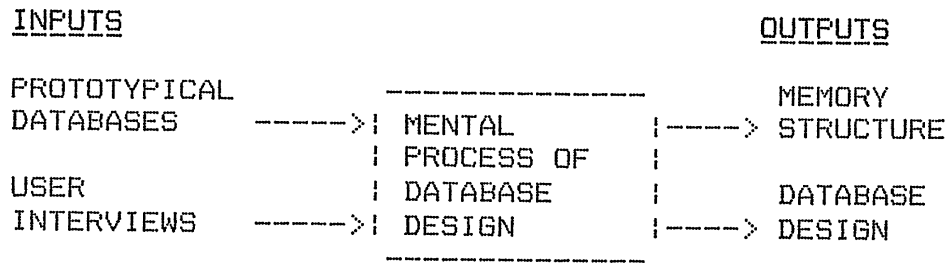


Figure 7 - Another Model of the Mental Process
of Database Design

This investigation into expertise in database designers has not only uncovered designer - non-designer differences along the dimension of memory organization, but has given us a means to explore the mental processes of designers. We expect further research to enable us to develop a mental model of the expert database designer.

Bibliography

Bachman, C.W. "Data Structure Diagrams," Database, Volume 1, Number 2, 1969, pp. 4-10.

Chase, W.G. and Simon, H.A. Perception in Chess, Cognitive Psychology, 1973(a), Volume 4, pp. 55-81.

Chase, W.G. and Simon, H.A. "The Mind's Eye in Chess," in W.G. Chase (Ed.), Visual Information Processing, Academic Press, New York, New York, 1973(b), pp. 216-281.

Codd, E.F. "A Relational Model of Data for Large Shared Data Banks," Communications of the ACM, Volume 13, Number 6, June 1970, pp. 377-387.

Database Design Inc., Logical Data Modeling, Ann Arbor, MI, 1984.

de Groot, A. "Perception and Memory Versus Thought: Some Old Ideas and Recent Findings," in B. Kleinmuntz (Ed.), Problem Solving, Wiley, New York, New York, 1966.

Hoffer, J.A. "An Empirical Investigation into Individual Differences in Database Models," Proceedings of the Third International Conference on Information Systems, Dec. 1982, pp. 153-167.

Larkin, J., McDermott, J., Simon, D.P., and Simon, H.A., "Expert and Novice Performance in Solving Physics Problems," Science, Volume 208, June 1980, pp. 1335-1342.

Martin, J. Computer Data Base Organization, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1977.

McKeithen, K.B., Reitman, J.S., Rueter, H.H., and Hirtle, S.C. "Knowledge Organization and Skill Differences in Computer Programmers," Cognitive Psychology, Volume 13, 1981, pp. 307-325.

Miller, G.A. "The Magic Number Seven Plus or Minus Two: Some Limits on Our Capacity for Processing Information," Psychological Review, Volume 63, 1956, pp. 81-97.

Reitman, J.S. "Skilled Perception in Go: Deducing Memory Structures from Interresponse Times," Cognitive Psychology, Volume 8, 1976, pp. 336-356.

Reitman, J.S. and Rueter, H.R. "Organization Revealed by Recall Orders and Confirmed by Pauses," Cognitive Psychology, Volume 12, Number 4, 1980, pp. 554-581.

Simon, H.A. "How Big is a Chunk?" Science, Volume 183, 1974, pp. 482-488.

Teichroew, D. and Hershey, E.A. "PSL/PSA: A Computer Aided Technique for Structured Documentation and Analysis of

Information Processing Systems," IEEE Transactions in Software Engineering, Volume SE-3, Number 1, 1977, PP. 41-48.

Teorey, T.J. and Cobb, R.E. "Functional Specifications for a Database Design and Evaluation Workbench," Working Paper 82DE1.15, Information Systems Research Group, The University of Michigan, Ann Arbor, MI, July 1982.

Teorey, T.J. and Fry, J.P. Design of Database Structures, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1982.