Using Conceptual Graphs to Represent Database Inference Security Analysis

Harry S. Delugach and Thomas H. Hinke

Computer Science Department, University of Alabama in Huntsville, Huntsville, U.S.A.

This paper describes an approach to database inference analysis based on conceptual graphs. The database inference problem is briefly described. Previous approaches are summarized, followed by a presentation of our inference model, called AERIE. The notions of an inference target class and an inference method class are introduced with examples given. Conceptual graphs are introduced as our means of representing database inference knowledge, as a first step toward analyzing and detecting database inference problems. The classification of inference target classes and the use of conceptual graphs for database inference detection are two important contributions of this paper. Four examples are used to illustrate the approach. We discuss some interesting issues raised by this work, and offer conclusions and our plans for future research.

Keywords: Database inference, Conceptual graphs, Inference models, Inference classification, Inference target, Inference representation.

Introduction

This paper describes an approach to database inference analysis based on the conceptual graphs knowledge representation. We will describe the database inference problem, introduce conceptual graphs as a means of representing knowledge relevent to database inference and show some examples that point to future directions in this area of study.

The database inference problem can be characterized as follows:

> Data that is classified at access level L_1 or below can be used to infer data classified at access level L_2 where the classification $L_2 > L_1$.

The problem of database inference can be illustrated with examples. Suppose we wish to keep secret the fact that XYZ Research Corporation is working on the AERIE missile project. While we may be able to prevent an adversary from knowing this particular fact directly, Laura Smith's name may appear on an unclassified list of attendees for a meeting about the classified AERIE missile project. If it is known that Laura Smith is an engineer for XYZ Research Corporation, then it can be inferred that XYZ Research may be working on the AERIE missile project.

As another example, we may wish to keep secret the fact that F–21As are based at Tinian Island. Suppose that an unclassified shipping manifest shows that part number 125–678–98–54 is being shipped to Tinian Island and an aircraft maintenance manual shows part number 125–678–98–54 is used only on the F–21A jet fighter. One can then make the inference that F–21A's are based at Tinian. A non-military example is the ability to infer that a patient has a particular disease based on knowledge of the treatment that is being applied.

With the publication of the Trusted Database Interpretation of DoD 5200.28–STD (National Computer Security Center, 1985)(National Computer Security Center, 1991), a major step has been taken to provide protection for large amounts of data stored in databases. While the ability to provide discretionary and mandatory access control (at least at the higher evaluation levels) is an important step forward in providing protection for the database, it is not sufficient. An organization could properly use the protection facilities provided by a multilevel secure database management system and still be at risk, due to the ability of an adversary to access unclassified or less classified data and be able to deduce more highly classified data using inference techniques.

This problem will become more acute as more data is placed under the control of databases and these database are connected to networks (e.g, the proposed National Information Infrastructure in the U.S), such that they can be accessed by many users located all over the world. In addition, as network accessibility becomes the norm for database access, an adversary could potentially have access to considerably more data than has been available to a single user in the past. This greatly multiplies the potential to infer unauthorized data, since there is a greater pool of unclassified data available with which to perform the inference.

What makes the inference problem especially difficult to overcome is the deep knowledge that could be applied by an adversary to perform the inference attack. An analysis approach that depended solely on the contents of a database, or even a collection of databases, will fall short of capturing the deep knowledge that a determined adversary might have. A useful inference detection approach must be able to model general knowledge available from a variety of sources, not just information contained in databases.

This paper describes our approach to modeling the database inference problem using conceptual graphs. The task of representing such problems is the first step toward detecting and preventing them from occurring in actual databases. The paper is organized as follows. The next section summarizes previous approaches to the database inference problem. The following section describes our approach, which we have dubbed the AERIE approach. We then show two examples illustrating how conceptual graphs are used to represent the knowledge used in performing a database inference. At the end of the paper, we present some interesting issues raised by this approach and give our conclusions and future research plans.

Approaches To Database Inference Analysis

Previous inference has been characterized in (Hinke, 1990b) into the following categories:

1. "efforts to discover fundamental laws that determine whether the potential for undesirable inferences exists within a given database,
2. efforts to discover automatically inference rules from fundamental relationships among data that pertain to a domain,
3. efforts to automate (via expert systems) the process of inferring sensitive data within a specific domain." (Hinke, 1990b).

   This list can be extended with a fourth category to reflect some recent work at SRI International (Garvey & Lunt, 1991).

4. Efforts to jam the inference channel with "noise" provided by plausible cover stories.

Research into the first category includes work on statistical databases (Cox, 1988; Denning, 1982; Matloff, 1988) and work exploring the inference issues with respect to functional and multivalued dependencies within relational databases (Su & Ozsoyoglu, 1990). Also included within the first category is the work of Morgenstern (Morgenstern, 1987, 1988) to propose a theoretical foundation for inference, using such concepts as a sphere of influence that is the transitive closure of all that can be inferred by a particular fact.

Research in the second category is represented by the work of Hinke (Hinke, 1988, 1990a) which sought to discover inference channels that would permit classified relationships between two entities to be discovered by finding second paths, consisting of relationships between other entities that could be used to make the sensitive linkage. One of the important results of this work was that the specific rules required to find the classified relationship did not have to be stated explicitly. While they could be stated upon viewing the second path discovered, all that was required to perform the inference discovery was to find this second path by traversing relationships between entities until the two target entities were joined, using a path that was less classified than that associated with the direct relationship between the two entities. This line of research has been continued by Binns (Binns, 1992b) and SRI Internation (Lunt, 1989) (Qian, Stickel, Karp, Lunt, & Garvey, 1993) through the design and implementation of path finding programs. Also included in this inference category is the work of Thuraisingham (Thuraisingham, 1991), which uses conceptual structures to represent multilevel applications, thus providing the basis for analyzing these multilevel structures for the likelihood that an adversary can draw unauthorized inferences. Our paper, while also using conceptual structures, extends earlier work by including the use of conceptual structures within a general model of inference.

The third category seeks to capture rules that could be used by an expert to detect inference problems within existing databases. Research in this area includes work by Ford Aerospace on their inference controller (Buczkowski, 1990).

The fourth category of inference work is represented by the current research on inference underway at SRI International (Garvey & Lunt, 1991). While the work addresses the general inference problem, one of the areas that they considered in their paper was the use of cover stories to add noise to the inference path. This is especially useful when the inference path may be composed of widely known facts that cannot be easily classified.

This paper reports on work in the second and third categories, with the use of a new inference model being developed at the University of Alabama in Huntsville. This model, called the AERIE Model, seeks to permit the discovery of inferences using inference targets (sensitive things that are to be protected from unauthorized disclosure through inference). It also utilizes a knowledge base, using conceptual graphs, that will permit an automated Inference Analysis Tool to reason about the existence of information using not only data within the database, but also other knowledge that would be assumed to be known by an adversary.

AERIE Model of Inference

Our inference model is called AERIE, which stands for Activities, Entities, and Relationships' Inference Effects. For our purposes, *entities* (symbolized by "E") are things that exist in the world and *activities* (symbolized by "A") are actions that take place in the world. *Relationships* can be categorized as:

- A relationship between two entities, symbolized by (E,E).
- A relationship between an entity and an activity, symbolized by (E,A) or (A,E).
- A relationship between two activities, symbolized by (A,A).
- A relationship between two or more relationships, symbolized by $((W_1, W_2), (W_3, W_4))$, where each W_i is either "E" or "A".

The goal of an adversary is to learn secret information where it not readily available. We refer to acquiring this knowledge as "materializing" the sensitive information.

Information to be kept secret will be referred to as an *inference target*. Inference targets can be categorized into inference target classes based on the above distinctions. The AERIE approach uses these target classes to classify sets of inference methods according to the target they can materialize. This section describes the target classes and their relationship to each other.

Inference Target Classes

Inference target classes identify the types of information that could be the target of an inference attack. The following seven classes have been identified:

Class 1. E: The materialization of an entity.

Class 2. A: The materialization of an activity.

Class 3. (E,E): The materialization of a sensitive relationship between two or more materialized entities.

Class 4. (A,A): the materialization of a sensitive relationship between two or more materialized activities.

Class 5. (E,A) or (A,E): The materialization of a sensitive relationship between one or more materialized entities and one or more materialized activities.

Class 6. $((W_1, W_2), (W_3, W_4))$: The materialization of a sensitive relationship between sensitive relationships.

Class 7. $[C_1, C_2, \ldots, C_{n-1}] \Rightarrow C_n$ (where each C_i is a materialization in one of the classes 1 through 6): the materialization of a sensitive rule from existing classes.

Knowledge in each inference target class can be inferred by one or more techniques. In categorizing these techniques, we refer to them as *inference method classes*. We will refer to the sequence of steps used in making an inference as an *inference path*. For each inference target class, we want to identify a set of methods that can be used in the derivation of an inference path. The purpose of categorizing inference methods is to reduce the number of possible inference paths; i.e., given a target in a particular class, we can identify a general pattern of steps that can be used to infer the target.

Our characterization of inference method classes leaves open the opportunity to include additional method classes as they are identified. The AERIE approach does not depend upon a closed set of classes; it merely provides a framework under which new classes (and their inference methods) may be identified and categorized. One potentially useful addition would be based on probabilistic reasoning; e.g., an entity or activity could be materialized (at least to some degree of confidence) based on probabilities embedded in the knowledge base. Another potentially useful addition would be to use statistical inference techniques, assuming that there were enough instances in the available databases. The statistics could be probed to isolate a particular; identifiable member of the group covered by them. The end result is the materialization of a particular entity; hence, statistical inference methods are included under the first inference target class, since they can be used to materialize a particular entity.

Examples for Each Target Class

Having provided an overview of the AERIE Model, its various components, the inference target classes and their related method classes, we will now describe an example of each inference target class, along with the methods that are used in performing the inference.

Class 1. An example of inferring a target entity can be found within a logistics database. If a site orders a part that is unique to a particular type of equipment, such as a certain radar unit, then an adversary with access to this database could infer that the site has this particular type of radar unit. This inference is made using the following method: $\{E_1, (E_1, E_2)\} \Rightarrow E_2$, where E_1 is some unique part and the (E_1, E_2) relationship is the part-of relationship showing the parts contained in each piece of equipment. The set $\{E_2\}$ represents all of those whole goods in which the part is used. If the part is used only on a single whole good, then the cardinality of the set is one and we have an inference that results in the unique identification of a piece of equipment.

Class 2. An example of inferring a target activity is the ability to infer that a construction project is occurring based on the class of equipment that is being ordered. Thus, if an equipment requisition includes equipment that can be used collectively for digging, pushing (e.g., pushing dirt) and carrying, this could indicate the existence of a construction project. On the other hand, if the equipment being ordered is used for mowing, plowing, or harvesting (e.g., wheat), then this would not be an indication of a construction project. Performing this inference requires that a construction project be modeled with a definition that characterizes it as including the activities of digging AND pushing AND carrying. Then the various equipment parts must be characterized with Class 5 information that has the form (E, A), relating the parts to the activities that they support. For example, a blade supports the activities of pushing, a bucket supports the activities of digging and a loader supports the activities of carrying and pushing. Finally, pieces of equipment must be characterized in terms of their component parts, with Class 3 (E, E) relationships that associate a part with an end-item. Thus, a backhoe contains a bucket and a loader, while a tractor contains a blade.

The inference of a construction project activity can then be made by determining whether there exists a requisition which contains equipment that can be used collectively for digging AND pushing AND carrying. To perform this inference analysis, the parts in an order can be used along with the (E,E) and (E,A) relationships to perform an inference using the following methods:

1. Definition, to determine the definition of a construction project, which would be listed as an instance of the inference target class 2;
2. Method $\{A_1,(A_1,E_1)\} \Rightarrow E_1$ to determine the equipment components that can be used for digging, pushing and carrying;
3. Method $\{E_1,(E_1,E_2)\} \Rightarrow E_2$ to infer pieces of equipment and component parts that are associated with the components that can be used for digging AND pushing AND carrying.

These various E_2's will be checked against the parts requisition and used with the previous rule to generate more component parts and end-items that are related to the entities that perform the digging, pushing and carrying activities associated with a construction project. For this inference, we have used two methods and a definition.

Class 3. A Class 3 inference is the determination of a sensitive relationship between two entities. For example, assume that some organization was attempting to keep secret its area of operation. This would be an association between the entity *organization* and entity *location*. Now assume that this organization ordered blades. If these blades were for bulldozers, then one could not make much of an inference. However, if these were snow blades, then one could make the inference that the organization operates in snow country. While this is admittedly not a precise location, it does narrow down the possible area of operation.

Class 4. A Class 4 inference is the determination of a sensitive relationship between two or more activities. An example was in the relationship in 1941 between Japanese trade negotiations with the U.S. and the movement of the Japanese fleet toward Pearl Harbor. Negotiators were given a strict deadline for their activity, but did not know the relationship between their negotiations and the attack.

Class 5. A Class 5 inference is the determination of a sensitive relationship between one or more entities and one or more activities. For example, if an intelligence activity required that a certain fixture be placed in the space shuttle's payload bay to support a particular type of sensor, then the association of this fixture with the intelligence gathering activity would represent a sensitive relationship that should be protected.

Class 6. A Class 6 inference is the determination of a sensitive relationship between sensitive relationships. An example of this class is a student grade inference in an academic setting. Assume that grades are posted by student numbers, to preserve the confidentiality of the grade that a particular student received. This represents a class 1 relationship between the entity student and the entity grade (e.g., (E,E)). However, if these posted grades were sorted by the last name of the student, this would represent a sensitive relationship, called "Sorted-by-name" between the (Student_number, Grade) relationship which is public knowledge and the (Student_name, Grade) relationship which is sensitive. If this Sorted_grade relationship can be inferred, then the very sensitive (Student_name, Grade) relationship can also be inferred.

Class 7. A Class 7 inference represents the inference of a sensitive rule. An example of a sensitive rule might be one used by a credit card company that says never to reject a charges for a restaurant meal, under the reasoning that since the food has already been consumed, a customer may be subjected to embarrassment by a rejected approval. In general, this class of inference incorporates any inference that results in a rule, rather than an entity, attribute or relationship. This class of inference target represents a different quality target than the previous ones that are shown; it is included since it represents another type of information about which one could launch an inference attack.

We note that Class 1 (inferring an entity) and Class 2 (inferring an activity) bear some resemblance to each other, since we are in fact inferring the existence of some particular thing irrespective of its relationship to others. Using conceptual graphs, in fact, the two classes are handled in a similar manner, since Class 1 involves deriving a single concept [ENTITY: *] and Class 2 involves deriving a single concept [ACTIVITY: *]. We separate the two because we may find in the future that some specific inference method classes for the two differ.

A similar line of reasoning shows that Classes 3, 4 and 5 are alike as well, since they involve deriving a single relationship between two entities or activities. We are interested in whether the inference methods differ for these three classes as well.

Inference Detection Using Conceptual Graphs

The knowledge representation chosen for this research is that of conceptual graphs (Sowa, 1984; Nagle, Nagle, Gerholz, & Eklund, 1992). Conceptual graphs are a graphically oriented notation based on first-order logic as denoted by Charles Peirce's existential graphs from the late 1800's. An extension of semantic networks (Lehmann, 1992), they provide a powerful, extensible means of capturing real-world knowledge, such as the difference between class types and instances of a class, multiple constraints on an individual or class and inheritance of type characteristics from a supertype. The advantage of using conceptual graphs is that they allow modeling of information without requiring that it be codified into If-THEN rules; i.e., knowledge can be applied in flexible ways as needed. Conceptual graphs are also being considered as a standard for knowledge interchange by the ANSI X3H4.6 task group on conceptual schema's IRDS committee (Information Resource Dictionary Systems) (Perez & Sarris, 1993). For a concise introduction to conceptual graphs, see (Polovina & Heaton, 1992) (Sowa, 1992).

Conceptual graphs support first-order logical inference using Peirce's beta rules — re-write rules governing the allowable transformations of a given set of conceptual graphs. Conceptual graphs are a visual representation that can also be expressed (albeit with less clarity) in a textual notation. In this paper, we generally show graphs in their display form, although to save space, we sometimes use the textual form. For example, the concept: $\boxed{\text{PERSON: *x}}$ meaning *some person, denoted by x*, is shown in text form as [PERSON: *x]. A group of concepts written together implies an AND relation between them. Negation is shown by marking a concept (or enclosed group of concepts, called a *context*) with a ¬ symbol. The beta rules support such logical constructs as *modus ponens* and double negation expressed in pictorial form. For example, a typical $P \supset Q$ logic rule can be expressed as $\neg\boxed{P \neg \boxed{Q}}$ in display form. With extensions, conceptual graphs can support reasoning about sets (Gardiner, Tjan, & Slagle, 1992; Tjan, Gardiner, & Slagle, 1992) and temporal logic (Delugach, 1991; Hartley, 1992; Moulin, 1992, 1993).

Conceptual Graph Inference Rules

Conceptual graphs have seven inference rules called the *beta rules*. An *enclosed graph* means a graph that is enclosed in nested negated contexts. An *evenly enclosed graph* is a graph enclosed in an even number of negative contexts A graph enclosed at level 0 (i.e., not subject to any negation) is considered an evenly enclosed graph. An oddly enclosed graph generally means its contents are being asserted to be false. In these descriptions *graph* means some graph in a given starting set.

The seven beta rules are the following:

Erasure. In an evenly enclosed negated graph, any graph may be erased, and any concept generalized. This is due to the fact that if something is true, its generalization is also true.

Insertion. In an oddly enclosed context, any graph may be inserted, and any graph restricted. This is due to the fact that if something is false, then its specialization is also false.

Iteration. A copy of any graph may be inserted into any context it dominates; i.e., any context nested within the graph's context. This corresponds roughly to the rule $A \wedge (B) \equiv A \wedge (A \wedge B)$

Deiteration. Any graph which could have been inserted through iteration may be erased.

Double negation. A double negation may be drawn around or removed from any graph, due to the fact that $\neg(\neg A) \equiv A$.

Coreferent join. Two identical coreferent concepts in the same context may be joined. This is a consequence of the definition for a line of identity.

Individuals. A generic concept may be instantiated if the instance appears in a context which dominates (i.e., encloses) the generic concept. This is because a nested constraint will apply to all the instances that dominate it.

These rules are only summarized here; for further details, consult (Sowa, 1984). The first five are from Peirce; the last two are Sowa's. These rules do not depend upon any database assumptions, and therefore apply to any knowledge that can be represented in conceptual graphs. The beta rules are equivalent to first-order predicate calculus. Sowa proves that the beta rules are both sound and complete. We will use conceptual graphs as our means of representing inference knowledge.

There are other rewrite rules that do not map directly to operations in logic; these rules are primarily notational conventions. For example, the rule of *name contraction* allows us to rewrite

[T] → (name) → [T: a] as simply [T: a].

Representing Inference Knowledge In Conceptual Graphs

The conceptual graphs of Peirce/Sowa provide our basis for collections of graphs representing:

- The database(s) of interest
- The sensitive targets of interest
- General knowledge
- Domain-specific knowledge related to the databases or sensitive targets of interest

In the AERIE inference classification scheme, target classes 1 and 2 (entities and activities) are modeled as conceptual graph concepts. Target classes 3, 4, and 5 (relationships among single entities and activities) are modeled as a relationship between conceptual graph concepts. Target class 6 (relationships between relationships) is modeled as relationship between two graphs, while target class 7 (new inference rules) is modeled as one of several rule forms in conceptual graphs.

With these models, we can identify inference methods using the conceptual graph inference rules with extensions and apply conceptual graph based operations to show how inference can be detected.

Inference Examples Using Conceptual Graphs

This section illustrates how conceptual graphs are used to represent the knowledge relevant to database inference analysis. Database inference analysis is performed with respect to one or more pieces of information desired to be kept secret. The first example shows inference class 2 — an activity materialization. The second example shows inference class 3 — an entity-to-entity relationship is materialized using, in part, information from a database. The third example shows another class 3 materialization — this one based on an existing inference technique known as *second path*. The fourth example shows class 5 — how an activity-to-entity relationship is materialized.

These examples use a sample database developed by the authors. The database is meant to resemble a practical database for a real-world enterprise. The translation from database instances to conceptual graphs is performed by a human knowledge engineer, because a database relation schema does not by itself determine any semantics for the elements in the relation. It is up to a knowledge engineer, in consultation with the designer or maintainer of the database, to decide on an appropriate semantics and ensure that those semantics are captured faithfully by a conceptual graph representation.

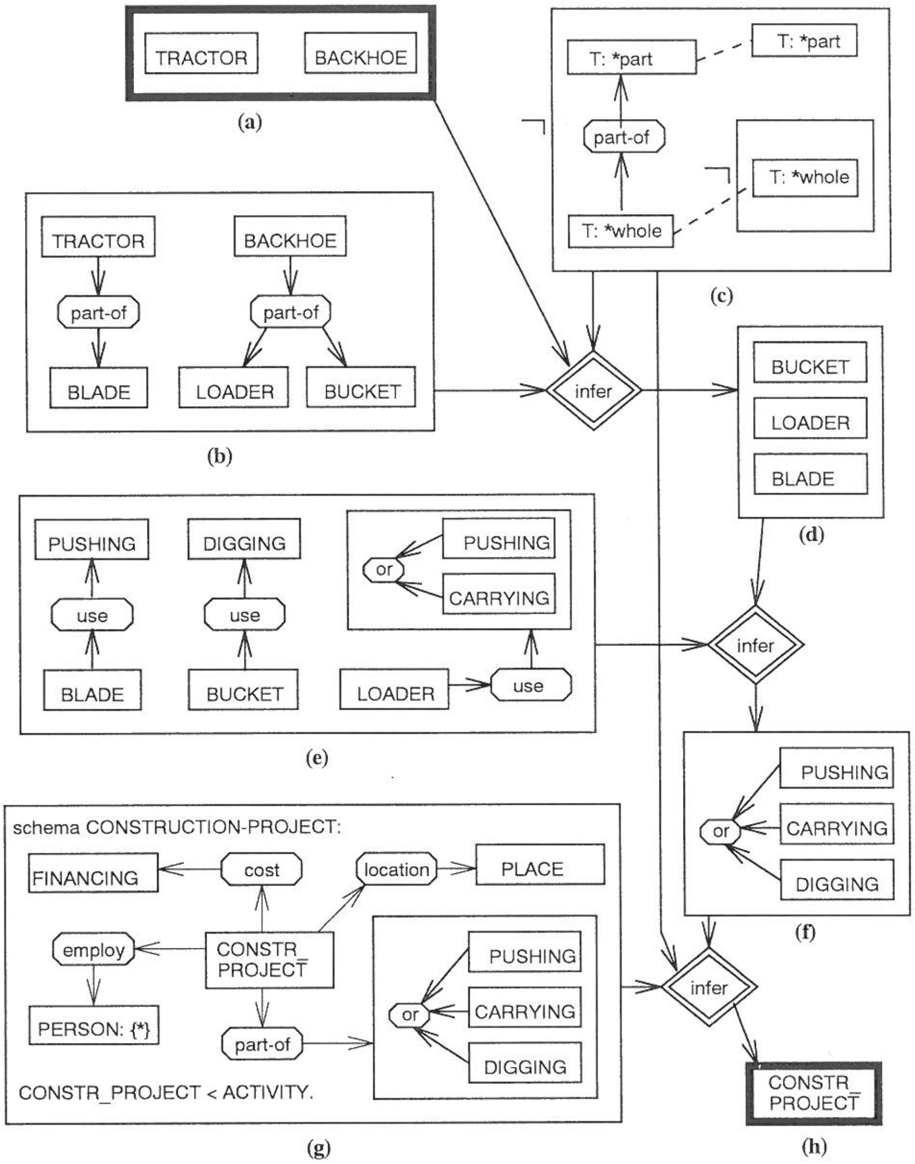

Fig. 1. Materializing Construction Project Activity.

Example 1: Materialization Of An Activity Using Domain-Specific Knowledge

We show in this example how domain-specific knowledge (i.e., not contained in a database) is used to materialize an activity. This example shows how the part-of relation and used-for relation interact to materialize an intermediate entity followed by an activity.

In this example, we assume information from a database has already materialized the entity *tractor* and the entity *backhoe*. We represent these in Fig. 1(a). Someone familiar with the construction equipment domain has domain-specific knowledge that *a blade is part of a tractor* and *a loader and bucket are both part of a backhoe*, as shown in Fig. 1(b). Using again the *part-of* rule shown in Fig. 1(c), we can infer the wholes from their parts, thereby materializing the bucket, loader and blade as in Fig. 1(d). Additional pieces of domain-specific knowledge are found in Fig. 1(e): *a blade is used for pushing*, *a bucket is used for digging*, and *a loader is used for pushing or carrying*. Joining Fig. 1(d) and (e), we get the materialization of the activities pushing, carrying or digging in Fig. 1(f). Finally from the schema for a construction project (also domain-specific knowledge), where digging, carrying, or pushing are part of a construction project, we can infer the existence of a construction project itself, as in Fig. 1(h).

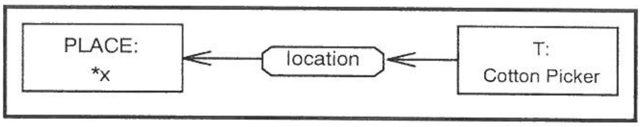

Fig. 2. Sensitive Target.

Example 2: Materializing An Entity-Entity Relationship

For this example, we assume that the sensitive target (i.e., the information desired to be kept secret) is: *The location X of a cotton picker.* We represent this target with the graph in Fig. 2.

Parts Database Instances

There is an agricultural equipment database available. A typical equipment database will contain information on various items ("parts") that are used in agriculture. Fig. 3(a) shows some of the instances it might contain.

Certain inventory items on hand and ordered are reserved for particular customers. Equipment reservations are kept in the database of Fig. 3(b). A customer may have some items ordered, or some already on-hand (perhaps waiting for customer pick-up) or some of both.

An equipment company's database will also contain information about the parts breakdown or "explosion" for each whole good item. That is, a composite part's component parts will be known. Sometimes the explosion is shown in a drawing to show the relative physical arrangement of parts; we assume that the part-of information is kept in a database relation, as shown in Fig. 3(c); in conceptual graphs, one instance of the schema is shown in Fig. 5(c).

Parts Catalog Relation

PartNo	Description	Class	UnitPrice	PrevSold
5500	Cotton picker	005	35467.00	9
G874-22	12V Battery	160	45.67	265
43729C	Spindle	505	1550.00	17
...	...	...	...	...

(a) Parts Catalog Relation.

Reserved Parts Relation

PartNo	CustNo	OnHand	Ordered
43729C	2846	0	7
93A4	376	2	4
...	...	...	...

(b) Reserved Parts Relation.

Parts Breakdown Relation

WholeNo	PartNo	QtyOfParts
5500	43729C	1
5500	G874-22	2
...	...	...

(c) Parts Breakdown Relation.

Fig. 3. Parts Relations.

Customer Information Relation

CustNo	CustomerName	CustomerAddr	LastSale	LastItem	SalesYTD
2846	Morgan Co.	987 Cherry Dr.	5 Dec 92	G874-22	2374.27
376	Wyatt Co.	765 Oak St.	7 Jul 92	93A4	43.21
1852	Clanton Corp.	127A Corral Av	15 Oct 92	G874-22	796.90
...	...	...	...	...	...

Fig. 4. Customer Information Relation.

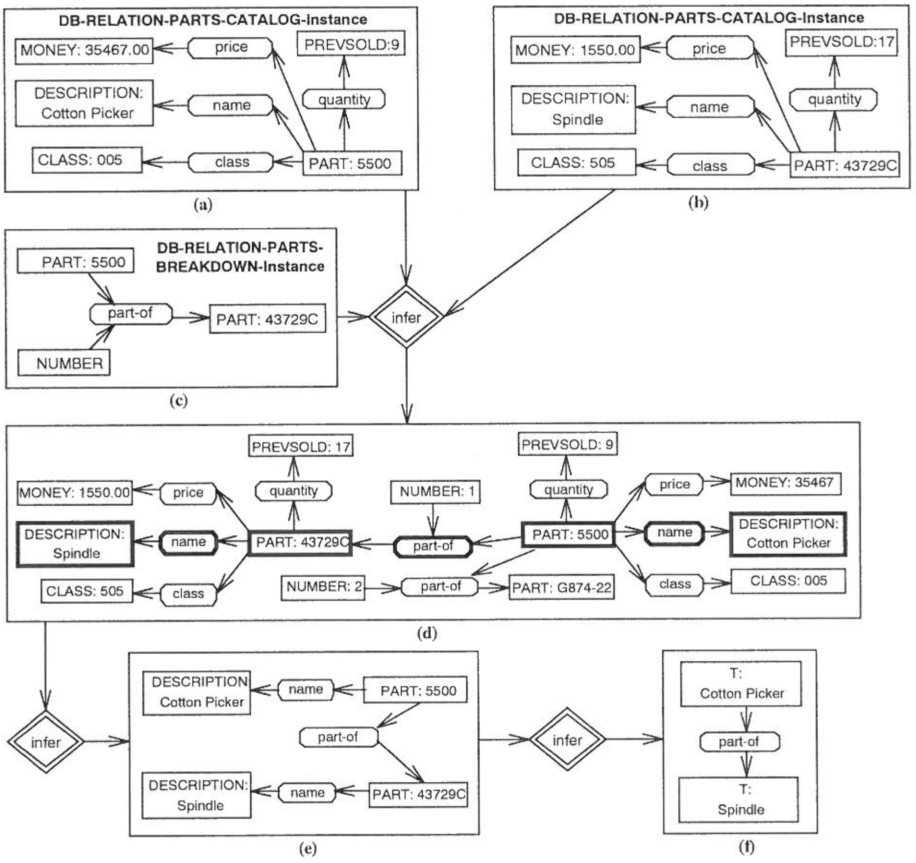

Fig. 5. Materialization Of Spindle-Partof-Picker Relationship

Customer information is kept in the database relation in Fig. 4. It contains a customer number, name and address, as well as the date and item of the customer's last sale, and the total sales by that customer for the year.

Once the schema's semantics are represented by relationships between generic concepts in a conceptual graph, database instances may be expressed as instances of those generic concepts. For example, Fig. 5(a) shows a cotton-picker instance, and Fig. 5(b) shows a spindle instance from the parts catalog relation in Fig. 3(a). Fig. 5(c) shows an instance of the relationship between spindle and cotton picker from the parts breakdown relation in Fig. 3(c).

The first inference step is to join the three graphs of Fig. 5(a), (b), and (c), to get the graph shown in Fig. 5(d).

Conceptual graph inference rules imply that we can generalize any existential graph within an evenly enclosed context (including depth zero) while preserving truth. Some parts of Fig. 5(d) are shown within thicker borders to make clear what subgraph we are using. A subgraph forms a generalization that becomes Fig. 5(e). We generalize by eliminating detail. By name contraction, Fig. 5(e) can be rewritten as shown in Fig. 5(f): *part of a cotton picker is a spindle.*

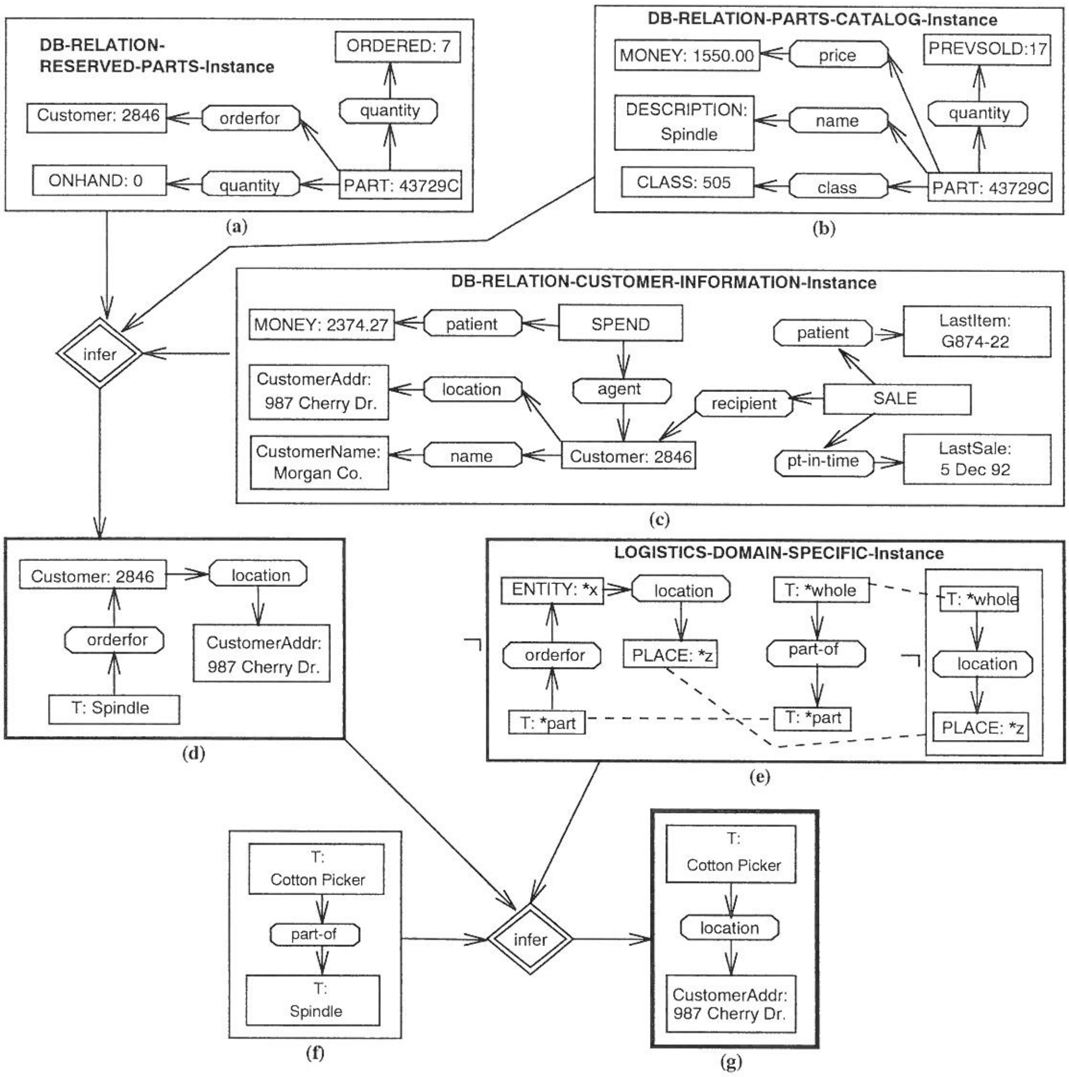

Fig. 6. Materialization Of Cotton Picker Location (Entity-Entity).

The next steps in the inference are shown in Fig. 6. The graph in Fig. 5(b) is repeated in Fig. 6(b) for clarity. From the reserved parts relation in Fig. 3(b), the graph in Fig. 6(a) is derived. From the customer information relation in Fig. 4, the graph in Fig. 6(c) is derived. Joining these graphs and then taking a subgraph (as in Fig. 5), we get the graph in Fig. 6(d), which represents the knowledge that *A spindle is ordered for customer 2846 who is located at 987 Cherry Dr.*

The AERIE approach assumes that many "common-sense" rules must be represented, such as the rule *If some entity orders something, and that something is a part of a whole, then there is an instance of the whole located where the entity is located.* This rule is shown in Fig. 6(e).

Since the type CustomerAddr is a subtype of Place, we can apply the rule to Fig. 6(d), and Fig. 6(f) (which was the result inferred from Fig. 5) to obtain the graph in Fig. 6(g), an instance of our original sensitive target in Fig. 2.

One important aspect to be learned from this example is that even seemingly simple inferences may involve more steps than apparent at first glance. Just to show that a spindle is part of a cottonpicker took several steps, due in part to the fact that relationships between things in the database are keyed to their part numbers rather than the things themselves.

There is an important constraint on the rule in Fig. 6(e) in that the part ordered must be unique to the whole that is inferred. If a part could be found in more than one whole item, then more than one valid inference could be made. Such a constraint is easily expressed in conceptual

Visitor Log Relation

Visitor Name	Company	Escort	Date	Time-in	Time-out
Susan	XYZ	Mike	Nov. 1, 1992	11:00 A.M.	12:00 P.M.
...	...	...	...	...	...

(a) Visitor Log Relation.

Job Accounting Relation

Employee Name	Job-number	Hours	Week-end date
Mike	5	40	Nov 5, 92
...	...	...	...

(b) Job Accounting Relation.

Job Number Relation

Project name	Job-number
AERIE	5
...	...

(c) Job Number Relation.

Fig. 7. Job Accounting and Visitor Log Relations.

graphs; it has been omitted for clarity in this example.

Example 3: Materializing An Entity-Entity Relationship Using Second Path

In materializing a relationship, previous work has centered around the notion of a *second path* between two things whose "first path" is the sensitive target (Hinke, 1988, 1990b; Binns, 1992b). Conceptual graphs represent second path inference analysis, but with additional power provided by generalization. A variant of this example has appeared in (Delugach, Hinke, & Chandrasekhar, 1993).

For this example, the sensitive target is *Any relationship between Company XYZ and Project Aerie.* In conceptual graph terms, this means that the two concepts will have some relationship between them.

We start with instances from a sample database. The following three relations contain information about the visitor log, job accounting by employees and job numbers used internally. Fig. 7(a) contains information from the sign-in sheets at the reception desk. Fig. 7(b) contains employee timecard information showing where employee effort is spent. Fig. 7(c) contains the internal job numbers used for accounting with respect to each project the company is working on.

Fig. 8 shows the conceptual graph representation of the entity-entity relationship materialized by this inference. Fig. 8(a) shows an instance from the relation in Fig. 7(b); Fig. 8(b) shows an instance from the relation in Fig. 7(c); and Fig. 8(c) shows an instance from the relation in Fig. 7(a).

The second path is shown as the set of concepts and relations in bold in the joined graph of Fig. 8(d). The second path can be paraphrased as: *Company XYZ contains a visitor Susan accompanied by escort Mike who works on job number 5 which is a characteristic of the Aerie project.*

The inference steps are simple joins; however, note that the concepts [ESCORT: Mike] and [EMPLOYEE: Mike] were joined to form the single concept [ESCORT: Mike]. Conceptual graphs provide the capability to use subtypes in

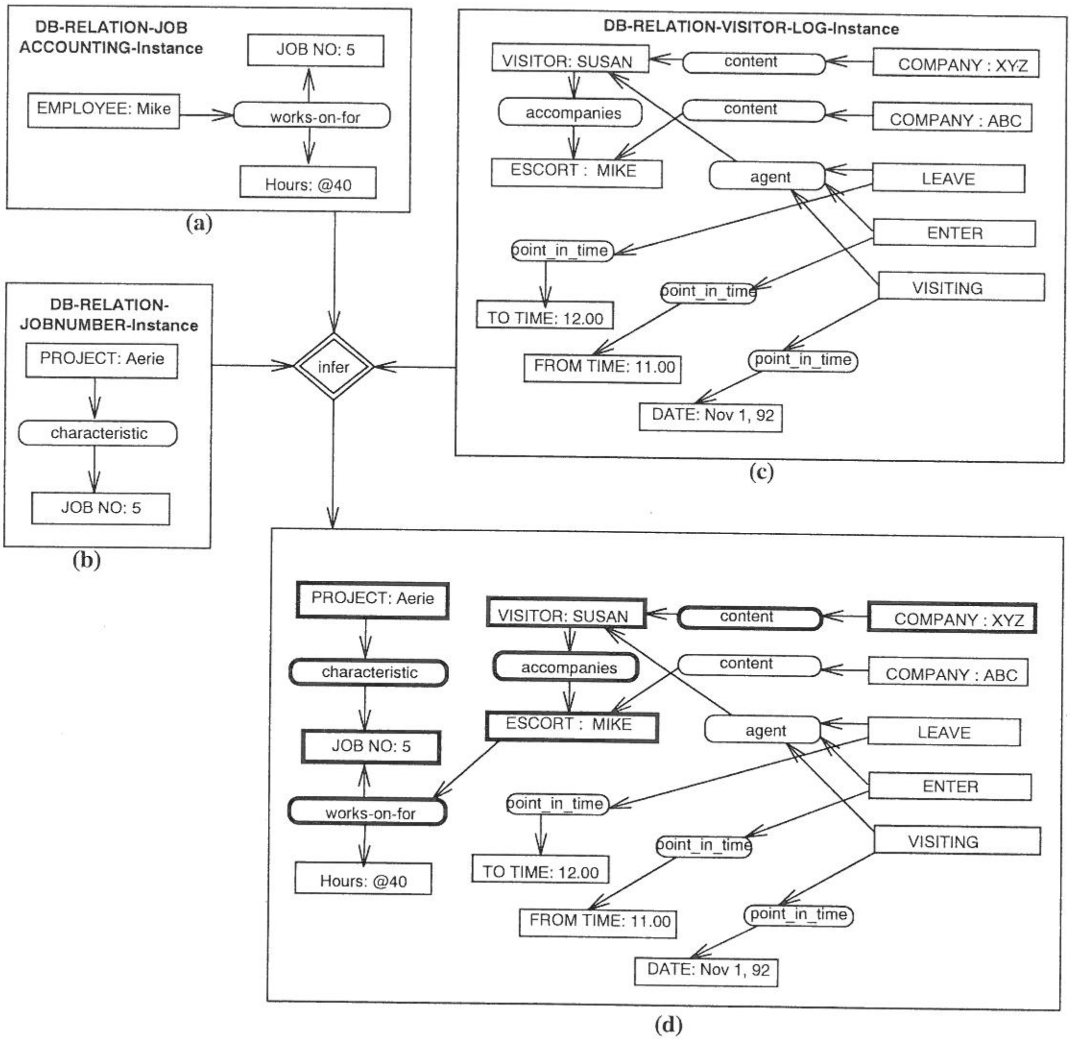

Fig. 8. Materialization Of Entity-Entity Relationship For Second Path.

joining graphs; such joins would not be possible using strict algebraic joins of the original relations. Of course, this added capability is a result of the knowledge engineer supplying semantics during his or her translation of the original relations into conceptual graphs.

Example 4: Performing An Activity-Entity Materialization From A Database

This example describes how an activity is materialized using primarily information in a database. For this example, we assume that the sensitive target (i.e., the information desired to be kept secret) is: *There exists the activity of cotton picking*. The mere existence of cotton picking can be represented by the conceptual graph of Fig. 9(h). All of the graphs in Fig. 9 are used to infer this sensitive target.

From the database, let us assume that a spindle part being ordered means a spindle exists, as shown in Fig. 9(a). Since an adversary may use any knowledge at his disposal to support inferences, we assume some general knowledge is available. One piece of "common" knowledge is the rule: *If something is a unique part of a whole, and the something is ordered, then the whole exists* as shown in Fig. 9(b).

For most useful inferences, an adversary might also use information about a particular domain. In this example, an adversary who is familiar with agricultural enterprises would know: *a spindle is part of a cotton picker* as in Fig. 9(c). We can specialize any graph within an oddly-enclosed context, so Fig. 9(b) and (c) permit us to directly write Fig. 9(d). Note that we have thereby instantiated an intermediate rule specific to the agricultural domain: *If a spindle exists and a spindle is part of a cotton-picker,*

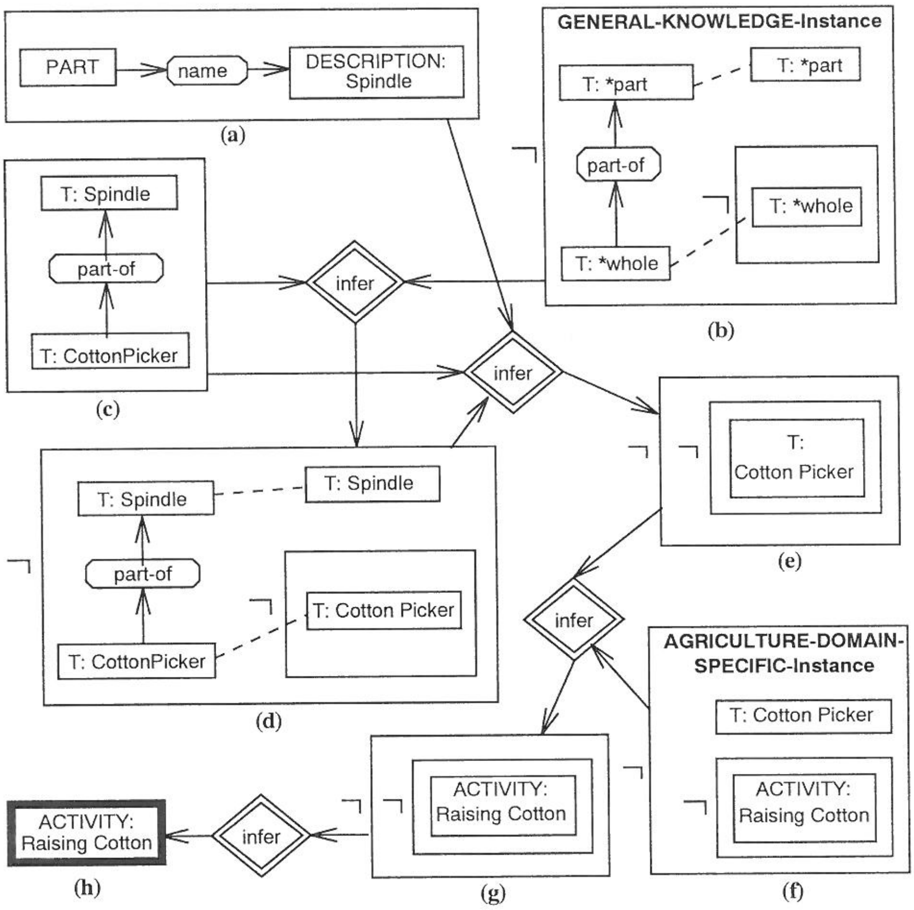

Fig. 9. Materialization Of An Activity-Entity Relationship.

then a cotton-picker exists. Since Fig. 9(a) can, through name contraction, be shown as [PART: Spindle], we can use deiteration on Fig. 9(d) to remove the [T : Spindle] concept; we can also use deiteration to remove the (part-of) relation from Fig. 9(d), resulting Fig. 9(e).

The next step in this inference process example involves a domain-specific rule from the agricultural domain: *If a cotton picker exists, then the activity raising cotton exists*, as shown in Fig. 9(f). From the result in Fig. 9(e) (which can be expressed as [T: Cotton Picker]), we can erase [T: Cotton Picker] from within Fig. 9(f) using deiteration to get Fig. 9(g). By the double negation rule from Fig. 9(f), we get Fig. 9(g) and we can now write [T: Raising Cotton], a materialized activity.

Discussion

We note some important limitations on the work we have presented here and contrast our work with that of deductive databases (for a summary see (Ullman, 1990)).

In this paper we have concentrated on the *representation* of inference analysis and how important classes of inference can be captured by conceptual graphs. Our examples do not deal with partial information, nor do we describe details of how different parts of conceptual graph **schema** are applied to infer new information. A conceptual graph schema is an arrangement of concepts that typically form a given construct (e.g., Fig 1(g) showing plausible features of a construction project) but do not form necessary and sufficient logical conditions for the existence of the construct.

Another limitation of the inference analysis is that if it cannot infer a sensitive target, then it is not decidable which of the following reasons apply:

- The inference analysis method is not sufficiently powerful; i.e., the inference involves meta-logical reasoning (e.g., defeasible rules, instance-specific facts).

- The pattern of inference has not been previously identified.
- The inference analysis lacks sufficient general or specialized knowledge to perform the inference.
- No possibility exists to infer the sensitive target under any circumstances.

If the inference analysis process is able to infer a sensitive target, then the likelihood exists that an adversary could perform the same inference. If the sensitive target can be inferred without using the database, however, then providing additional protection for data in the database will not prevent the inference. As SRI and others have noted, cover stories could be used to provide some plausible alternatives to the sensitive target (Binns, 1992a; Garvey & Lunt, 1991). If the sensitive target can be inferred using some data in the database, then that data within the database needs to be classified to break the inference path, or else cover stories could be used to sow confusion as suggested by SRI.

Some existing databases, called deductive databases, support interpreted queries that allow one to essentially derive information from a database which is not explicitly stored. Our work differs from that of deductive databases in two important ways:

- Our rules and operations are not database specific, but instead are general for a variety of databases. Even our domain-specific knowledge (e.g., information about cotton-picking) is applicable to several databases within the given domain. As a general knowledge representation, conceptual graphs offer the potential to bring pre-existing bodies of knowledge into the inference analysis process.
- We desire techniques to analyze existing databases, where rules and operations for deduction are not likely to have been included in a database's design. The AERIE approach allows us to take an already-populated database and analyze its contents, whether the database was specifically designed to support logical inference or not.

There are several enhancements to the AERIE approach that we are currently exploring:

- Inclusion of probabilistic and/or statistical inference techniques, such as discussed by Pearl (Pearl, 1988).
- Techniques for organizing both the general knowledge and sets of domain-specific knowledge to trim the search space for knowledge relevant to a given inference.
- Identification of different inference policies that permit differing interpretation of an inference result; e.g., is the inference certain, subject to change, based on some generalization, etc.?

Conclusion

This paper has introduced *inference target classes* as a new organizing framework for classifying database inferences, and shows hows conceptual graphs effectively model the information used to make the inferences. We began by identifying target classes as our categorization of different kinds of inference. We then showed how database inference knowledge can be represented using conceptual graphs.

The question of how to represent inferences is a first step toward automated analysis and detection of database inference problems in actual databases. Having a categorization framework means that for each class, we can organize different strategies for inference detection. We have shown that conceptual graphs have sufficient power to reperesent these inference classes.

Conceptual graphs are also a useful means of representing inference knowledge for our analysis purposes. For instance, since both an entity and activity are represented as concepts (e.g., [COTTON-PICKING] and [COTTON-PICKER]), the mechanism for inferring either of them in conceptual graphs will be similar. Conceptual graphs therefore do not require separate methods for inferring the two different classes. (We maintain the distinction in case any important differences arise later in our work.)

Conceptual graphs also provide a unified representation; an If-then rule is merely a negated context where no new operations are required.

Our current focus is in developing analysis techniques based upon both the AERIE approach and conceptual graphs. We are investigating how various inference paths can be detected. We are interested both in a theoretical characterization of various kinds of inference, as well as the development of practical tools that will organize and manage the (possibly deep) knowledge required to perform database inference analysis on real databases. We eventually expect to offer design guidance for database designers and administrators that will make them aware of inference problems in their databases and help them find solutions to these problems.

Acknowledgments

This work was supported under Maryland Procurement Office Contract No. MDA904-92-C-5146. We thank Asha Chandrasekhar, of the University of Alabama in Huntsville, for her contributions to the example dealing with second path analysis. Reviewers of an early draft of this paper also provided helpful comments.

BINNS, L. J., (1992a). Inference Through Cover Stories. In *Proceedings of the Sixth IFIP 11.3 Working Conference on Database Security*. IFIP, Vancouver, Ontario, Canada.

BINNS, L. J., (1992b). Inference Through Secondary Path Analysis. In *Proceedings of the Sixth IFIP 11.3 Working Conference On Database Security*. IFIP, Vancouver, Ontario, Canada.

BUCZKOWSKI, L. J., (1990). Database Inference Controller. In Landwehr, C. E. (Ed.), *Database Security, III: Status and Prospects*. North-Holland, Monterey, CA, U.S.A. Results of the IFIP Working Group 11.3 Workshop on Database Security.

COX, L. H., (1988). Modeling and Controlling User Inference,. In Landwehr, C. E. (Ed.), *Database Security: Status and Prospects*. North-Holland.

DELUGACH, H. S., (1991). Dynamic Assertion and Retraction of Conceptual Graphs. In Way, E. C. (Ed.), *Proceedings of the Sixth Annual Workshop on Conceptual Graphs*, pp. 15–26 Binghamton, New York. SUNY Binghamton.

DELUGACH, H. S., HINKE, T. H. & CHANDRASEKHAR, A., (1993). Applying Conceptual Graphs for Inference Detection Using Second Path Analysis. In *Proceedings of the First International Conference on Conceptual Structures*, pp. 188–197 Laval University, Quebec City, Canada.

DENNING, D., (Ed.). (1982). *Cryptography and Data Security*. Addison-Wesley.

GARDINER, D. A., TJAN, B. S. & SLAGLE, J. R., (1992). Extending Conceptual Structures: Representation Issues and Reasoning Operations. In (Nagle et al., 1992), pp. 67–86. Ellis Horwood.

GARVEY, T. D., & LUNT, T. F., (1991). Cover Stories for Database Security. In *Proceedings of the Fifth IFIP Working Group 11.3 Working Conference on Database Security* Shepherdstown, WV, U.S.A.

HARTLEY, R. T., (1992). A Uniform Representation For Time And Space And Their Mutual Constraints. In (Lehmann, 1992).

HINKE, T. H., (1988). Inference Aggregation Detection in Database Management Systems. In *IEEE Symposium on Security and Privacy* Oakland, CA, USA.

HINKE, T. H., (1990a). Database Inference Engine Design Approach. In Landwehr, C. E. (Ed.), *Database Security, II: Status and Prospects*. North-Holland, Kingston, Ontario, Canada. Results of the IFIP Working Group 11.3 Workshop on Database Security, October, 1988.

HINKE, T. H., (1990b). Response to Research Question 3 in Research Questions List, Answers, and Revision. In Landwehr, C. E. (Ed.), *Database Security, III: Status and Prospects*. North-Holland, Monterey, CA. Results of the IFIP Working Group 11.3 Workshop on Database Security, Sept. 1989.

IRDS93 (1993). Perez, Sandra and Anthony Sarris, editors, Information Resource Dictionary System (IRDS) Conceptual Schema (CS) Technical Report. Tech. rep. X3H4/92-003 and ISO/IEC JTC1/SC21 N7486, American National Standards Institute (ANSI) X3H4 IRDS and the International Organization for Standardization (ISO).

LEHMANN, F., (1992). *Semantic Networks in Artificial Intelligence*. Pergamon Press, Oxford.

LUNT, T. F., (1989). Aggregation and Inference: Facts and Fallacies. In *IEEE Computer Society Symposium on Security and Privacy* Oakland, CA, USA.

MATLOFF, N. S., (1988). Inference Control vs. Query Restriction vs. Data Modification: A Perspective. In Landwehr, C. (Ed.), *Database Security: Status and Prospects*. North-Holland.

MINEAU, G. W., MOULIN, B. & SOWA, J. F., (Eds.). (1993). *Conceptual Graphs for Knowledge Representation*. No. 699 in Lecture Notes in Artificial Intelligence. Springer-Verlag.

MORGENSTERN, M., (1987). Security and Inference in Multilevel Database and Knowledge-Base Systems. In *Proceedings of SIGMOD (ACM Special Interest Group on Management of Data*. Association for Computing Machinery.

MORGENSTERN, M., (1988). Controlling Logical Inference in Multilevel Database Systems. In *1988 IEEE Symposium on Security and Privacy* Oakland, CA, USA.

MOULIN, B., (1992). Modeling Temporal Knowledge in Discourse: A Refined Approach. In (Pfeiffer, 1992), pp. 89–98.

MOULIN, B., (1993). The representation of linguistic information in an approach used for modeling temporal knowledge in discourses. In (Mineau et al., 1993), pp. 182–204.

NAGLE, T., NAGLE, J., GERHOLZ, L. & EKLUND, P., (Eds.). (1992). *Conceptual Structures: Current Research and Practice.* Ellis Horwood.

National Computer Security Center (1985). *Trusted Computer System Evaluation Criteria.* Unitd States Department of Defense. DoD 5200.28.STD.

National Computer Security Center (1991). *Trusted Database Management System Interpretation of the Trusted Computer System Evaluation Criteria.* National Computer Security Center. NCSC-TG-021, Version-1.

PEARL, J., (1988). *Probabilistic Reasoning In Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, San Mateo, Calif.

PFEIFFER, H., (Ed.). (1992). *Proceedings of the Seventh Annual Conceptual Graphs Workshop.* Springer Verlag. Las Cruces, NM, U.S.A., July 8-10.

POLOVINA, S. , & HEATON, J., (1992). An Introduction to Conceptual Graphs. *AI Expert*, 36–43.

QIAN, X., STICKEL, M. E., KARP, P. D., LUNT, T. F. & GARVEY, T. D., (1993). Detection and Elimination of Inference Channels in Multilevel Relational Database Systems. In *Proceedings 1993 IEEE Computer Society Symposium on Research in Security and Privacy*, pp. 196–205.

SOWA, J. F., (1984). *Conceptual Structures: Information Processing in Mind and Machine.* Addison-Wesley, Reading, MA.

SOWA, J. F., (1992). Conceptual Graphs Summary. In (Nagle et al., 1992), pp. 3–52.

SU, T.-A. , & OZSOYOGLU, G. (EDS.)., (1990). *Multivalued Dependency Inferences in Multilevel Relational Database Systems*, Monterey, CA. North-Holland.

THURAISINGHAM, B., (1991). The Use of Conceptual Structures for Handling the Inference Problem, and Cover Stories for Database Security. In *Proceedings of the Fifth IFIP WG 11.3 Working Conference on Database Security* Shepherdstown, WV, U.S.A.

TJAN, B. S., GARDINER, D. A. & SLAGLE, J. R., (1992). Representing and Reasoning with Set Referents and Numerical Qualifiers. In (Nagle et al., 1992), pp. 53–66. Ellis Horwood.

ULLMAN, J. D., (1990). The Theory Of Deductive Database Systems. In *Proceedings of the 35th IEEE Computer Society International Conf (COMPCON).*

Received: March, 1993
Accepted: January, 1995

Contact address:

Computer Science Department
University of Alabama in Huntsville
Huntsville, AL 35899 U.S.A.
Phone: (205) 895-6614
FAX: (205) 895-6239
E-mail: delugach@cs.uah.edu
thinke@cs.uah.edu

HARRY S. DELUGACH has been an Assistant Professor at the University of Alabama in Huntsville since 1990. He received his Ph.D. in Computer Science at the University of Virginia. His current research interests are in the areas of knowledge representation (particularly conceptual graphs) and software development.

THOMAS H. HINKE an Associate Professor of Computer Science who joined the UAH Computer Science Faculty in 1990 after a 16-year career in industry. He received his Ph.D. in Computer Science from the University of Southern California. He has been an active researcher in the area of computer security, including database security since the mid-1970's.