Applying Conceptual Graphs for Inference Detection Using Second Path Analysis ⋆

Harry S. Delugach
Thomas H. Hinke
Asha J. Chandrasekhar

Computer Science Department,
University of Alabama in Huntsville,
Huntsville, AL. 35899
Electronic mail: delugach@cs.uah.edu
Telephone: (205) 895-6614
FAX: (205) 895-6239

Abstract. Conceptual graphs are applied to a problem in database inference known as second path analysis. A sensitive target is proposed, namely to keep secret that a certain company is working on a certain project. Typically available non-sensitive information is introduced in the form of relational database schemata. These schemata are then expressed as conceptual graphs, followed by a series of joins which are performed revealing the sensitive target. It is shown that conceptual graph representations are able to analyze second path problems and achieve the same results as previous techniques. A new inference path was discovered that had not been identified by previous work.

1 Introduction

In this paper, we discuss the role of conceptual graphs in analyzing a particular problem in database inference called the *second-path inference method*. At this stage of the research, we have succeeded in representing appropriate knowledge using conceptual graphs. Our goal is to ultimately use guided transformations to determine whether a given database is vulnerable to inference problems.

This paper is organized as follows. Section 2 describes general features of the inference detection problem, and summarizes an approach to that problem called *second path analysis*. Section 3 shows a complete example of how second path analysis is accomplished using conceptual graphs to represent all the relevent knowledge of the analysis. Section 4 discusses some issues raised by this approach. Section 5 summarizes the results and outlines some anticipated future work.

2 Inference Detection Through Second Path Analysis

The concept of second path analysis for inference detection was originally proposed by Hinke [9, 10] based on Chen's ER-model [7] as a means to infer a sensitive relationship between entities, which in Chen's model are just undefined things. Additional work on second path analysis has been performed by Binns [2, 3, 4].

⋆ This work was supported under Maryland Procurement Office Contract No. MDA904-92-C-5146

Under the original formulation of second path analysis, data within an enterprise was modeled using entities and relationships between entities. This data model is a simplification of that proposed in Chen's ER-model, since the distinction between entities and attributes, as descriptors of entities, is dropped. Under this simplified model, only entities and relationships between entities are considered. It is assumed that the information relevant to an enterprise is modeled as a graph with entities and relationships. Entities are represented as nodes and relationships are represented as links. Each of the links in an enterprise model has associated with it a security sensitivity label or classification. It is assumed that security controls are applied by the organization responsible for the data to ensure that sensitive data about these relations are released only to authorized people. While some of these relationships may be highly sensitive and releasable only to those with proper clearances, other data may be public and available to one's adversary.

The inference problem arises when an adversary is able to infer sensitive relationships that are more highly classified than the data to which he has been given access. In this case, what the adversary seeks is a relationship between two entities, where the direct path connecting them is protected through an appropriate classification label at a security level above that authorized to the adversary. With second path analysis, the adversary can infer this direct relationship by discovering a second, non-direct path, that joins these two entities. To understand how second path analysis can be used to perform inference analysis, an example based on one by Hinke [9] will be presented.

Consider a sensitive project that has a number of companies supporting it. For various reasons, the names of the companies supporting this particular project may be sensitive. In effect, what is sensitive is the relationship between project and company. This sensitive relationship is not made available to our adversary. Now assume further that an employee of one of these supporting companies visits the research center that is performing the project to attend a meeting for the project. If the attendee list can be obtained by the adversary, and if this list contains the names of the companies, or if through such means as professional society membership lists the employers of the attendees can be retrieved, then the inference can be made using the second path shown in Figure 1.

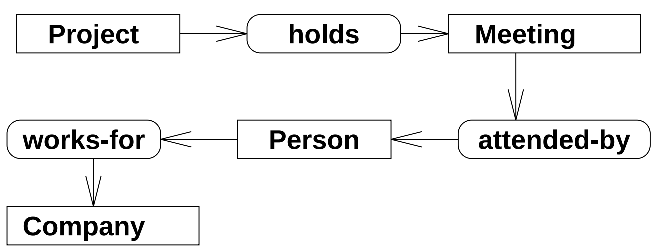

Fig. 1. Second Path Via Meeting

The example can be altered from the original in [9] to analyze different paths. To make the problem more challenging, suppose that the attendee list is not made available; however, there is some additional information that can be obtained by our adversary. This might include data showing which rooms are reserved for meetings by some employee, perhaps a project secretary. In addition, in this research center, all visitors must be escorted. Now, if the escort of an attendee of the project meeting is the same person that reserved the meeting room, we have another second path that can be used to make the project-company inference. This new second path is shown in Figure 2.

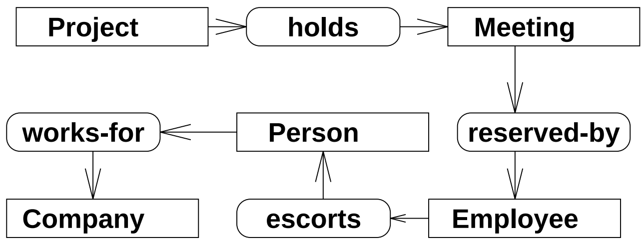

Fig. 2. Second Path Via Escort

Note this may not be a correct inference, since it is possible that a visitor is attending a meeting other than the one for which the escort reserved a room. However, in some cases this may provide an accurate inference.

From the previous base of research, the current research project, called AERIE, is attempting to push inference detection forward in a number of new areas. The first area is in the data structures used. While Thuraisingham [16] has noted that the simple ER-model used in [9] is a primitive type of conceptual graph, the current AERIE research has adopted conceptual graphs [14] as the basic data structure with which to describe the enterprise's databases as well as a means for representing relevant specialized and generalized knowledge.

The second area in which the AERIE research is attempting to extend inference work is through the classification of various types of inferences, based on the nature of the data that is being inferred, and the identification of appropriate inference approaches for each type of inference [6, 11]. One of these inference types is the entity-to-entity type for which second path analysis is just one of the inference detection methods. The extension of second path analysis to conceptual graphs represents a new thrust of this analysis technique. The next section of this paper describes an example of how second path inference detection technique can be extended to conceptual graphs.

3 Second Path Analysis with Conceptual Graphs

This section discusses an example of second path analysis using conceptual graphs. In this example, the sensitive target is knowing which project a company is working on, based on whether any of its employees attends a meeting accompanied by a person who works for that project. Our example is mainly to illustrate how the sensitive information can be inferred from non-sensitive information found in typical databases. For our purposes, it is not essential that the information be available in actual databases, *per se*, only that the information be available (even in handwritten form) to an adversary.

3.1 Example Database Description

Typical relational database schemata are shown below with particular instances. The corresponding conceptual graph representation is also shown for each of the schemata. Table 1 represents a typical sign-in log, where visitors to an installation check-in at a reception area. Table 2 represents some of a company's payroll accounting information. Table 3 represents a list of meeting room reservations. Table 4 represents the list of projects and jobs numbers.

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

Table 1. Schema for Visitor Log of Company ABC

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

Table 2. Schema for Payroll Accounting in Company ABC

Employee Name	From time	Totime	Date
Mike	11:00 A.M.	12:00 P.M.	Nov. 1, 1992

Table 3. Schema for Meeting Room Reservations

Project name	Job-number
AERIE	5

Table 4. Schema for Projects and Job Numbers

3.2 Initial Conceptual Graphs

A knowledge engineer will translate each database schema into conceptual graphs to capture the semantics of the database relations. The graph in Figure 3 represents the schema for the visitor-log, the graph in Figure 4 represents the payroll accounting schema, Figure 5 represents the the meeting room reservations schema and the graph in Figure 6 represents the project schema.

3.3 Inference Detection with Conceptual Graphs

We begin with a sensitive target which is the following: *Company XYZ is involved in the project AERIE*. Looking at Figure 3, we see that there is no direct relationship connecting the

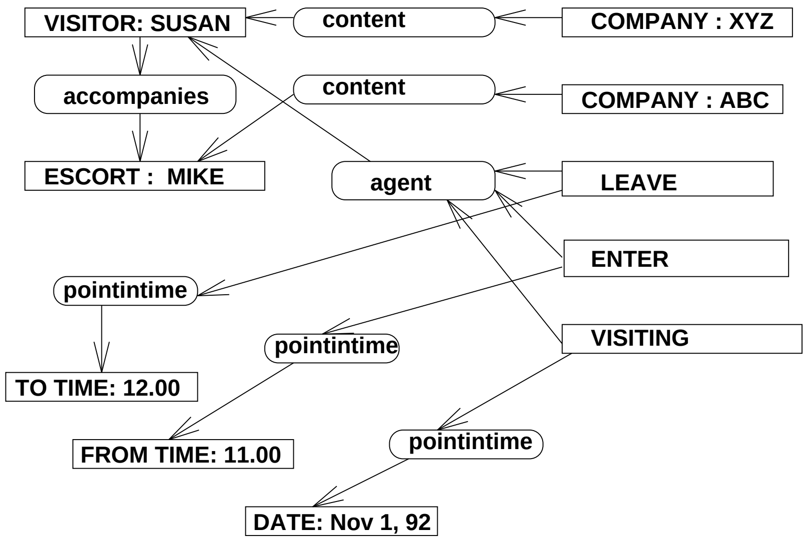

Fig. 3. Graph Representing Visitor Log

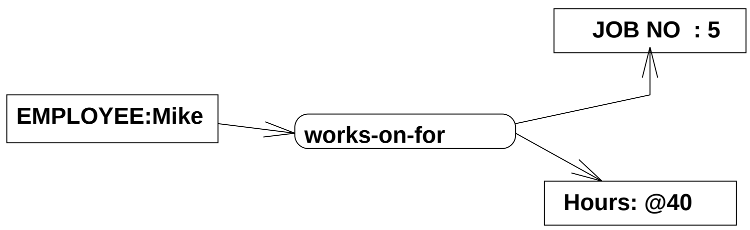

Fig. 4. Graph Representing Payroll Accounting

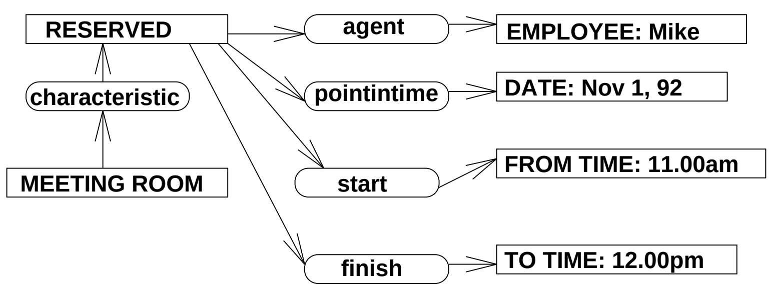

Fig. 5. Graph Representing Room Reservations

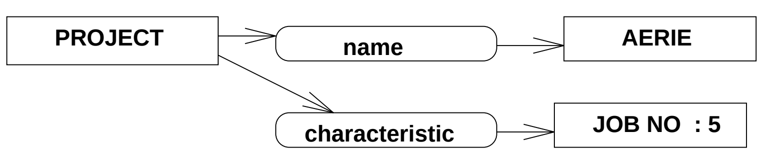

Fig. 6. Graph Representing Projects and Job Numbers

two concepts XYZ and AERIE. We start with Figure 3. The only other graph with a common concept is Figure 5. We therefore apply a join to the Figure 3 and Figure 5 on the common concepts [TO_TIME], [FROM_TIME] and [DATE]. Also the concept [ESCORT: Mike] in Figure 3 can be generalized to [EMPLOYEE: Mike] and joined with [EMPLOYEE: Mike] in Figure 5. The resulting graph of both these joinings is Figure 7.

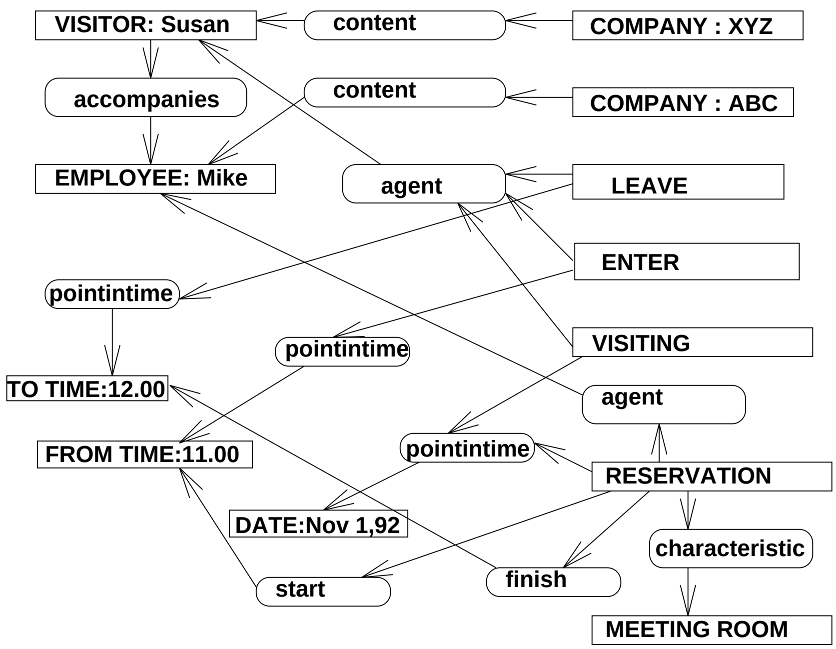

Fig. 7. Graph Joining Figure 3 and Figure 5

Figure 7 does not yet show any relationship between company XYZ and project AERIE. Note that the join of Figure 4 and Figure 6 gives a relationship between the concepts [PROJECT] and [EMPLOYEE] where the common concept on which they are joined is [JOB_NO]. The resulting graph is Figure 8; however, this does not yet give us any relationship between company and project.

Thus Figure 7 shows a relationship between [COMPANY] and [EMPLOYEE: Mike] and Figure 8 shows a relationship between [EMPLOYEE: Mike] and [PROJECT]. We can join the concept [EMPLOYEE: Mike] in Figure 7 and the concept [EMPLOYEE: Mike] in Figure 8 resulting in Figure 9. The second path in Figure 9 is shown by a dotted line between [VISITOR: Susan] and [EMPLOYEE: Mike] and is expressed in English as follows: *Susan of company XYZ accompanies Mike who works on job number 5 which is a characteristic of the project named AERIE*.

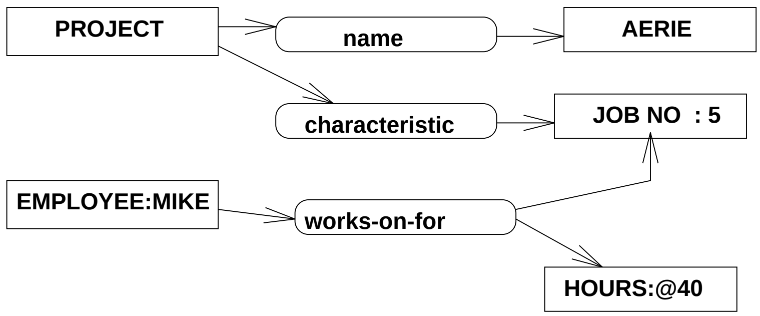

Fig. 8. Graph Joining Figure 4 and Figure 6

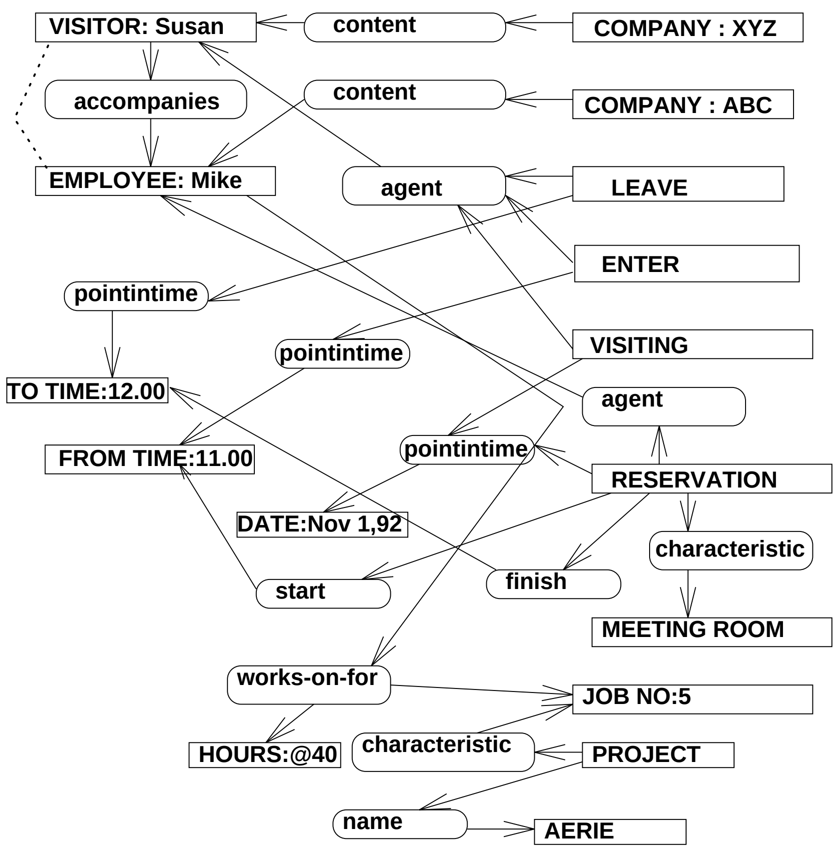

Fig. 9. Graph Joining Figure 7 and Figure 8

Thus the second path is obtained by applying conventional conceptual graph operations to the database schemata's graphs.

4 Discussion

An interesting observation can be made in comparing this work to the same example given by Hinke in [9], where the second path used the meeting to arrive at the inference. But here the accounting information associated with the escort is used to connect the company with the project. In other words the fact that the escort works on a project connects the visitor and so her company with the project. The conceptual graph representation therefore helped us find an additional second path that the previous work did not reveal.

We note that the semantics of the arrow directions have nothing to do with finding the second path. Without the arrow directions we can still see a path between company and project. But the arrow directions are essential to expressing the second path relationship in English.

The second-path analysis shown above is merely the result of applying a series of joins and seeing the plausibility of whether an inference can be made. This final graph may not be the exact answer for this particular inference problem, since there are other interpretations that fit the given information. For instance, Mike's company may be supporting a number of projects and it can't be said definitely that Susan's company is involved specifically in this project; it may be some other project, or in another instance, Mike may be escorting her but she may be attending a project which may not necessarily involve Mike. In either case, the current analysis could only say that there is a possible inference vulnerability, which is the main purpose of this research effort.

The analysis shown here does demonstrate a clear advantage of conceptual graphs over and above the capabilities offered by simple algebraic joins. In our example, we made use of subtype information (e.g., ESCORT is a subtype of EMPLOYEE) that allowed joins on the two concepts. Strict algebraic joins cannot exploit the similarity of these two concepts the way that conceptual graph joins can.

Characterizing database inference in terms of conceptual graphs is important for these reasons:

- It is valuable to show that conceptual graphs semantics can, at the very least, solve existing, well-known database problems, in order to substantiate the claims that conceptual graphs are more powerful than existing database techniques.
- Conceptual graphs are often used in situations where inferences are to be drawn; this work adds one new method to the toolkit of available inference techniques.
- Since current databases are being incorporated as part of larger knowledge-based frameworks, it is useful to develop conceptual graph approaches that capture database semantics, and not just their content.

5 Conclusion

We have shown that one of the most accepted and understood database inference techniques – second-path analysis – can be accurately and succinctly captured by using conceptual graphs. We have provided a general approach using conceptual graphs, and shown a complete example of second-path analysis to illustrate both the problem and its solution. We are incorporating this technique into a collection of techniques that will be developed by the AERIE project. We believe that database inference is an excellent application domain in which to apply conceptual graphs.

The inference detection problem involves identifying inferences that may be made using knowledge from a database along with some amount of general knowledge. While our particular area of research is concerned with preventing such inferences once they have been identified, we feel it has relevance to several problems involving databases in general. It has applicability to the general problem of making intelligent inferences about database queries, where a query manager can aid a questioner in refining or clarifying a query, or even suggest other related databases whose information might be of interest to the questioner. Identifying inference paths is useful in preventing unwanted inferences, as well as in facilitating additional inferences in other situations.

The current work addresses an already-known problem in database inference and shows how conceptual graphs can characterize a solution to this problem. Our future work is aimed toward addressing new problems in database inference, using deep knowledge about specific domains and then identifying new inference methods. We are attempting to partition different kinds of database knowledge being represented, and are pursuing an approach to pre-analyze pieces of knowledge before they are added to the larger knowledge base. We intend to use conceptual graphs to solve the various classes of database inference targets which we have identified in our research. Some of them are materialization of an activity, materialization of a sensitive relationship between two or more materialized activities,etc. We are also studying the suitability of conceptual graphs for specifying database design guidelines which can make the database immune to the inference attack by an adversary.

References

1. Committee on Multilevel Data Management Security, *Multilevel Data Management Security*, Air Force Studies Board, National Research Council, National Academy Press, Washington, D.C., 1983.
2. Binns, Leonard J., "Inference Through Polyinstantiation", *Proceedings of the Fourth RADC Database Security Workshop*, April 1991.
3. Binns, Leonard J., "Inference Through Cover Stories", *Proceedings of the Sixth IFIP 11.3 Working Conference on Database Security*, Vancouver, Canada, August 1992.
4. Binns, Leonard J., "Inference Through Secondary Path Analysis", *Proceedings of the Sixth IFIP 11.3 Working Conference on Database Security*, Vancouver, Canada, August 1992.
5. Buczkowski, L. J.: "Database Inference Controller", in *Database Security III, Status and Prospects*, North-Holland, 1990.
6. Delugach, Harry S. and Thomas H. Hinke. AERIE: Database Inference Modeling and Detection Using Conceptual Graphs. *Proc. 7th Annual Workshop 0n Conceptual Graphs*, New Mexico State University, Las Cruces, New Mexico, July 8-10, 1992.
7. Chen, Peter Pin-Shan, The Entity-Relationship Model - Toward a Unified View of Data. *ACM Transactions on Database Systems*, Vol. 1, March 1976.
8. Garvey, Thomas D. and Teresa F. Lunt, Cover Stories for Database Security, *Proceedings of the 5th IFIP WG 11.3 Working Conference on Database Security*, Shepherdstown, WV, USA, Nov 1991.
9. Hinke, Thomas H.: Inference Aggregation Detection in Database Management Systems. *1988 IEEE Symposium on Security and Privacy*, Oakland, CA, USA, April 1988.
10. Hinke, Thomas H., "Database Inference Engine Design Approach," in *Database Security, II: Status and Prospects, Results of the IFIP WG 11.3 Workshop on Database Security*, Kingston, Ontario, Canada, C. E. Landwehr, ed., North-Holland, 1990.
11. Hinke, Thomas H. and Harry Delugach, "AERIE: An Inference Modeling and Detection Approach for Databases," *Proceedings of the 6th IFIP 11.3 Working Conference on Database Security*, Vancouver, Canada, August 1992.
12. Lunt, Teresa: Aggregation and Inference: Facts and Fallacies. *Proceedings of the 1989 IEEE Symposium on Research in Security and Privacy*, Oakland, CA USA, May 1989.
13. Morgenstern, Matthew: Security and Inference in Multilevel Database and Knowledge-Base Systems, *Proceedings of SIGMOD* 1987, Association of Computing Machinery, 1987.
14. Sowa, John F., *Conceptual Structures: Information Processing in Mind and Machine*. Addiion-Wesley, Reading, MA., 1984.
15. Su, Tzong-An and Gultekin Ozsoyoglu: *Data Dependencies and Inference Control in Multilevel Relational Database Systems*, IEEE Symposium of Security and Privacy, Oakland, CA., April 1987.
16. Thuraisingham, Bhavani: "The Use of Conceptual Structures for Handling the Inference Problem, Cover Stories for Database Security", *Proceedings of the 5th IFIP WG 11.3 Working Conference on Database Security*, Shepherdstown, WV, USA, Nov 1991.

This article was processed using the LaTeX macro package with LLNCS style