

## DBMS and AI: Is there any common point of view?

### Moderator:

Michael Stonebraker, University of California

### Panelists:

Daniel Sagalowicz, SRI International

Robert Wilensky, University of California

There have been two recent workshops (Pingree Park and Intervale) that have explored commonality among the points of view of AI, DBMS and Programming Language researchers. On the surface, it would appear that DBMS researchers are building smarter data base systems and that AI researchers are building "expert systems" containing a knowledge base. However, it appears that there is minimal commonality to the points of view expressed by these communities. Therefore, I am dubious of any significant "cross fertilization".

I will illustrate a few differences in points of view that I perceive in the two communities by 3 examples. Consider a collection of facts, e.g.

John loves Mary  
John is a boy  
Boys love girls  
Mary owns a car

AI researchers would organize this data into some sort of semantic network. Nodes in the network would be objects (e.g. John, Mary, boy, girl, car) and labelled arcs would connect these nodes to indicate the desired relationship (e.g. loves, owns, is a). On the other hand, DBMS researchers might organize this data into three relations `owns(object, object)`, `loves (object, object)` and `is_a(object, object)`.

Notice immediately that the DBMS representation is biased toward a large number of instances of a small number of types of facts while the AI representation is biased toward a small number of instances of a larger number of types of facts. It is not obvious how to reconcile these two points of view.

A second difference in point of view is the kind of queries asked. An AI researcher would want to ask the query "Tell me everything about John" and expect an answer "loves Mary, is a boy, and loves girls". The last response requires a simple inference on the part of the responder. DBMS researchers appear to ignore queries of this form when they design query language and typically do not deal with inference at all.

A third difference in point of views concerns rules. So-called expert systems (e.g. Mycin, Prospector) are composed of a data base consisting of a collection of

rules. For example, a medical diagnosis expert might have the following rules:

If patient is green  
then it is Halloween or he is very sick

If the patient is very sick and has trouble breathing  
then administer oxygen

If the patient has trouble breathing and looks well  
then slap him hard on the back

A more sophisticated expert might have several hundred such rules. Basically when an expert system is presented with the input "the patient is green", it attempts to find rules to apply. From the first rule it can conclude that the date is Halloween or the patient is very sick. It could then ask what date it is. On hearing that is not Halloween, it can conclude the patient is very sick and inquire if the patient has trouble breathing, and so forth.

Two concepts appear fundamental to most expert systems: the data base is a collection of rules and the program is basically an inference engine. From a DBMS point of view rules are a close analog to triggers. However, the inclusion of triggers in a DBMS is not considered particularly desirable by most DBMS researchers. Few general purpose DBMS include any notion resembling triggers.

These three illustrations have indicated widely differing points of view concerning data representation and services provided. This panel will focus on discussing these differences, why they are present and whether they are fundamental. Other fundamental issues where there is either commonality or a difference of opinion will also be identified.