- 96 -

UNIVERSITE
DE TECHNOLOGIE
DE COMPIEGNE

B.P. 233
60 206 Compiègne cedex
France

PROPERTY DRIVEN DATA BASES

J. Paul A. BARTHES, Michel VAYSSADE
and Monika MIACZYNSKA

Department of Applied Mathematics and
Computer Science

ABSTRACT

In this paper we suggest how one can use a very small number of general operators (functions) to organize a factual database whose information structures are constructed as simple association lists. It is shown how the operators are built around the central concept of property. This approach is derived from experimental work on various database prototypes. It has been influenced by previous work of ABRIAL, LINDGREEN, and some A.I. projects such as TEIRESIAS. The extreme simplicity of the data structure does not prevent addressing such issues as user classes and privacy, or metalevels of representation. Our experience is presented in a somewhat formal framework, but most of the involved operators are direct abstractions of implemented functions. The approach was wed to design and implement a database management system, operating in a small - core slow secondary storage environment, as a part of a low cost multimicroprocessor robot system. We feel that further work along this line will provide efficient solutions to many database problems, due to the central role of the property concept.

Acknowledgment : This work was supported in part by CNRS under ATP granted by the Committee for Industrial Robotics and by the University of Technology of Compiègne.

TABLE DES MATIERES

0. INTRODUCTION

The A.I. community has developped efficient techniques for in core handling of symbolic knowledge representation. On the other hand commercial data bases use extensively secondary storage, but for various reasons suffer from very rigid data structures. Both sides have proposed abstract representation models with the hope to find an efficient solution (ABRIAL, 1974, CODD, 1970, DAVIS, 1978, MC DERMOTT, 1975, LINDGREEN, 1974, MINSKY, 1974). However difficulties still subsist when one has to actually implement the models, in particular when secondary storage is involved. Our recent work aimed at building a low cost multimicroprocessor robot system led us to address this problem in a small-core/large-but-slow secondary storage environment. Our objective was to implement a data base using a 16 bit microprocessor. We then reconsidered the simple property value representation which proved to be surprisingly powerful. Indeed it is possible to build easily a data base management system as an interpretor of a set of properties, each being described in turn as a property - value list (introducing readily a metalevel of description). Then, while processing data, the presence of any given property in the data stream can trigger associated functions, which renders the overall approach somewhat analogous to a production rule system. Hence the name of Property Driven Data Base.

The main characteristics of our approach are

- no distinction between physical and logical models, which avoids the so-called "data independence" problem plaguing most commercial systems (DATE),

- no a priori information (or object) classes ; each entity stands for itself. It is represented as a property list and possesses an internal name. The representation is very similar to the one found in (ABRIAL),

- operators imbedding semantic meaning are attached to properties,
- relational information is implemented at property level,
- access to the data base is done through specific entry points unlike in (MC DERMOTT),
- privacy and user classes are dealt with using the notion of model and related access paths,
- access paths are computed dynamically.

One experience is summarized in this paper organized as follows :

Sections 1 to 4 describe the data structure from a static point of view, sections 5 to 7 introduce general exploring mechanisms, section 8 and 9 introduce new structures for the sake of operational efficiency, section 10 defines user classes, and section 11 gives some considerations about implementation.

1. DEFINITION - ENSEMBLES

We consider 3 distinct ensembles :

E the entity set or set of objects

P the property set or set of attributs

V the value set.

One has to notice that we do not consider any "relation set", relations being implemented through the use of properties.

The property set can be further split into two subsets :

. terminal properties T
. structure properties S

with P = T U S.

This distinction is equivalent to "definitive properties" and "general properties" in (LINDGREEN 1974).