

## A PANEL ON AI AND DATABASES

Raymond Reiter (Session Chairperson), Herve Gallaire, Jonathan J. King,  
John Mylopoulos, Bonnie L. Webber

### INTRODUCTION AND OVERVIEW

Raymond Reiter  
Department of Computer Science  
University of British Columbia  
Vancouver, B.C. V6T 1W5

During the past several years there has been considerable research within AI on various aspects of databases. The purpose of this panel is to provide non specialists with an overview of this research. To that end, I will use this introduction to survey briefly what I take to be the nature of the field, its significance for AI, and how it impacts database theory and practice. Each panelist in turn will then describe in greater detail the main concerns and issues surrounding his/her particular research area.

#### *The Nature of the Field*

The area of AI and databases divides naturally into the following three principal categories. (Names in parentheses refer to the position papers of the corresponding panelists).

#### *1. Database Interfaces (King, Webber)*

The concern here is with providing co-operative, friendly interfaces between the user and the information stored in the database. Users might range from database designers and administrators to end users with varying degrees of sophistication. As King and Webber point out, a wide variety of AI issues arise in this setting: query evaluation as a planning process, natural language input, query clarification, co-operative responses to requests etc. Many of these topics are linguistic in nature and hence overlap considerably with AI's longstanding concern with natural language. While most natural language research in databases has focussed on front-ends - there are even commercial systems available now - a growing body of literature is addressing the pragmatics of question asking and answering, as described by Webber.

#### *2. Conceptual Modelling (Mylopoulos)*

The objective of this research is to provide natural ways of representing, organizing and structuring knowledge about application domains so that a database designer will have a suitable set of tools (a conceptual model) for translating

his/her conceptualization of an application into a concrete, structured representation. Typical issues of concern are: How should one carve up a world into the right conceptual units in order to facilitate the design of a database? How do these conceptual units relate to one another? What inferences are sanctioned? In what formal language should such world views be represented? These, of course, are standard issues of knowledge representation in AI, so it is not surprising that most research in conceptual modelling informs, and is informed by, such notions as semantic networks, property inheritance, and knowledge representation languages like KL-ONE, KRL and FRL.

#### *3. Logic and Deduction (Gallaire)*

Research in this area draws on longstanding results in AI on logic, theorem proving and deductive question-answering. A database is viewed as a collection of logical formulae and queries are answered by proving the queries using the formulae of the database as premises. It is this perspective which underlies current work on logic programming and databases [Kowalski 81].

There is a complementary view of the role of logic in database theory [Reiter 83]. The argument is that logic provides a language for the abstract *specification* of conceptual models for databases, including the modelling of incomplete knowledge. Such a conceptual model may be *realized* in any number of ways by any number of data representations and operations on these data representations. The importance of this distinction is that, by respecting it, we are in a position to prove the correctness of a conceptual model's realization with respect to its logical specification. This is a latter day version of McCarthy and Hayes' distinction between the epistemological and heuristic adequacy of a representation.

#### *Significance of Database Research for AI*

The *practical* significance for AI of flexible intelligent databases is obvious. They are essential for any AI application requiring access to, and modification of large knowledge bases. But there is a less obvious, but equally important *scientific* and *methodological* significance for AI of research in databases. And that is that typical database domains, even the most mundane, provide a very rich source of problems for *foundational* studies in AI. Moreover, the worlds that databases are called upon to represent are often really quite simple, at least compared to, say, representations for story comprehension, medical diagnosis, or electronic circuit design. For example,

databases are often concerned with representing such simple worlds as inventories, office records, airline reservations etc. As a result of the simplicity of these worlds, the fundamental AI problems that arise do so in simple and perspicuous settings so that the essential features of these problems reveal themselves, unobscured by confusing details. Therefore, solutions to these problems are easier to come by. It is also clearer exactly what constitutes a solution. In her position paper, Webber makes a similar point, and shows how various concerns in the pragmatics of conversation arise with wonderful clarity in a database setting. By the same token, I think it was no accident that the major advances in computational linguistics due to Woods with respect to parsing and semantics emerged from the design of LUNAR, a natural language front-end to a database of facts about lunar rock samples.

Many other examples quickly come to mind of current research topics in AI which arise in a particularly perspicuous way in the database setting:

1. Non-monotonic reasoning and the closed world assumption for the treatment of negation in databases [Reiter 78].
2. Knowledge, belief, and incomplete information. The information in the database represents what it knows or believes about a world. It may have only partial information. The database should "know" that there are certain things it doesn't "know". These problems lead to metatheoretic solutions [Konolige 81] and modal logics [Levesque 82].
3. Truth maintenance problems [Doyle 79] arise in databases subject to updates and deletions, and in the maintenance of database integrity.
4. Knowledge representation for conceptual modelling [Mylopoulos 81].

Because of examples like these I believe that databases will come to be used as a fruitful blocks world for AI research, a blocks world with the nice additional property that it has enormous practical significance.

#### *Significance for Database Theory and Practice*

At the moment there is a wide but narrowing culture gap between the AI community of database researchers and the large community of researchers outside of AI who are involved with database theory and practice. The concerns of this latter group tend to be far more practical and "low level", involving as they do issues like file organization, concurrency, data models (as opposed to conceptual models - see Mylopoulos' position paper), efficient retrieval from very large databases, multiple user views, protection and security, etc. [Ullman 80]. Moreover, there is a deeply rooted reluctance to viewing a database as a representation of some world i.e. as a representation of knowledge. Mylopoulos characterizes this distinction between the non-AI and AI perspectives on databases:

"A (non-AI) database stores ... data. In the case of databases there is no commitment to an interpretation of the information stored in the database; for (AI) knowledge bases, there definitely is such a commitment."

This divergence of opinion about the nature of a database, together with AI's general lack of concern for efficiency considerations and large databases, have tended to separate the two communities. Fortunately, the situation is changing, mainly due to various proposals for semantic and conceptual data models as alternatives to the relational model [Brodie and Zilles 80, Tschritzis and Lochovsky 82]. Contributions from logic, deductive question-answering, and logic programming are also beginning to be acknowledged. The influence of AI on database theory and practice can only increase. The result will be user-friendly, moderately intelligent, efficient, well designed database systems. Their impact will rival that currently enjoyed by expert systems.

## LOGIC AND DATABASES

Herve Gallaire

Laboratoires de Marcoussis  
Centre de Recherches de la C.G.E.  
Route de Nozay  
91460 Marcoussis - France

A logic database (LDB) is a specification, using logic, of some data model, not necessarily relational. Deductive databases (DDB) are a proper subset of LDB's which provide a natural generalization to relational databases through a deductive mechanism. We argue that LDB's will play a major role in the future of database (DB) systems. LDB's are among the major contributions of AI to the DB field, the others being AI knowledge representation languages and natural language techniques.

DDB's contribution is far more reaching than was first thought after the pioneering work on question answering systems. DDB's offer the following features: deductive retrieval of implicit information, not stored in the DB; view expression, access and maintenance (a view is nothing but an external schema of the DB derived from its conceptual schema and specific to a class of users); mappings between various external models, possibly through a common relational DB, enhanced with deductive capabilities; a trivial embedding of a DB query language in a programming language, namely that of the deductive component; hints and possible paths to solutions to non-trivial problems such as integrity constraint expressions and verification, borrowing techniques found in the deductive component; semantic as well as syntactic query transformations geared either towards improving the man-machine interface or for optimizing data retrieval.

One may wonder whether these features will eventually belong to commercial products. Although there is no simple answer to this, it is felt that provided enough effort is put on all research aspects of the problem, much can be achieved. The basic reason for this controlled optimism is due to the fact that, for the first time, there is a continuous set of complementary tools resting on a common formal basis, i.e. logic: relational database machines with special operations for DB operators and retrieval, relational DB systems, and logic programming. Each of these levels already flowers with competing products which are even partially integrated; for instance relational DB machines on the market do provide a good interface

to existing DB systems. However a product such as a PROLOG system at the logic programming level, has not yet been truly marketed; hence it has not been integrated in a product with a relational DB system. One may foresee several such prototypes in the near future. The first phase of the Japanese 5th Generation project is providing one such prototype.

This development does not mean that such an integration is without problems. There are three types of problems to be solved: technical problems, performance problems, research problems. The first type has to do with problems the theory of which is well understood but which lack experiments for deciding important issues. Two examples of these issues are decisions about compiling or interpreting deductive queries, and about generating or deducing implicit data. The reasons why such experiments have not yet been conducted has to do with the maturity of DB experts; not too long ago they were (and some are) still disputing the fact that query languages have to be extended rather than embedded into programming languages. The second type of problem - performance - must also be overcome. In the same way as many DB experts still reject relational systems for poor performance, deductive DB's are supposed to be impractical. There is no reason why this should be true; rather one should not expect to get a whole set of new features without extra cost. Clearly what is to be decided is whether these new features are worth the overhead, as in the case of the relational database systems. Such a decision will obviously depend on much needed experiments, but one is faced with the fact that either these will be successful or a whole set of new markets of DB will be unfeasible. Finally, there are a number of research problems: will DDB provide enough capabilities for e.g. integrity constraint checking where it is known that the kind of logic needed goes, on practical grounds, beyond what is probably sufficient for purely deductive capabilities? Similarly, will deductive capabilities cater to technical issues which have no known answer in conventional DB's, such as correctly handling incomplete and unknown data, but which are all the more difficult in a deductive context? Will the theory provide solutions to a whole new set of problems brought up by introducing reasoning capabilities to DB systems, such as questions about non-monotonic reasoning, hypothesis and belief revisions, etc...?

LDB's share with DDB's all these capabilities. But they can help solve another class of problems which is important in DB circles: what after relational models? Semantic data models, which parallel knowledge representation languages in AI are advocated by many. LDB's offer a rigorous semantics for such data models and others. This capability is attained by using a logic programming language as a specification language while its execution mechanisms provide the deductive capabilities.

There is definitely a future for logic database systems: for the user, for the DB administrator. Logic programming is the key to it. Some references which are representative of the point of view of this position paper are [Gallaire and Minker 78, Gallaire et al. 83, Kowalski 81, Reiter 83].

## THE NEXT GENERATION OF INTELLIGENT DATABASE INTERFACES

Jonathan J. King  
Symantec  
Sunnyvale, California  
U.S.A.

The next generation of intelligent database interfaces will act as partners in the problem solving process. The right metaphor for them is the skilled research librarian or information retrieval specialist. Producing interfaces with the right capabilities requires basic research in the representation, use, acquisition, and coordination of basic facts, general rules, and metaknowledge within the same information system.

Skilled information retrieval specialists possess a host of special skills. They have a general knowledge of particular fields, at the level of classifying and relating their subfields and knowing the main contributors. They have precise knowledge about information resources - what is in them, how they are organized, what research aids are available. They have strategies of searching for answers. And they can evaluate the quality of the information they find based on its source and their general knowledge of the field.

When presented with a problem, the information specialist determines whether his resources are likely to provide an answer. It may be necessary to refine the problem statement so that useful information can be provided. The specialist devises a search plan, then forms questions precisely tailored to the requirements of his information resources. Intermediate results may cause him to revise the search plan. When the search is complete, the specialist may qualify the answer to account for the accuracy or currency of his sources. He can also answer questions about the information resources themselves so that the user can get a better idea of what kinds of problems the specialist can handle. An intelligent database interface with these capabilities is more than an interface; it is a broker and manager of information. Current artificial intelligence research bears on each of these tasks.

### *Formulating Questions*

Natural language research aims to increase linguistic coverage, and there has obviously been much progress along this line. Still, we are far from guaranteeing that all NL input is properly understood. There are two interim solutions. The first is to clarify the question by interrogating the user. The RENDEZVOUS system [Codd 78] does this. RENDEZVOUS' dialogues can be cumbersome, largely because clarification is carried out in terms of underlying database concepts; still, RENDEZVOUS achieves zero error rate. (The IFAP system [McLeod 78] is another clarification interface, but for a formal query language; its clarification dialogues are also cumbersome and based on database concepts). The second interim solution is to constrain NL input so the question is sure to be understood. Texas Instruments is developing a system in which the user builds an English sentence by selecting from menus of acceptable continuations [Tennant 83]. Choices are expressed in terms of sentence words, not underlying database concepts. Selection using a pointing device also reduces typing to a minimum. Users can't freely compose a sentence, but can usually express what they want.

### *Tailoring Questions to Information Resources*

Tailoring the question to the available information resources means producing a better question than the user can by himself; it is an "added value" function of interfaces. The interface must take into account general rules about the database domain. Hafner [Hafner 79] uses a semantic network of concepts from an area of the law to add useful qualifications to queries against a full-text retrieval system of legal documents. The IR-NLI system [Guida 83] does this also for a different full-text system. The QUIST system [King 81] changes a query into an equivalent one that can be processed more efficiently; it uses domain knowledge to transform the query, and knowledge of database structure and access methods to tell whether a transformation will be beneficial. Several systems (e.g. NLP/DBAP [Ginsparg 83], TEAM [Grosz 83]) are exploring ways to use inference to fill in necessary information the user omits.

### *Devising Retrieval Strategies*

Figuring out how to answer a question is essentially a planning process. Furukawa [Furukawa 79] posed retrieval from deductive databases as a problem of planning a search strategy. The IR-NLI system mentioned above is consciously modelled on the role of the information retrieval specialist, in the realm of full-text retrieval using key word descriptors. The aim is to produce an expert system whose expertise is a model of retrieval strategies [Bates 79]. After helping to formulate the query, IR-NLI is designed to revise it in light of intermediate results and produce a final search strategy that satisfies various criteria imposed by the user. Ideally, an intelligent database interface should produce an "opportunistic plan" [Hayes-Roth 78] that augments the basic retrieval plan by getting other information of use to the user, or, if it determines the question can't be answered precisely, answers a different question that may nevertheless be helpful. Current work in planning utterances [Appelt 82] and communicating between cooperative agents [Konolige 80] is relevant to this.

### *Explaining and Qualifying Answers*

Some intelligent database interfaces have been designed to correct faulty presuppositions when a question fails ([Janas 79], [Kaplan 79]). This does not require a knowledge of general domain rules, but just an analysis of the subparts of the query against ground facts in the database. Current research [Webber 82] concerns when and how to offer explanations with answers in general, not just when the answer is empty, and also when answers are produced by a combination of ground facts and general domain rules.

### *Responding to General Questions*

The inclusion of general domain rules and metaknowledge (knowledge about the general contents and structure of the database) makes it possible to respond to general questions. McKeown [McKeown 80] uses an augmented database schema to support natural language questions about the structure of

the database. Konolige has proposed [Konolige 81] representing the mapping between real-world concepts and database contents formally as a metalanguage and a database language which is its intended model. Database queries are then formally derived from metalanguage questions stated in terms of real-world concepts. It is also possible to derive whether a particular kind of database query can be asked, thus identifying areas where the database is incomplete. Work on FOL [Weyhrauch 80] and SPHERE [Filman 83] is along these lines as well. To respond to general questions, it is also essential to extend the linguistic analysis so that it is not assumed that a question translates directly into a database query. Work on KLAUS [Hendrix 80] is taking up this question.

### *Integrating Multiple Information Resources*

Several systems are now under development to permit a single interface to reference several information resources and to control the operation of related programs, including expert systems [Weiss 82]. Portability of natural language interfaces is being extended to include nonformatted databases [Thompson 83]. The KLAUS project is also explicitly concerned with tying together multiple information resources with mixed types of data.

These capabilities, combined with advances in natural language understanding, will constitute a large step forward for the next generation of intelligent database interfaces. It is only fair to add that we are a long way from attaining them, because we must confront very hard problems in the representation and use of knowledge.

## KNOWLEDGE REPRESENTATION AND DATABASES

John Mylopoulos  
Department of Computer Science  
University of Toronto  
Toronto, Ontario

I'd like to draw a sketch of what "AI and Databases" means to me and then concentrate on some of the issues we have been addressing with our work at the University of Toronto.

### *Knowledge bases vs. Databases*

A knowledge base contains knowledge about something represented in a suitable notation. A database stores (large amounts of) data. In the case of databases, there is no commitment to an interpretation of the information stored in the database; for knowledge bases, on the other hand, there definitely is such a commitment. Thus a relational database stores tuples and one can ask "Is there a tuple  $x$  such that  $Q(x)$ ?" Moreover, the answer to this question can only be "yes" or "no". With a knowledge base, however, we are storing information about people, accounts, etc. and a query takes the form "Is there an employee  $x$  such that  $P(x)$ ?" Now the

answers can be "yes", "no" or "I don't know" (and there are other possibilities too). The distinction is actually quite clearly reflected in the terms that have been used for the notations we have developed in order to construct, manipulate and search databases and knowledge bases: a Data Model (such as the relational) offers a set of data structures and associated operations for the construction and manipulation of a database. A Knowledge Representation scheme or Conceptual Model offers tools for the representation and organization of knowledge. The first commits its user to a particular way of visualizing what is stored in a machine. The second to a particular way of viewing the world. The so-called "semantic data models" seem to offer the promise of a dual commitment. I consider this approach to the introduction of semantics wrong. Rather, I prefer the strategy of having a knowledge base at one level (the conceptual level) and a database at a different level (the logical level).

The distinction I am drawing here is by no means absolute. For example, proposals for the treatment of "null values" (e.g. [Vassiliou 81]), extensions of the Relational Model (e.g. [Codd 79]), or revisions (e.g. [Reiter 83]) do lead to versions of the Relational Model that make some semantic commitments.

#### *Technology Transfer: From AI to Databases*

There are three main ways I can see AI contributing to Database technology:

1. Increasing the functionality of a DBMS - with natural language front ends, deductive question-answering, consistency checking, planning and the like; clearly this type of technology transfer is of importance to the DBMS user.
2. Using expert system technology to help the database builder - by offering expert systems which are fed information on the characteristics of a database and generate automatically or semi-automatically a database; here the beneficiary is the database builder who gets an apprentice (in the sense used by [Rich 81]).
3. Using Knowledge Representation work to construct Conceptual (rather than Data) Models - now the designer is helped in terms of notational tools so that he can first describe a knowledge base and then proceed with the construction of one or more databases that implement the knowledge base.

#### *Technology Transfer: The Other Way*

So "our" knowledge bases are much better suited for talking about the world while "their" databases are clearly more implementable. Is there any reason for worrying about implementability? I think there is, and it's a pragmatic one. We have a technology that allows us to build knowledge bases whose size is, say, order 1K. We need a technology that will allow us to build knowledge bases whose size is order 100K. One (but not the only) obstacle in achieving this goal is performance. Database research has developed tools to deal precisely with this issue, and it seems natural that we should turn to them for inspiration. I should add here that "turning to them" doesn't necessarily mean adapting, say, relational or

DBTG technology. It may well happen that these technologies are obsolete, due to advances in hardware and communications research. But the research area that created these technologies seems to have an important role to play in dealing with performance issues for any new technology that tries to replace them.

Apart from performance considerations, there are other features of database systems such as error recovery and concurrency which are essential for the construction of robust knowledge base systems.

#### *Knowledge Representation and Databases*

It follows from this discussion that Knowledge Representation has a dual role to play in Database applications:

- (a) a direct role in providing ideas for the design of conceptual models, and
- (b) an indirect role in providing tools and methodologies for the construction of expert systems that aid the designer and/or user of a database.

Much of our work at the University of Toronto has concentrated on the first role and overviews of our results appear in [Brachman and Smith 80], [Mylopoulos 81], [King 83].

#### *Some Research Issues*

AI, or more specifically Knowledge Representation, applications to Databases has raised fundamental research problems which must be tackled before there can be progress. Let me mention three such problems:

##### *1. Representation of Social Knowledge*

Much of the knowledge that is relevant to a large class of databases is social in the sense that it involves rules, regulations and procedures for the operation of a social structure, e.g. a hospital, an office, a corporation. Such knowledge is fundamentally different from either (mathematically) logical knowledge or common sense knowledge.

Consider, for example, a fact such as "Incoming patients must present their Hospital Insurance Number" which may be associated with a hospital admission system. Clearly, we do not want to treat its enforcement in the same way we would treat, say, "Every natural number has a successor" or the way we would treat the enforcement of, say, "Unless you know otherwise, an elephant is grey".

Designing Knowledge Representation schemes which enforce social regulations such as the one above but in a *flexible* way appears to be a difficult and important research question.

##### *2. Relativism of Representation*

The view that for any problem domain we can construct a knowledge base that presents a global (and consistent) viewpoint is wrong. Consider a large organization for which we are trying to build a knowledge base that describes its operations. It is possible, indeed likely, that each department

has its own view of what it and other departments are doing. Moreover, this view may be contradictory to the views of other departments. Our knowledge representation tools must allow us to construct a knowledge base consisting of a collection of interrelated and possibly contradictory views.

### 3. Compilation of Knowledge Bases

Even small database applications require hundreds of thousands of facts to be stored, accessed and manipulated. To build knowledge bases of comparable size, we must deal with performance issues, such as the compilation of a knowledge base into a database along with applications programs (e.g. [Nixon 83]), incremental compilation and efficient consistency checking, [Smith 81].

## PRAGMATICS AND DATABASE QUESTION ANSWERING

Bonnie Lynn Webber

Department of Computer & Information Science  
University of Pennsylvania  
Philadelphia, PA 19104  
U.S.A.

Artificial Intelligence research is fortunate to have database systems around: otherwise, it would have had to invent them (and probably not do as good a job). This is not such an outrageous claim: database systems provide an ideal environment for investigating purely AI-type questions (formal reasoning, common sense reasoning, Natural Language (NL) use, including question answering and answer justification and explanation, NL understanding, etc.), independent of worldly domain. On the other hand, AI research in the context of database systems has already benefitted those systems in several ways, one of which is by making possible Natural Language *input*. This makes such systems more accessible to naive or infrequent users by allowing them to state their requests as they would everyday.

AI can now benefit database systems in another way - by making possible system *output* that follows everyday conversational principles and practices. In this way, a user's *normal expectations* about responses to his/her utterances and *normal strategies* for interpreting those responses will not fail him/her, when those responses come from a machine.

Since this is only a brief position paper, I will try to make my point by means of a few short examples. Note that *Natural Language* is not the primary issue: misleading or unexpected responses, even in the King's English, would be just as bad. What is at issue is standard conversational principles and practices. My point is that AI can provide the means to enable database systems (and knowledge-based systems, in general) to follow such principles and practices and hence participate in more effective and satisfying interactions with their users.

(In each of these examples, the first system response is of the kind one might see today: the second (*equally correct*) response is as it could be, were normal conversational practices and principles being followed.)

### Example 1

U: Which department managers earn over \$40K per year?  
S1: Abel, Baker, Charley, Dogg  
S2: All of them.

By enumerating managers who *do* earn over \$40K per year, the first response implicates that there are managers who *don't* earn that much. In linguistic pragmatics, this is called a "scalar implicature": it follows from the assumption that a cooperative respondent will say as much as s/he can. In this case, s/he would only enumerate positive instances if s/he *couldn't* say the more inclusive "All of them". Thus S1 falsely misleads a user, who assumes the system would say S2 if it were true. What AI can do here is to provide tools for (1) computing when such implicatures are in force (a peculiar thing about implicatures being their potential cancellability [Gazdar 79]) and (2) reshaping responses to accord with them.

### Example 2

U: Where does the Ubu Roi stop on her current tour?  
S1: Bombay, Cape Town, Honolulu, Lisbon, Los Angeles, Nantes, New York, Sidney.  
S2: New York, Nantes, Lisbon, Cape Town, Bombay, Sidney, Honolulu, Los Angeles.

Both responses S1 and S2 convey that the cities noted constitute the stops on the Ubu Roi's current tour. However, an enumerated response to a question such as U's implicates that the enumeration corresponds to an expected ordering - in this case, temporal order - first stop, second stop, etc. Both S1 and S2 will be interpreted in this way (S1, incorrectly so). What AI can do is specify such expectations, so that responses like S2 are produced if possible (i.e., if the system can determine the order of stops) and if not, an explicit denial is made that the user's expectations are not being met - e.g.,

S1': The stops are (not necessarily in this order): Bombay, Cape Town, Honolulu, Lisbon, Los Angeles, Nantes, New York, Sidney.

The two examples above show systems giving unintentionally misleading responses by ignoring their standard implicatures. There are also cases where a system's responses are misleading because they do not disabuse the user of misconceptions s/he seems to have about the domain or its encoding in the databases. This may lead to the user's drawing false conclusions from the system's response. This is what I meant earlier by the user getting an unexpected response: in cooperative human-human interactions, the other party normally takes responsibility for helping wherever possible, ideally in a way that does not cause loss of face.

The most common illustration of this type of misleading behaviour comes from Kaplan's work in this area [Kaplan 81].

### Example 3

U: Which Economics students got an F in CIS531 in Fall '82?  
S: None.

Someone asking such a question must believe, *inter alia*, that there were Economics students and that CIS531 was given in

Fall '82. The system's simple response "None" may lead such a person to conclude that all Economics students got grades other than F - i.e., that they all passed. If "None" in truth follows from the fact that there were no such students or the course wasn't given, then the system has misled the user.

There are many types of misconceptions a user may have - that a relation can hold between certain entities when it can't (e.g., "Which chickens teach CIS531?" presumes that chickens can teach graduate courses), that an object has a particular attribute when it doesn't (e.g., "What's the maximum age for opening a Money Market Account?"), etc. [Mays 80, Webber 83]. Not all such misconceptions can be detected, not all can be easily corrected, and not all make a difference to the system's ability to convey its information responsibly [Joshi 82]. What I believe AI can contribute here (and what several of us at Penn are working to do) is to improve a system's ability to detect and correct misconceptions detrimental to the successful transfer of information from the database to the user, thereby increasing the system's potential for doing well by its users.

One final note: the examples above presume no more of a database system than standard retrieval capabilities. However as database systems begin to take advantage of AI's deductive capabilities in order to provide answers to a wider range of questions, the issue of adhering to standard conversational principles and practices becomes even more important. Though I haven't the space here to elaborate upon this, the panel I hope will provide an opportunity to do so.

## REFERENCES

- Appelt, D., "Planning Natural-Language Utterances", AAAI-82, 1982, pp. 59-62.
- Bates, M., "Information Search Tactics", J. of the American Society for Information Science, 30:205-214, 1979.
- Brachman, R. and Smith, B., *Special Issue on Knowledge Representation*, SIGART Newsletter, February 1980.
- Brodie, M.L. and Zilles, S.N. eds., *Proc. Workshop on Data Abstraction, Databases and Conceptual Modelling*, Pingree Park, Colorado, June 23-26, 1980.
- Codd, E.F., "Extending the Database Relational Model to Capture More Meaning", Transactions on Database Systems, December 1979.
- Doyle, J., "A Truth Maintenance System", Artificial Intelligence 12, 3, Nov. 1979.
- Furukawa, K., "Relational Strategies for Processing Universally Quantified Queries to Large Data Bases", IJCAI-79, 1979, pp. 294-299.
- Gallaire, H. and Minker, J. (eds.), *Logic and Data Bases*, Plenum Press, NY, 1978.
- Gallaire, H., Minker, J. and Nicolas, J.M., "Logic and databases - an overview and survey", submitted to Computing Surveys, 1983.
- Gazdar, G., "A solution to the projection problem". In Oh, C.-K. and D. Dinneen, eds. *Syntax and Semantics*. XI. NY: Academic Press, 1979, pp. 57-90.
- Ginsparg, J., "A Robust Portable Natural Language Data Base Interface", ACL Conference on Applied Natural Language Processing, February 1983, pp. 25-30.
- Grosz, B., "TEAM: A Transportable Natural Language Data Base Interface", ACL Conference on Applied Natural Language Processing, February 1983, pp. 39-45.
- Guida, G. and Tasso, C., "IR-NLI: An Expert Natural Language Interface to Online Data Bases", ACL Conference on Applied Natural Language Processing, February 1983, pp. 31-38.
- Hafner, C., "An Information Retrieval System Based on a Computer Model of Legal Knowledge", Ph.D. thesis, University of Michigan, 1979.
- Hayes-Roth, B. and Hayes-Roth, F., "Cognitive Processes in Planning", Tech. Report R-2366-ONR, RAND Corporation, December 1978.
- Hendrix, G., "KLAUS: A System for Managing Information and Computational Resources", Tech. Note 230, SRI Artificial Intelligence Center, October 1980.
- Janas, J., "How Not to Say 'NIL' - Improving Answers to Failing Questions in Data Base Systems", IJCAI-79, 1979, pp. 429-434.
- Joshi, A.K., "Mutual Beliefs in Question Answering Systems", in N. Smith (ed.), *Mutual Belief*. New York: Academic Press, 1982.
- Kaplan, S.J., "Cooperative Responses from a Portable Natural Language Data Base Query System", Ph.D. thesis, University of Pennsylvania, 1979.
- Kaplan, S.J., "Appropriate Responses to Inappropriate Questions". In A. Joshi, B. Webber & I. Sag (eds.), *Elements of Discourse Understanding*. Cambridge, England: Cambridge University Press, 1981, pp. 127-144.
- King, J., "QUIST: A System for Semantic Query Optimization in Relational Databases", VLDB-7, September 1981, pp. 510-517.
- King, J. (ed.), *Special Issue on AI and Databases*, SIGART Newsletter, 1983 (forthcoming).
- Konolige, K. and Nilsson, N., "Multiple-Agent Planning Systems", AAAI-80, pp. 138-142.
- Konolige, K., "A Metalanguage Representation of Relational Databases for Deductive Question-Answering Systems", IJCAI-81, 1981, pp. 496-503.
- Kowalski, R., "Logic as a database language", Proc. Advanced Seminar on TIDB, CETRARO, September 1981.
- Levesque, H.J., "A Formal Treatment of Incomplete

Knowledge Bases", FLAIR Tech. Report 3, Fairchild Lab. for AI Research, Palo Alto, CA., February 1982.

Mays, E., "Failures in natural language systems: application to data base query systems", Proceedings of the first meeting of the American Association for Artificial Intelligence (AAAI), Stanford, California, August 1980.

McKeown, K., "Generating Relevant Explanations: Natural Language Responses to Questions about Database Structure", AAAI-80, 1980, pp. 306-309.

McLeod, D., "A Semantic Data Base Model and its Associated Structured User Interface", Ph.D. thesis, MIT, August 1978.

Mylopoulos, J., "A Perspective for Research on Conceptual Modelling", in Brodie, M. and Zilles, S. (Eds.) *Data Abstraction, Databases and Conceptual Modelling*, SIGART Newsletter, January 1981.

Nixon, B., *A Taxis Compiler*, M.Sc. Thesis, Department of Computer Science, University of Toronto, 1983.

Reiter, R., "On Closed World Databases", in [Gallaire and Minker 78], pp. 55-76.

Reiter, R., "Towards a Logical Reconstruction of Relational Database Theory", in Brodie, M., Mylopoulos, J. and Schmidt, J. (Eds.) *On Conceptual Modelling*, Springer-Verlag, 1983.

Rich, C., "A Formal Representation for Plans in the Programmer's Apprentice", Proceedings IJCAI-81, August 1981.

Smith, J.M., "Consistency of Models" in Brodie, M. and Zilles, S., 1981.

Thompson, B. and Thompson, F., "Introducing ASK, A Simple Knowledgeable System", ACL Conference on Applied Natural Language Processing, February 1983, pp. 17-24.

Tsichritzis, D.C. and Lochovsky, F.H., *Data Models*. Prentice-Hall, Englewood Cliffs, NJ, 1980.

Ullman, J.D., *Principles of Database Systems*. Computer Science Press, Potomac, Maryland, 1980.

Vassiliou, Y., *Formal Treatment of Imperfect Information in Database Management*, Ph.D. Thesis, Department of Computer Science, University of Toronto, 1981.

Webber, B. and Joshi, A., "Taking the Initiative in Natural Language Data Base Interactions: Justifying Why". COLING-82, 1982, pp. 413-418.

Webber, B., "Varieties of User Misconceptions: Detection and Correction". Submitted to IJCAI-8, Karlsruhe, Germany, August 1983.

Weiss, S. et al. "Building Expert Systems for Controlling Complex Programs", AAAI-82, 1982, pp. 322-326.

Weyhrauch, R., "Prolegomena to a Theory of Mechanized Formal Reasoning", *Artificial Intelligence* 13(1-2): 133-170, 1980.