

Washington University in St. Louis
Washington University Open Scholarship

All Computer Science and Engineering Research

Computer Science and Engineering

Report Number: WUCS-80-04

1980-09-01

Design Studies Suggested by an Abstract Model for Medical Information System

Authors: Jerome R. Cox Jr., Takayuki D. Kimura, P. Moore, Will D. Gillett, and Mishell J. Stucki

We have developed a formal model of a database system that is unusual in that it has the ability to represent information about its own structure and to insure semantic consistency. The model distinguishes general laws from instances of events and objects, but many of its mechanisms serve both categories of information. The model form a substrate upon which an information structure appropriate to neonatology is being developed. Some example queries are shown and a design study for an associative memory suggested by the model is described briefly.

Follow this and additional works at: http://openscholarship.wustl.edu/cse_research



Part of the [Computer Engineering Commons](#), and the [Computer Sciences Commons](#)

Recommended Citation

Cox, Jerome R. Jr.; Kimura, Takayuki D.; Moore, P.; Gillett, Will D.; and Stucki, Mishell J., "Design Studies Suggested by an Abstract Model for Medical Information System" Report Number: WUCS-80-04 (1980). *All Computer Science and Engineering Research*. http://openscholarship.wustl.edu/cse_research/882

**Design Studies Suggested by an Abstract Model
for a Medical Information System**

**J. R. Cox, Jr., T. D. Kimura, P. Moore,
W. D. Gillett and M. J. Stucki**

WUCS-80-05

September 1980

**Department of Computer Science
Washington University
Campus Box 1045
One Brookings Drive
Saint Louis, MO 63130-4899**

This paper was presented at the Fourth Annual Symposium on Computer Applications in Medical Care and published in the proceedings of that conference.

DESIGN STUDIES SUGGESTED BY AN
ABSTRACT MODEL FOR A MEDICAL INFORMATION SYSTEM

J. R. Cox, Jr., T. D. Kimura, P. Moore, W. Gillett, M. J. Stucki

Department of Computer Science and Biomedical Computer Laboratory
Washington University, St. Louis, Missouri 63130

Abstract

We have developed a formal model of a database system that is unusual in that it has the ability to represent information about its own structure and to insure semantic consistency. The model distinguishes general laws from instances of events and objects, but many of its mechanisms serve both categories of information. The model forms a substrate upon which an information structure appropriate to neonatology is being developed. Some example queries are shown and a design study for an associative memory suggested by the model is described briefly.

Introduction

This paper reports progress toward the construction of a formal model of a database system. The model, called the Abstract Database System (ADS),^{1,2} is concise and self-describing, and forms the basis for the development of application-oriented or vernacular models.³ The major strength of ADS is the ability it gives the user to express and enforce semantic constraints on the information stored. This is achieved by providing each database name with a resident descriptor that specifies constraints that can be associated with that name, constraints that can be either local or global. The database interpretation mechanism rejects update requests that conflict with a descriptor.

Semantic consistency is of special importance in medical information systems because of the natural diversity of organizational approaches, because of the disaggregated nature of medicine into specialties and subspecialties, because of the variation in times at which different application areas mature, and because of the constant growth and change in medical knowledge. A mature methodology for the design of medical information systems will be required to amalgamate previously independent, but related, systems; to expand dramatically a small system associated with a

This work was supported in part by the National Center for Health Services Research under Grant HS03792 and the National Institutes of Health under Grant RR00396.

clinically successful methodology; and to create a new system to aid in medical decision-making upon the introduction of a new diagnostic or therapeutic procedure.

As the nation's morbidity continues to shift from acute to chronic disease, the duration of illnesses for an increasing number of patients will be a greater fraction of their lifetimes (e.g., diabetes, hypertension). The quality of the medical information stored about a patient must be maintained throughout the course of his illness. Furthermore, throughout this period we must insure the ability to interpret this information correctly despite intervening improvements in medical knowledge. We are far from the implementation of information systems that can assure the quality of information for a lifetime. In fact, current medical systems, like hospitals themselves, seem to be in a constant state of flux. It is our goal, through the development of ADS, to provide a logical foundation for the development of systems that can endure.

Data Models

It is the logical model of the database (data model for short) that holds the key to the development of enduring database systems. This model must be able to survive multiple technological eras, must be able to help the user maintain the quality of stored data, and must provide a means for amalgamation, growth, and change of the database system.

Smith⁴ emphasizes the distinction between the logical structure of information and the physical representation of information. The structure embodies the data semantics and is associated with the understanding of the information, while the representation is associated with implementation technologies and administrative policies. Today's data models⁵⁻⁷ relegate the enforcement of semantic constraints to the programs that update the database. Since these programs lie outside the data model, it follows that current database systems are incapable of taking responsibility for data integrity.⁸ Furthermore, today's data models include representational issues, and this makes it difficult to adapt current systems to technological change.

Because the diversity of approaches to the organization of medical information makes unlikely the acceptance of a single data model, an enduring information system must provide for the exchange of information among users with differing data models. Toward this end, we have adopted the approach of Kent³, who separates the data model into a primary and a secondary model. The secondary model corresponds to the traditional data model used to describe the semantics of a particular application area, while the primary model provides a basic set of building blocks from which various secondary models can be implemented. Our strategy is to base an information system on a single primary model (ADS) and to use that model to support the various secondary models needed by system users. The common foundation can be used to assure a semantically correct interchange of data among the secondary models.

An important characteristic of ADS is that it incorporates unstratified interpretation⁹ of database names. That is, a name can mean itself or what it designates and the model's interpretation mechanism can interpret a name either way depending on the context in which the name appears. Gorn⁹ introduced the usage of the term unstratified in the above sense and Abrial¹⁰ and Laine¹¹ have used the notion in their models. We believe that unstratified interpretation is an important property for a primary model.

We report below some design studies using ADS to construct a preliminary version of a secondary model for a neonatology database.

A Neonatology Database

We have chosen neonatology for the development of a secondary model because it contains, in microcosm, much of the complexity of medicine in general and because the pace of the development of disease and its cure is accelerated.

Elsewhere in these proceedings, Maurer et al.¹² have described a system for organizing data from a Newborn Intensive Care Unit. This operational and useful system has provided us with a rich source of ideas that have helped shape ADS.

Figure 1 is an informal abstraction of a small portion of the in-hospital information associated with the Newborn Intensive Care Unit at St. Louis Children's Hospital. Table 1 defines the abbreviations used in Figure 1. The meaning of the arcs shown between the ovals is that some known, but unspecified, relationship exists between the abstract ideas represented

Table 1

IHI	Inhospital information
PUL	Pulmonary
CARD	Cardiovascular
+O2	Hypoxemia
PIA	Pulmonary interstitial air
PDA	Patent ductus arteriosus
MAP	Meconium aspiration pneumonia
HMD	Hyaline membrane disease
+O2PUL	Hypoxemia secondary to a pulmonary problem
+O2CARD	Hypoxemia secondary to a cardiovascular problem
PIAPUL	Pulmonary interstitial air secondary to a pulmonary problem
+O2PDA	Hypoxemia secondary to PDA
PIAMAP	Pulmonary interstitial air secondary to MAP
+O2MAP	Hypoxemia secondary to MAP
+O2HMD	Hypoxemia secondary to HMD
PIAHMD	Pulmonary interstitial air secondary to HMD

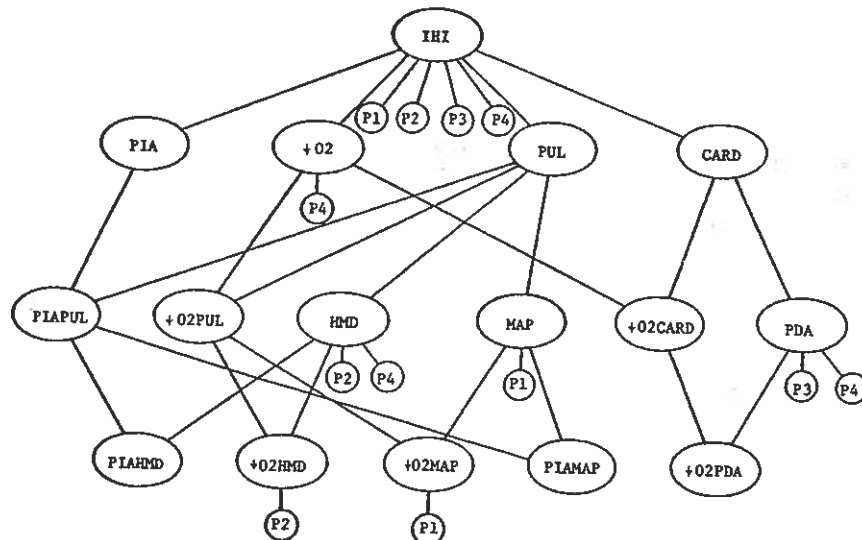


Figure 1: Abstraction of certain in-hospital information

by the ovals. We defer a careful description of these relationships until after we have presented informally the abstract ideas named in Figure 1.

The abbreviations P1, P2, P3, and P4 stand for the four patients in the database, diagnosed with HMD (P2, P4), MAP (P1), and PDA (P3, P4). P2 has the complication hypoxemia (+O2) of HMD, P1 has the same complication (+O2) of MAP, and P4 has hypoxemia (+O2) but without a causal relationship to HMD or PDA. Only two areas (PUL, CARD) are shown in this portion of the inhospital information. Those complications (PIAPUL, +O2PUL) and problems (HMD, MAP) associated with the pulmonary system are shown linked to PUL. A complication of a specific problem (e.g., +O2HMD) is linked to both +O2PUL and HMD.

Other organizations of these abstract ideas, as well as different selections of the abstract ideas themselves, are possible. The organization shown in Figure 1 is, however, consistent with that shown in Tables 2 and 3 of Maurer et al.¹² Below we use examples drawn from Figure 1 to aid in the exposition of ADS.

An Overview of ADS

In its most general form, ADS is a state machine (Figure 2a). All possible commands constitute the set of input symbols and all possible responses constitute the set of output symbols. The current information state will be called i ; the interpretation function ϕ is both the next state function and the output function. Responses and updates are both determined from the current command and the current state i .

A single sequence of commands is presented to ADS. Each command is derived from either a user's declaration or query. If after interpretation by ϕ the command is consistent with the current state i , it is accepted (otherwise rejected). If the command represents a user declaration, it will be recorded. If the command represents a user query, a response will be constructed and displayed to the user without changing i . Thus, the current information state i in ADS corresponds to the usual concept of database state.

A number of variations of ADS can be defined, each with its own base language B in which the command set, the response set, and the information state are represented. Our formal definition of ADS^{1,2} chooses the set of all binary trees for B since it is the simplest set having the required properties and, also, it is a well-known set. Here we will suppress the formal nature of B , using instead an informal and easily interpretable language based on first order logic.

The base language B , a set of expressions, is a part of the universe of discourse for the model, i.e., what the user can name and describe by an expression in B . The universe of discourse is called the objects in ADS. There are three categories of objects: elements (an expression

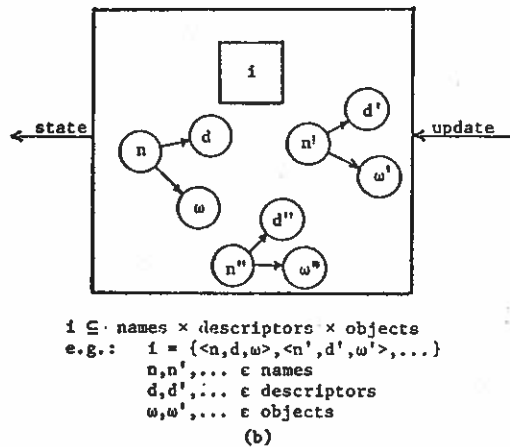
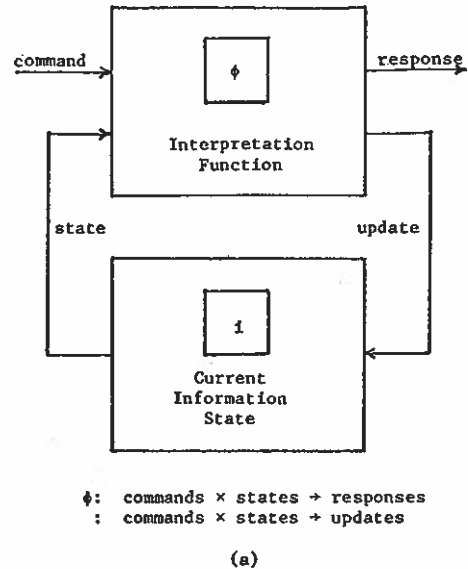


Figure 2: ADS viewed as a state machine

in B), assertions (a logical value, truth or falsehood) and sets (a set of expressions in B). Mathematically, the universe of discourse, Ω , can be defined as:

$$\Omega \cong 2 \cup B \cup 2^B, \quad (1)$$

where $2 \cong \{\text{truth, falsehood}\}$, B is the set of all elements, and 2^B is the power set of B .

Some expressions in B are used for naming objects, and some are used for describing objects. We call them the names and the descriptors, respectively. An information state is a collection of triples of the form

$$\langle \text{name, descriptor, object} \rangle$$

as indicated in Figure 2b. The functions that associate a descriptor with a name and an object with a name are called μ and τ , respectively. That is, for the triple $\langle n, d, \omega \rangle$ we have $\mu(n) = d$ and $\tau(n) = \omega$.

At this point it is helpful to introduce two notions: the intension and the extension of a name. With respect to ADS, the intension of a name is what can be denoted by the name, whereas the extension of a name is what is denoted by the name. The descriptor gives us a means for determining whether an object can possibly be associated with a name and thus implicitly gives the intension. The set associated with a set name, i.e., the intension of the set name, may not be finite (e.g., the set of integers), but it is possible, using the capability of the model for recursive evaluation, to write down a descriptor that can be used to identify any instance of the set with that name.

The extension of a name, on the other hand, represents the user's declaration regarding his view of the world. The set of elements extensionally associated with a set name must be finite since each represents a known element in the intension. The extension of a name n is $\tau(n)$ which must be a subset of the intension of n .

In general, intensional declarations made by the user represent general facts and are manifest in the information state by μ . Extensional declarations represent specific facts and are realized in the information state by τ .

Extensional declarations are required by ADS to be consistent with intensional ones. It is this property of ADS that supports the automatic checking of semantic constraints. Inconsistent declarations are detected by ADS, and it is anticipated that programs external to ADS will assist in making the user's declaration consistent with the user's previous declarations. Alternatively, such programs could assist the user in revising the relevant intensional declarations.

In Figure 3, the relationships between names (N), descriptors (D), and objects (Ω), are shown for each of the three categories of objects: assertions, elements, and sets. Note that for an arbitrary name n the extension is $\tau(n)$ and the intension is $\phi_D \mu(n)$. Here τ and μ are known for names of assertions (N_a), names of elements (N_e), and names of sets (N_s). The function μ leads to descriptors of assertions (D_a), descriptors of elements (D_e), and descriptors of sets (D_s). The range of τ for each case must be a set of finite objects. The function ϕ_D is a restriction of ϕ , mapping descriptors into their intension.

Some examples of each of the three categories of objects are shown in Table 2. The element name THISYR has an intension that consists of all integers as indicated by its descriptor. The determination of the occurrence of meconium staining (MECSTAIN) is recorded with either a yes, no or unknown value. Should it become necessary, a fourth value, such as marginal, may be added to the extension. The intension of MECSTAIN is the set of all names which is a subset of B . The set name DATES has an intension that consists of all triples of integers satisfying constraints associated with the number of months in a year, days in a month, and years from 1975 to the present. The extension of DATES is the null set.

The first patient's record is identified by the element P1. The intension for P1 is a triple with components from the sets: names, DATES, and MECSTAIN. Note that the extension of MECSTAIN influences the intension of P1. Here, the meaning of DATES is the newborn's birth date. These patients are collected in the set PTS which if it had been defined first could have been used to simplify the descriptors for P1, P2, P3, and P4.

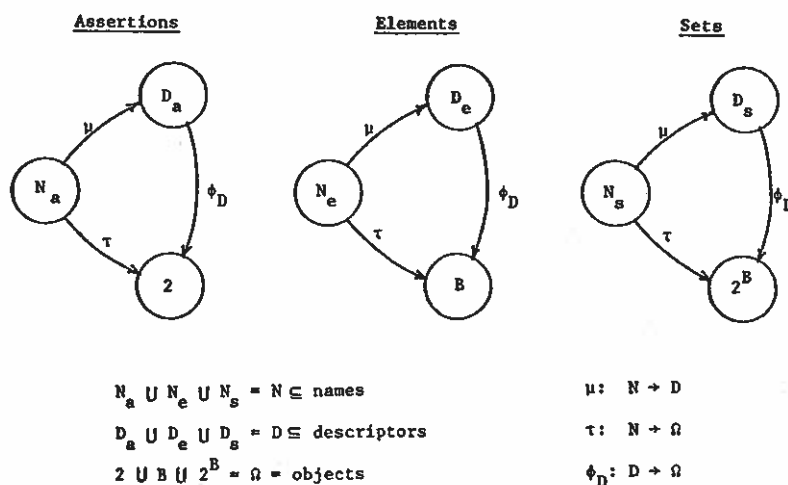


Figure 3: Relations between names, descriptors, and objects in the three categories of objects

Table 2

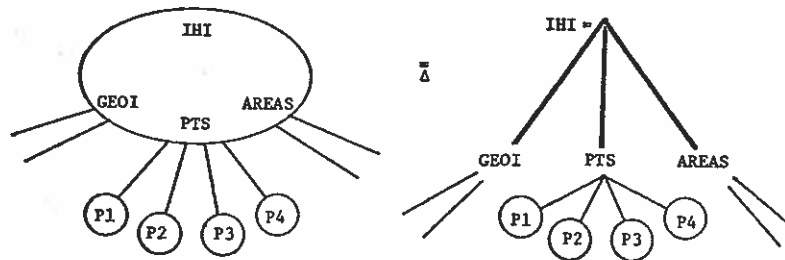
Name: n	Category	Descriptor: $\mu(n)$	Category	Extension: $\tau(n)$
THISYR	N_e	$\langle x \in \text{integers} \rangle$	D_e	$\langle 80 \rangle$
MECSTAIN	N_s	$\{x \in \text{names}\}$	D_s	$\{\text{yes,no,unknown}\}$
DATES	N_s	$\{\langle x,y,z \rangle \in \text{integers}^3 \mid (x \leq 12) \wedge (y \leq 31) \wedge (75 \leq z \leq \tau(\text{THISYR}))\}$	D_s	$\{ \}$
P1	N_e	$\langle x \in \text{names} \times \text{DATES} \times \tau(\text{MECSTAIN}) \rangle$	D_e	$\langle \text{Jones}, \langle 6,4,80 \rangle, \text{yes} \rangle$
P2	N_e	"	D_e	$\langle \text{Smith}, \langle 7,5,80 \rangle, \text{no} \rangle$
P3	N_e	"	D_e	$\langle \text{Chang}, \langle 7,10,80 \rangle, \text{unknown} \rangle$
P4	N_e	"	D_e	$\langle \text{Brown}, \langle 8,2,80 \rangle, \text{no} \rangle$
PTS	N_s	$\{x \in \text{names} \mid \tau(x) \in \text{names} \times \text{DATES} \times \tau(\text{MECSTAIN})\}$	D_s	$\{P1,P2,P3,P4\}$
UNIQUENESS	N_a	$(\forall x,y \in \tau(\text{PTS})) (\tau(x)=\tau(y) \Rightarrow x=y)$	D_a	$\{\text{truth}\}$

No patient should have more than one record. This constraint is expressed as an assertion named UNIQUENESS. Here, we have used parentheses to identify the descriptors and extensions for assertions, in a fashion parallel to the use of set braces to identify the descriptors and extensions for sets; angle brackets are used in a similar way to identify elements. The names in capitals are all in N .

There is a subtle difference between the items in the second and third columns of Table 2. For example, $\mu(\text{MECSTAIN})$ is a symbol string that requires interpretation to obtain the intension; in contrast, $\tau(\text{MECSTAIN})$ is a symbol string that requires no further interpretation. It is always true that the extension is finite, but in this case the intension is not. Thus, in Table 2 the descriptor $\mu(n)$ is uninterpreted. It is a symbol string and not the set that it denotes.

Vernacular Model for Neonatology

We are now in a position to develop a vernacular (or secondary) model for a portion of the in-hospital information from the neonatology database. In Figure 4 we define a set named RECOGNIZERS. The elements in this set are 3-tuples of names from N_s . Like other objects in ADS, the set RECOGNIZERS will have a significance in the user's mind that can only be captured in part within the model. To us, the word recognizer is a means for the organization of information, used somewhat analogously to the database community's use of the word relation. Both words have a precise definition, but in each case, much more is connoted by our experience with the abstract idea than the word denotes. We say here only that a recognizer can determine whether something is an instance to be associated with an abstract idea or not. The examples to follow should help to elucidate this notion.



Name: n	Descriptor: $\mu(n)$	Extension: $\tau(n)$
RECOGNIZERS	$\{x \in \tau(N_s)^3\}$	$\{\text{IHI}, \text{PUL}, \text{MAP}, \dots\}$
IHI	$\langle x \in \text{RECOGNIZERS} \rangle$	$\langle \text{GEOI}, \text{PTS}, \text{AREAS} \rangle$
GEOI	$\{x \in \text{names}\}$	$\{\text{PIA}, +02\}$
PTS	$\{x \in \text{names}\}$	$\{P1, P2, P3, P4\}$
AREAS	$\{x \in \text{names}\}$	$\{\text{PUL}, \text{CARD}\}$

Figure 4: IHI recognizer

The abstract idea inhospital information can be represented by the element IHI. Instances associated with inhospital information are: a general event of interest that occurs during an infant's hospital stay (in GEOI), a record of a patient (in PTS), and an area that is used to classify medical problems, therapies, and procedures (in AREAS).

The heavy lines at the right side of Figure 4 signify the extension of an element name while the light lines signify the extension of a set name. Furthermore, with regard to the concepts of aggregation, classification and generalization, the three kinds of abstraction introduced by Smith⁴ and Smith and Smith,¹³ the heavy lines correspond to aggregation, the light lines correspond to classification, and generalization is contained within the concept of recognizer.

A recognizer similar to that for IHI can be defined for the concept PUL (Figure 5). It recognizes instances of complications of a pulmonary problem (PULCOMP), instances of patients with pulmonary problems (PULPTS), and instances of specific pulmonary problems (PULPROBS). The recognizer is quite similar to that for IHI, but the object named PULPTS introduces a new situation. Note that there is no extension for PULPTS. The set of patients is found indirectly by interpretation of the intension. (The step-by-step interpretation of the descriptor will be discussed below.) Here we only note that the notation $y[2]$ stands for the second component of y and assumes that y is an element with two or more components.

The recognizer for the concept MAP is shown in Figure 6. Here MAPPTS is also a subset of PTS with an extension consisting of the singleton set $\{P1\}$. A user made the extensional declaration that $P1 \in \tau(\text{MAPPTS})$, and this declaration was consistent with the requirement that the objects in MAPPTS be also in PTS and the observation of meconium staining ($P1[3]=\text{yes}$).

By continuing the process begun in Figures 4, 5 and 6, all of the concepts listed in Table 1 can be defined explicitly, the relationships expressed by the arcs in Figure 1 are given definite meaning, and a portion of the information structure for neonatology is defined. This information structure can be augmented to include additional areas and problems (see Tables 1, 2 and 3 in Maurer et al.¹²). Note that this process requires only augmentation of the extensions of names already defined and requires no modification of their descriptors. Thus, the information structure can grow gracefully as new patients and new concepts and their recognizers are added.

We expect to improve our vernacular model for neonatology to include concepts (like time) not presently considered. Different vernacular models will be necessary for other specialties. In both cases, we believe that ADS will provide useful building blocks for these new models. The common substrate provided by ADS makes translation among models possible.

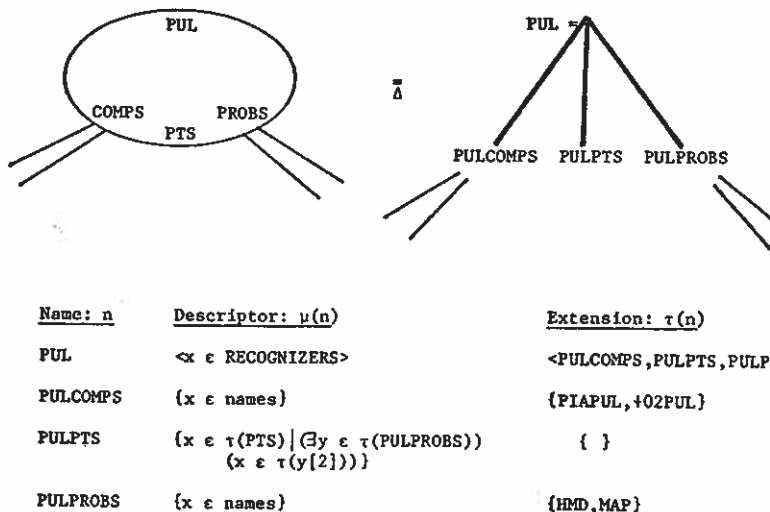


Figure 5: PUL recognizer

Examples of Queries

Examples of some simple queries will help set the stage for a discussion of the use of associative memories with ADS. Suppose we ask for the record of patient P1,

$$P1? \xrightarrow{\tau} \langle \text{Jones}, \langle 6, 4, 80 \rangle, \text{yes} \rangle \quad (2)$$

The extension of P1 can be obtained directly from τ . Similarly, the patients with meconium aspiration pneumonia are given by,

$$\text{MAPPTS?} \xrightarrow{\tau} \{P1\} \quad (3)$$

and those with hyaline membrane disease by,

$$\text{HMDPTS?} \xrightarrow{\tau} \{P2, P4\} \quad (4)$$

A more complicated example seeks the patients with pulmonary problems,

PULPTS $\xrightarrow{\mu}$

$$\{x \in \tau(\text{PTS}) \mid (\exists y \in \tau(\text{PULPROBS})) (x \in \tau(y[2]))\} \quad (5)$$

$$\{x \in \tau(\text{PTS}) \mid (\exists y \in \{\text{HMD}, \text{MAP}\}) (x \in \tau(y[2]))\} \quad (6)$$

In Expression 5, μ obtains the descriptor for PULPTS. In Expression 6, the extension of PULPROBS is obtained through τ . Note that the extension of PTS could equally well have been obtained first, but either way the final interpretation of PULPTS would be identical. Next, the interpretation of the righthand side of Expression 6 yields an equivalent expression,

$$\{x \in \tau(\text{PTS}) \mid (x \in \tau(\text{HMD}[2])) \vee (x \in \tau(\text{MAP}[2]))\} \quad (7)$$

$$\{x \in \tau(\text{PTS}) \mid (x \in \tau(\text{HMDPTS})) \vee (x \in \tau(\text{MAPPTS}))\} \quad (8)$$

where Expression 8 is obtained by finding the extensions of HMD and MAP and then obtaining their second components.

Primitive objects (elements whose extensions contain no names in N), such as P1, P2, P3 and P4, may often constitute the major fraction of the information stored. Non-primitive or abstract objects (e.g., PULPROB, HMD, HMDPTS) often occupy a smaller fraction of the information stored and correspond to what is usually thought of as index information. So far in the query example only abstract objects have been encountered. The next step in the interpretation of PULPTS? requires access to primitive objects. One choice is to obtain the extension of PTS and then qualify each patient through evaluation of the logical expression in Expression 8. This plan yields

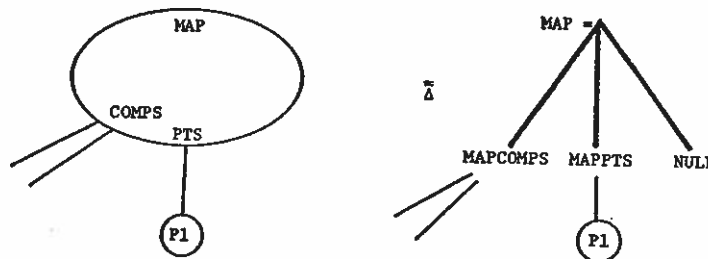
$$\{x \in \{P1, P2, P3, P4\} \mid (x \in \tau(\text{HMDPTS})) \vee (x \in \tau(\text{MAPPTS}))\} \quad (9)$$

$$\xrightarrow{\phi} \{P1, P2, P4\} \quad (10)$$

Such a choice would yield good performance whenever the number of patients in PTS is substantially less than the number of instances of patients with pulmonary problems. Alternatively, the extension of HMDPTS and MAPPTS could be obtained first, the set union taken and the members of the resulting set each qualified as belonging to PTS.

$$\{x \in \tau(\text{PTS}) \mid (x \in \{P2, P4\}) \vee (x \in \{P1\})\} \quad (11)$$

$$\xrightarrow{\phi} \{P1, P2, P4\} \quad (12)$$



Name: n	Descriptor: $\mu(n)$	Extension: $\tau(n)$
MAP	$\langle x \in \text{RECOGNIZERS} \rangle$	$\langle \text{MAPCOMPS}, \text{MAPPTS}, \text{NULL} \rangle$
MAPCOMPS	$\{x \in \text{names}\}$	$\{\text{PIAMAP}, +02\text{MAP}\}$
MAPPTS	$\{x \in \tau(\text{PTS}) \mid x[3]=\text{yes}\}$	$\{P1\}$
NULL	$\{x \mid x^*x\}$	$\{\}$

Figure 6: MAP recognizer

The same result is obtained, of course, but the method of Expression 11 is to be preferred if the number of patients is substantially more than the number of instances of patients with pulmonary disease.

Note that ADS is neutral with regard to the best order in which to evaluate a descriptor. This descriptive, rather than prescriptive, characteristic is desirable because technological and usage factors may alter the method of choice even though the semantics of the information structure remain unchanged.

Associative Memories and ADS

The processes involved in descriptor evaluation -- plus other ADS considerations -- have led us to believe that a particular type of associative memory (AM) would be appropriate to information systems based on ADS. We are currently refining a functional specification for such a memory and are also investigating VLSI designs for implementing it. Although our research in this area is not yet complete, the following description will serve as an introduction.

The associative memory under consideration is a storehouse for ordered pairs, the components of which are variable-length character strings. Figure 7 shows the block diagram for the memory and also gives the input/output specification for three types of retrieval command. The command $\langle a, b \rangle ?$ returns the tuple $\langle a, b \rangle$ if that tuple is in the memory. The command $\langle a, * \rangle ?$ returns all tuples whose first component is "a", and the command $\langle *, b \rangle$ returns all tuples whose second component is "b". The utility of these commands is illustrated below.

Following the example from neonatology, a fragment of the contents of the AM might be,

```
AM = { <PTS, P1>, <PTS, P2>, ..., <PTS, P4>,
      <P1, <Jones, <6, 4, 80>, yes>>, ...,
      <UNIQUENESS, truth>, ...,
      <MAP, <MAPCOMPS, MAPPTS, MAPPROBS>>,
      <MAPCOMPS, PIAMAP>, <MAPCOMPS, +O2MAP>,
      <MAPPTS, P1>, <HMDPTS, P2>,
      <PUL, <PULCOMPS, PULPTS, PULPROBS>>,
      <PULCOMPS, PIAPUL>, <PULCOMPS, +O2PUL>,
      <PULPROBS, HMD>, <PULPROBS, MAP>, ... } (13)
```

Expression 13 suggests that all τ information can be stored in the AM. Elements like the record for P1 are stored with the name in the first component and the extension in the second component of the pair. Assertions are stored similarly. Sets



$$\langle a, b \rangle ? \rightarrow \{x \in AM \mid x = \langle a, b \rangle\}$$

$$\langle a, * \rangle ? \rightarrow \{x \in AM \mid (\exists y)(x = \langle a, y \rangle)\}$$

$$\langle *, b \rangle ? \rightarrow \{x \in AM \mid (\exists y)(x = \langle y, b \rangle)\}$$

Figure 7: Associative memory retrieval operations

like PTS require a separate ordered pair for each member of the set. The set name is the first component of the pair and the set member is the second component. By examination of Expression 13, the reader can verify that the following commands would give the responses shown.

$$\langle \text{PULPROBS}, * \rangle ? \rightarrow \{ \langle \text{PULPROBS}, \text{HMD} \rangle, \langle \text{PULPROBS}, \text{MAP} \rangle \} \quad (14)$$

$$\langle \text{PTS}, * \rangle ? \rightarrow \{ \langle \text{PTS}, \text{P1} \rangle, \langle \text{PTS}, \text{P2} \rangle, \langle \text{PTS}, \text{P3} \rangle, \langle \text{PTS}, \text{P4} \rangle \} \quad (15)$$

These commands return the extensions of PULPROBS and PTS.

The extensions required for Expressions 2, 3, 4, 6, 8, 9 and 11 can be obtained in a similar manner. A different type of operation is used in the final evaluation steps of Expression 10. These steps involve membership tests such as

$$\langle \text{MAPPTS}, \text{P1} \rangle ? \rightarrow \{ \langle \text{MAPPTS}, \text{P1} \rangle \} \quad (16)$$

which determines that P1 is a member of $\tau(\text{MAPPTS})$.

We have carried out some VLSI design studies in the style of Mead and Conway.¹⁴ These studies lead to some very tentative, but promising conclusions. An architecture with parallel comparators examining the contents of circular shift registers is readily suggested. Performance is likely to exceed that for a RAM with some mixtures of commands. Much work remains to be done to determine typical mixtures of commands and representative performance comparisons applicable to database machines.

Conclusions

Design studies with ADS are promising. We plan to continue them and, in addition, take some modest steps toward obtaining experimental results. An implementation of ADS in the language SAIL¹⁵ is already underway. An experimental program to understand access time trade-offs in the associative memory is planned.

From our work so far, five properties of ADS seem significant for medical information systems:

- 1) Semantic checking. The inherent capability to verify the consistency of user's declarations is effectively carried out through the central role played by the resident descriptors that enforce both local and global semantic constraints.
- 2) Unstratified interpretation. A name and descriptor of an object in ADS are themselves objects. Thus, ADS has the capability of unstratified interpretation, one advantage of which is arbitrarily high levels of abstraction. The flexibility thereby obtained is useful in handling the relatively volatile abstract information found in medicine.
- 3) Primary model. The building blocks provided allow the construction of vernacular models tailored to various medical specialties.
- 4) Translation of views. The common view of shared information provided by a common substrate makes possible the automatic translation of one user's view of objects of interest to another user's view.
- 5) Descriptive language. The language of the model allows a description (as opposed to prescription) of queries and declarations. Alternative implementations are possible, encouraging a system design that can survive multiple technological and usage eras.

Two recent papers^{16,17} have emphasized the important role that computers can play in medicine. In both cases, medical information systems are central to this role. Work on information systems that emphasize the five properties outlined above will, we believe, make important contributions to biomedical computing, medicine, and health care.

Acknowledgements

We are indebted to C. Frank Starmer and Richard Dammkoehler for many important ideas that helped lay the foundation for ADS. More recently, Michael Maurer, G. James Blaine, Lewis J. Thomas, and Ken Wong participated in a number of fruitful discussions with one or more of us.

References

1. Cox, Jr., J. R., A Medical Information Systems Design Methodology, Progress Report, Department of Computer Science, Washington University, June 1980.

2. Kimura, T., Cox, Jr., J. R., and Gillett, W., "An Abstract Model of an Unstratified Database System," Working Paper, Department of Computer Science, Washington University, July 1980.
3. Kent, W., Data and Reality, North Holland Publishing Company, Amsterdam, 1978.
4. Smith, J. M., "Comments on the Papers 'A Software Engineering View of Data Base Management' by A. I. Wasserman and 'A Software Engineering View of Data Base Systems' by H. Weber," Issues in Data Base Management, North Holland, 1979, 99-102.
5. Codd, E. F., "A Relational Model of Data for Large Shared Data Banks," Communications of ACM, 13:377-387, June 1970.
6. Data Language/I - Systems/370 DOS/VS, General Information Manual GH20-1246, IBM, White Plains, New York, 1974.
7. National Bureau of Standards Handbook 113, CODASYL, Data Description Language Journal of Development, U. S. Department of Commerce, National Bureau of Standards, Washington, D.C., 1974.
8. Yao, S. B., Bernstein, P. A., Goodman, N., Schuster, S. A., Shipman, D. W., and Smith, D. C. P., "Data-base Systems," Computer, 11:46-60, September, 1978.
9. Gorn, S., "The Identification of the Computer and Information Sciences: Their Fundamental Semiotic Concepts and Relationships," Foundations of Language, 4:339-372, 1968.
10. Abrial, J. R., "Data Semantics," in Data Base Management, ed. J. W. Kimble, Amsterdam, North Holland Publishing Company, 1974, 1-60.
11. Laine, H., Maanavilja, O., and Peltola, E., "Grammatical Data Base Model," Information Systems, 4:257-267, 1979.
12. Maurer, M. M., Moore, P., Mead, C. N., Johnson, J. A., Marshall, R. E., and Thomas, Jr., L. J., "Data Management for Rapid Access to Newborn Intensive-Care Information," in these proceedings.
13. Smith, J. M. and Smith, D. C. P., "Database Abstractions: Aggregation and Generalization," ACM Transactions on Database Systems, 2: 105-133, June 1977.
14. Mead, C. and Conway, L., Introduction to VLSI Systems, Addison-Wesley Publishing Company, Reading, Massachusetts, 1980.
15. Reiser, J. F., SAIL, Stanford Artificial Intelligence Laboratory Report No. STAN-CS-76-574, Stanford California, August 1976.