Expert Database Systems (Workshop Review)

Larry Kerschberg (Panel Chair)
University of South Carolina

Panel Participants

Michael Brodie, *Computer Corporation of America*
Charles Kellogg, *MCC Corporation*
D Stott Parker, *UCLA and Silogic, Inc*
Gio Wiederhold, *Stanford University*
Carlo Zaniolo, *MCC Corporation*

ABSTRACT

This panel reports on the Working Group deliberations of the First International Workshop on Expert Database Systems, held at Kiawah Island, South Carolina during October 24-27, 1984 The workshop was sponsored by the Institute of Information Management, Technology and Policy, College of Business Administration, of the University of South Carolina, in cooperation with the ACM Special Interest Groups SIGMOD and SIGART and the IEEE Technical Committee on Data Base Engineering

The Working Groups were charged with exploring research issues related to the various facets of Expert Database Systems The Panel Chair served as Program Chairman for the EDS Workshop and the Panel Members organized and coordinated the Working Groups

The goal of this panel is to review the Working Group discussions and conclusions, and to explore research topics related to Expert Database Systems that span the boundaries of the various Groups

1 INTRODUCTION

The First International Workshop on Expert Database Systems provided a forum to address the theoretical and practical issues involved in making database systems more knowledgeable and supportive of Artificial Intelligence applications

Although the fields of Expert Systems and Database Management Systems are well defined and have had major successes, the concept of "Expert Database Systems" connotes diverse definitions and decidedly different architectures

In order to define, discuss, and elaborate on *Expert Database Systems*, this Workshop brought together individuals from diverse backgrounds — Artificial Intelligence, Database Management, Fuzzy Set Theory, Information Retrieval, Logic and Logic Programming — to merge the collective knowledge and insights of each area into this new and exciting research field

In response to the Workshop's Call for Papers, some 96 papers from 13 countries were submitted Each paper was reviewed by three referees, all of them members of the Program Committee The papers were evaluated and ranked on the basis of their originality, relevance to the Workshop themes, clarity and significance

The Program Committee's invitation policy reflected a desire to keep Workshop attendance reasonably small, yet representative of the major intellectual currents that influence Expert Database Systems The Committee accepted 36 full papers and 30 position papers representing a total of 115 authors Of these 115 authors, 91 were invited A special effort was made to invite graduate students so that they might use the insights gained from the Workshop in their own thesis research

The Workshop Proceedings are contained in a 1000-page two-volume set, and a book entitled *Expert Database Systems* will contain the Keynote Addresses, Full Papers, Plenary Session discussions, and Working Group Reports

2 EDS Working Groups

This panel will focus on the Workshop's Working Group discussions The Working Group titles and coordinators are given below

- Knowledge Base Management Systems — Michael Brodie
- Deductive Reasoning for Expert Database Systems — Charles Kellogg
- Logic Programming and Databases — D Stott Parker, Jr
- Intelligent Database Interfaces — Gio Wiederhold
- Object Oriented Database Systems and Knowledge Systems — Carlo Zaniolo

The Working Groups met in the evenings to discuss topics and listen to featured speakers In the sections that follow we present the major themes and issues of each Working Group

2 1 Knowledge Base Management Systems

The Knowledge Base Management Systems Working Group, chaired by Michael L Brodie, had the following featured speakers Robert Balzer (Information Sciences Institute), Gio Wiederhold (Stanford University), Ron Brachman (Fairchild Lab for AI Research), and John Mylopoulos (University of Toronto)

A major reason for the interest in interactions between AI and Databases is the realization that, on one hand, significant improvements in productivity and functionality of information systems require AI techniques such as those provided by knowledge-directed and reasoning systems On the other hand, practical application of the same AI technology requires progress in systems and efficiency issues such as those addressed by database management research A meeting point for the two technologies is the concept of a **knowledge base management system** (KBMS) which could be defined as

A system providing highly efficient management of large, shared knowledge bases for knowledge-directed systems

KBMSs are being proposed in order to provide friendly environments for the construction, retrieval, and manipulation of large, shared knowledge bases with functionalities such as deductive reasoning, concurrent access, distribution of a knowledge base over several geographic locations error recovery, and security

At this early stage of KBMS development, we might ask four important questions

1 What is a KBMS's potential functionality, knowledge representation capabilities, and architecture?
2 What are the potential requirements and application classes for a KBMS? That is, what services will a KBMS provide to what human user or software system?
3 What important research questions must be answered in order to make KBMSs a reality?
4 What approaches are being taken to the research questions and to KBMSs in general? In particular, which of the following four approaches are being taken to the integration of AI and Database technology to realize KBMSs?
   - A Loosely couple an existing AI system with a DBMS
   - B Extend a DBMS by enhancing the data model with knowledge representation and other AI capabilities
   - C Enhance an AI system with database functionality
   - D Tightly integrate database and AI system functionality including data model and knowledge representation concepts

The current situation for KBMSs is reminiscent of the early days of Database Management when there was an open discussion of desirable features and capabilities for database management systems (DBMSs)

2 2 Deductive Reasoning for Expert Database Systems

This Working Group was chaired by Charles Kellogg of MCC Corporation and had featured speakers Rolf Stachowitz (MCC Corporation), Dana Nau (University of Maryland), Enrique Ruspini (Hewlett-Packard Labs), Richard Tong (AI & DS), and Dan Fishman (Hewlett-Packard Labs)

Most Expert Systems developed up to this point have been strongly influenced by the dictum "In the knowledge resides the power" propounded by Edward Feigenbaum several years ago As a result these systems have usually placed emphasis on the acquisition and representation of knowledge, often at the expense of the reasoning component which has been either ad hoc or limited in scope

We do not argue with the primacy of knowledge acquisition and representation However we do claim especially for *Expert Database Systems*, that a general purpose logic based, deductive reasoning component can be realized that is useful, reasonably efficient, and complete

We take an Expert Database System to be a database system, specifically a relational database system, to which a body of ***application-specific expertise*** is appended From the point of view of Logic, a relational database is a collection of concrete facts constituting a specific state of affairs realized as a finite conjunctive (logical) ***model*** Logic expressions (predicate calculus queries) may then be directly evaluated against such a database structure Application-specific expertise in the form of judgmental knowledge, general laws, etc is formulatable as a (logical) ***theory***, a description of a general or permissible state of affairs as a series of rules Queries in the form of logic expressions posed to this combined system of rules and facts will require a deductive reasoning mechanism to produce answers that are a logical consequence of the contents of the knowledge base

Logic thus provides a powerful description formalism for representing knowledge in both the form of facts (tuples in a relational database) and rules (logic statements in implicational form), and it provides a powerful general mechanism for deriving (logically) correct answers to user questions from facts and rules expressed in this formalism

This working group discussed, among other issues, the following

- Correct reasoning with uncertain knowledge (facts and rules that vary in plausibility),
- Achieving practical efficiency versus realizing theoretical completeness,
- Open versus closed world deductions,
- New deductive reasoning strategies,
- The need for and utility of higher-order nonstandard logics,
- Utility of specialized inference mechanisms for manipulating facts,
- Coping with incomplete fact and/or rule bases, and
- Reasoning with time, events, and cause-effect relationships

2 3 Logic Programming and Databases

This Working Group was chaired by D Stott Parker, Jr , and had featured speakers Michael Carey (University of Wisconsin), Forouzan Golshani (Arizona State University), Matthias Jarke (New York University), Edward Sciore (Boston University) and Adrian Walker (IBM Research, Yorktown Heights)

The interface between Logic Programming and Databases is an important aspect of a much larger phenomenon ***the confluence of the information sciences*** As the emphasis on information processing has grown, previously independent symbolic processing disciplines have begun borrowing concepts heavily from one another

A great deal of excitement is resulting from this confluence Unfortunately, a great deal of confusion is resulting as well, since the fields all refer to similar concepts with different terminology, and use concepts that are ad hoc or lack a solid foundation Logic Programming offers a direction out of the confusion It is unique in that it offers a consolidation of what has already been done, as well as a sound formal basis on which to build predicate calculus with Horn clauses

It is surprising that the merging of Logic Programming and Databases has taken this long to occur The two fields followed parallel lines of development throughout the 1970's, but have largely ignored one another in spite of a great deal of work at the interface

Perhaps only the selection of Prolog by the Japanese FGCS project has precipitated reevaluation of inaccurate perceptions of Logic Programming, such as difficulty of use or of learning (Prolog is taught in some British elementary schools) inferiority to LISP, inefficiency, etc

Much has been written recently about *Knowledge Bases* (also called Inferential Databases, Deductive Databases, and Expert System kernels) These are the information management systems of the future, and will incorporate the results of several decades of work in the Database, Knowledge Representation, Expert System, and Automated Theorem Proving fields They are now being developed directly using Logic Programming systems

Both Databases and Logic Programming have much to obtain from confluence Logic Programming extends Relational Databases with deduction, storage of non-record-oriented information, and the ability to combine schema, metadata and constraints with database facts Logic Programming also provides an elegant and uniform way of implementing views, query languages, and null values

However, Logic Programming systems such as Prolog do not yet have all the qualities one would want of a Database system Prolog does not directly support some types of queries, integrity constraints, schema definition, and so forth Moreover most Prolog implementations are *in-memory* systems *all* information is loaded into memory (possibly virtual memory) before execution Secondary storage media are not used to store and query the current state of the information, unless the underlying operating system uses paging

Clearly there are many issues to be resolved We have chosen to focus on two key issues

(1) *Extending Logic Programming for Database Applications*

How can Logic Programming systems be augmented to support query interfaces that are more responsive to user needs? What ways can Logic Programming be used to support modeling, in particular knowledge representation, data definition, and incomplete information? Which Database notions (transactions, concurrency, sophisticated indexing, and so forth) should be incorporated into Logic Programming?

(2) *Adding Database Storage & Query Capability to Logic Programming*

What techniques are useful in implementation of combined Logic Programming/Database systems? Problems range from the most basic architectural issues to optimization of queries

2 4 Intelligent Database Interfaces

This Working Group was chaired by Gio Wiederhold and had as featured speakers, Michael Brodie (Computer Corporation of America) Francisco Corella (Fairchild Lab for AI Research), Jonathan King (Teknowledge), and Michel Missikoff (IASI-CNR, Rome)

The topic of this Working Group is the Interface between database and knowledge base systems We begin with the premise that the two should interact, but that the degree of integration which is desired is not clear

Having an interface implies that there is a distinction An operational definition for data versus knowledge

> Data can be entered, updated, and verified by clerical personnel Data refers to instances in the real world and admits little ambiguity
>
> Knowledge requires an expert to collect, enter (directly or indirectly), and maintain Knowledge refers to many instances and provides generalizations and abstractions It is not necessarily objectively verifiable

We ignore now the derivable question "Who is a clerk and who is an expert?" Some fields have certification procedures, most do not The extent of blurring over the boundaries is a central determinant in interface discussions

Current State

The two extreme position are represented in practice today

Complete disjointness

> Data analysis on a database is performed by experts using traditional computing techniques The deductions are reviewed and eventually published Experts read the literature and augment the taught knowledge with their own experiences
>
> Knowledge engineers squeeze rules out of experts and build expert systems The expert systems provide advise to folks who are lesser experts or non-experts

Thorough integration

> A single mechanism handles knowledge and data alike The semantic value or power of a data element to affect a knowledge unit is equal to the power that knowledge has to constrain data Imperfect data or imperfect knowledge leads to similar problems in deduction

Discussion points on interfaces Where and How?

1 Where should the interface be?

> Should future systems be near one of the extreme points, or nearer the center? We probably want it all, and will spring for a balanced approach If we do, is it realistic?

Arguments for separation, and hence a formal, relatively low band-width interface are

> Ability to let the database activities proceed independently, gather the data, generate the reports, etc, and in so doing provide a foundation for knowledge extraction

Arguments for a high degree of integration and a high band-width interface are

> The importance of the knowledge in controlling the semantic integrity of the database Assurance that the database and the knowledge base are consistent, so that deductions will not be led astray The power derived from operating on the joint body of information (data/knowledge) and being able to explore a variety of paths The consistency at the users point of view when no distinction of data and knowledge need to be made

2 Can the desired interfaces be implemented?

> How can a balanced systems be implemented? Will a balanced system be able to make users sharing a data-knowledge base happy?

Does a technology exist to have systems which support applications where the desired balance is more to the right or the left?

The issues of bandwidth indicated above now translate into issues of modular design, and the ability to partition the systems Without some partitioning long-term growth in a scientific pattern will be hindered This problem is seen already in traditional operating databases and is a strong disincentive to increased functionality

2 5 Object Oriented Database Systems and Knowledge Systems

This Working Group was chaired by Carlo Zaniolo and had as featured speckers, Hassan Ait-Kaci (MCC), David Beech (Hewlett-Packard Labs), Stephanie Cammarata (University of California, Los Angeles), Larry Kerschberg (University of South Carolina), and David Maier (Oregon Graduate Center)

The concept of objects and object-oriented architectures represent a most interesting unifying trend in the design of both Knowledge Based and Database Systems Versatility and flexibility represent proven virtues of this approach, that has been successfully applied to a wide spectrum of applications and programming environments Important application areas include

(1) Programming Languages

(2) CAD Systems and Engineering Design Databases

(3) Office Information Systems

(4) Knowledge Representation Systems

(5) Database Management Systems

The flexibility of the concept of objects is also demonstrated by its realization in the framework of functional programming languages and Logic Programming

Because of their applicability to different programming environments, it is hoped that objects will supply a basis for unifying and integrating different programming environments The issue of implanting database facilities in an object-oriented programming language was addressed by Maier and Copeland Zaniolo has shown that a better interface between Prolog and a database system can be constructed by adding to Prolog the notion of object identifiers

The objective of this working group is to clarify the principal research issues relating to the concept of objects and object-based system architectures In particular, we want to identify the potentially important roles that objects can play in unifying database systems and knowledge based systems