

A White Paper

Semantic Integration of Databases

Dr Harry Delugach,

The University of Huntsville in Alabama

Dr David Skipper

Bevilacqua Research Corporation

The Problem

A customer has access to a large number of computer databases, describing many, possibly overlapping, enterprises. Assuming that these are typical relational databases, each has one or more associated relation files. Likewise, this customer has access to personnel who are familiar with the details of these databases and who can describe the purpose of the database.

However, it is not assumed that mere access to the database schema assures the database can be understood. This is because the field names of a database have no inherent meaning without the context from the enterprise where they were developed and used. As an illustration, a relation EMPLOYEE may have the following fields:

| Name | ID | Address | Position | Starting Date |
|------|----|---------|----------|---------------|
|------|----|---------|----------|---------------|

An observer may think the meanings of these fields are obvious, but there may be multiple interpretations of their meanings. For example, **Starting Date** may mean (a) the date the employee began working for the enterprise, (b) the date the employee started in the **Position** given, or (c) the date the employee became vested in a retirement plan. Such multiple interpretations can render the meaning of the fields ambiguous.

In a typical deployed database, there also may be relationships between the field contents that are not obvious from the schema. For example, there may be a defined **Position** "NONE" to indicate that the employee has been terminated; in such a case, the "starting" date may actually be the termination date. This would constitute a meaning decided upon by the database designers, and kept consistent through the programming that accesses the database. The importance of this example is that this particular meaning was established outside the database schema, therefore this additional constraint must be provided by other means.

Another database relation TRAVEL might have these fields:

| Name | ID | Destination | Travel Date |
|------|----|-------------|-------------|
|------|----|-------------|-------------|

It would be convenient to work with the combined set of EMPLOYEE and TRAVEL data. In this example, consider correlating each **Position** with a **Destination**. If a single database with a consistent design was utilized, it would be possible using standard database operations to simply join the two relations to form a new relation. That new relation looks like this:

| Name | ID | Address | Position | Starting Date | Destination | Travel Date |
|------|----|---------|----------|---------------|-------------|-------------|
|------|----|---------|----------|---------------|-------------|-------------|

This relation could then be sorted by **Position**. Within a given database, such operations are routine. In fact, such operations are an important reason to use relational databases. Forming such a new relation (either formally or in an ad-hoc manner) is a way of constructing a *second path* to a relationship between **Position** and **Destination**. This approach to a *second path* was developed in previous works on conceptual modeling of databases [7, 9, 11, 12, 13].

Unfortunately, the above example only works because it is possible to assume that **Name** and **ID** have exactly the same meaning (whatever that meaning was) in both relations. This is due to the "single database with a consistent design" assumption noted above. With a collection of databases, however, such a "join" is hard to do, for conceptual reasons. A field name may appear in more than one database, but, as already discussed, how do we know that it means the same thing in both?

The Conventional Approach

Joining disparate databases provides additional relationships to enhance understanding of the enterprise's data, unless the conceptual differences in the databases defeat the attempt. The obvious approach is to utilize a single team to develop a consistent design of each database, with a consistent conceptual basis. The single team ensures a consistent conceptual basis. A formal design, such as Entity Relation diagrams combined with textual descriptions of the meanings, provides a basis for joining relations. There are difficulties with this approach. First, this process is labor intensive, with limited options for automation. Second, in order to provide consistent conceptual meanings, the team must be stable while all of the databases are re-designed. Finally, the textual conceptual meanings are still open to interpretation, so that future joins may require the services of the original team to supply definitions. An approach is needed which clearly captures the conceptual meanings independent of particular personnel and which provides a clear path to automation of substantial portions of the process.

Our Proposed Approach

Our approach is to establish conceptual meaning of the database in a formal consistent manner using techniques described below. The conceptual meaning is first captured for two or more databases. When their meaning has been represented, the representations are combined to form a conceptual ontology of the combined domains. Once that is accomplished, then the process of reasoning proceeds as in

Figure 1 on the next page.

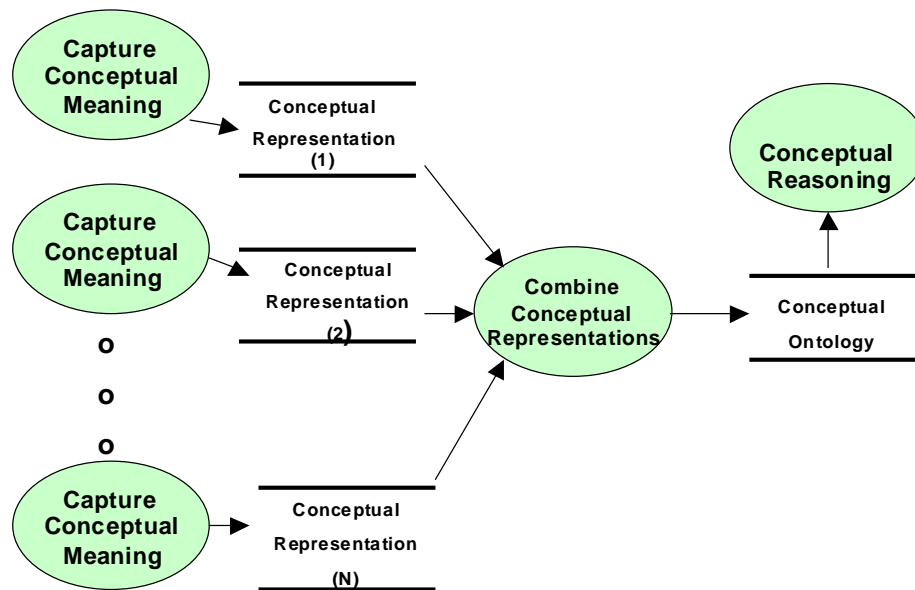


Figure 1. Conceptual Processing Model

Key to this approach are the techniques and representations being develop to address major the aspects of such a solution architecture. Our capabilities are as follows:

Capture Conceptual Meaning - Tracked Repertory Grids [16, 17, 18], an extension of repertory grids [10], will be used for knowledge acquisition in conjunction with conceptual graphs, as introduced in [6]. An earlier approach called *microanalysis* may be tried in parallel to fine-tune the graphs [8]. Our intent is to construct conceptual graph representations of the database semantics.

Combine Conceptual Representations - Since multiple databases constitutes a form of multiple views, our previous work in multiple-viewed conceptual graph representations is relevant [2, 3, 4, 5].

Conceptual Reasoning / Data Mining - One of our strengths has been the ability to represent and manipulate concepts that represent activities or processes. Through the use of actors conceptual graphs can become animate agents which model enterprise operations. Data mining (i.e., the triggering of automatic searching through a database) is also well matched to the semantics of actors within conceptual graphs [14]. Our long experience in the practical application of actors in graphs has led to our extension of them into *demons* to support dynamic assertion and retraction of graphs in a knowledge base – which is an essential feature of practical reasoning [1, 15]. This extension is now gaining acceptance in the conceptual graph community.

Our approach is to demonstrate the power of combining these techniques by a reasonably sized application of these techniques. For example, two databases (of a customer's choosing) can be semantically analyzed, and combined to support reasoning. The proposed architecture is shown in Figure 2. In summary the steps are as follows:

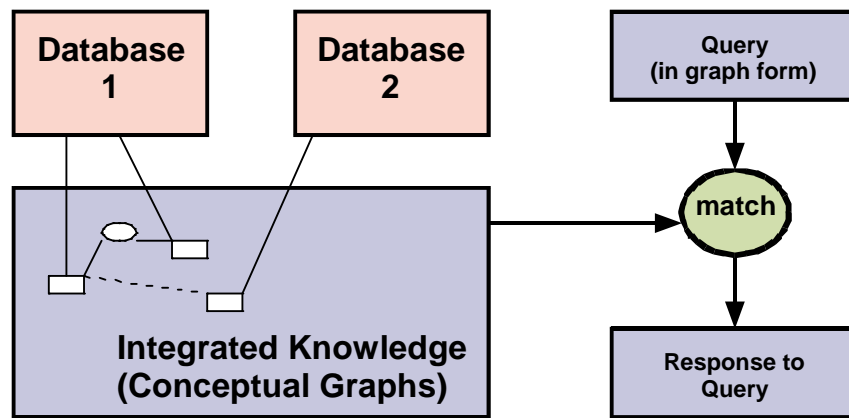


Figure 2. Database Integration Architecture

1. Use our existing conceptual graph processing tools, including display editors and tools to implement dynamic actors using bounded neural networks. These tools will then be leveraged to provide knowledge engineering support at the start of the effort.

Establish the semantics of one database (with instances), and represent the results using conceptual graphs. This is shown as Database 1 in

2. Figure 1. The initial technique will be the microanalysis mentioned above [8, 9]. This effort will be supplemented by instance-level analysis on the populated database and supported by our conceptual graph tools. The intent in this phase is to develop an understanding of how to represent database semantics using conceptual graphs in a useful way. This process should take about six months at a \$75K LOE.
3. Establish the semantics of a second database, this time employing some of the automated techniques, which have potential in building representations efficiently. This process should take six months at a \$75K LOE, and outline strategy for integrating the two databases.

After these steps have defined some advanced tool concepts, and a general process, as well as proving the validity of that process, the heart of the work commences: namely, learning how to integrate the databases using their conceptual graph semantic representations, and respond to queries based on the integrated knowledge. This is step 4 and it constitutes our intent for a follow-on effort.

4. Establish an integrated representation of the two databases in steps 1 and 2, supported by general tools for making some simple queries. This phase would be at a "best effort" level over a period of one to two years.

References

- [1] Harry S. Delugach, "Dynamic Assertion and Retraction of Conceptual Graphs," presented at Proc. Sixth Annual Wkshop on Conceptual Graphs, Binghamton, New York, 1991.
- [2] Harry S Delugach, "A Multiple Viewed Approach to Software Requirements," in *Computer Science Department*. Charlottesville, VA: University of Virginia, 1991.
- [3] Harry S. Delugach, "Analyzing Multiple Views Of Software Requirements," in *Conceptual Structures: Current Research and Practice*, P. Eklund, T. Nagle, J. Nagle, and L. Gerholz, eds., Ellis Horwood, 1992, pp. 391-410.
- [4] Harry S. Delugach, "Specifying Multiple-Viewed Software Requirements With Conceptual Graphs," *Jour. Systems and Software*, vol. 19, no. , pp. 207-224, 1992.
- [5] Harry S. Delugach, "An Approach To Conceptual Feedback In Multiple Viewed Software Requirements Modeling," in *Viewpoints 96: International Workshop on Multiple Perspectives in Software Development, Joint Proc. of the SIGSOFT'96 Workshops*. San Francisco, 1996, pp. 242-246.
- [6] Harry S. Delugach, "Repertory Grid Graphs: A Hybrid of Conceptual Graphs and Repertory Grids," in *Proc. 11th Knowledge Acquisition Workshop (KAW-98)*, Brian Gaines and Mark Musen, eds. Banff, Alberta, Canada, University of Calgary, 1998.
- [7] Harry S. Delugach and Thomas H. Hinke, "Using Conceptual Graphs To Represent Database Inference Security Analysis," *Jour. Computing and Info. Tech.*, vol. 2, no. 4, pp. 291-307, 1994.
- [8] Harry S. Delugach and Thomas H. Hinke, "Microanalysis: Acquiring Database Semantics In Conceptual Graphs," in *Conceptual Structures: Knowledge Representation as Interlingua*, vol. 1115, *Lecture Notes in Artificial Intelligence*, P.W. Eklund, G. Ellis, and G. Mann, eds., Springer Verlag, 1996.
- [9] Harry S. Delugach and Thomas H. Hinke, "Wizard: A Database Inference Analysis And Detection System," *IEEE Trans. Knowledge and Data Engr.*, vol. 8, no. 1, pp. 56-66, 1996.
- [10] B. R. Gaines and M.L.G. Shaw, "Knowledge Acquisition Tools Based on Personal Construct Psychology," *Knowledge Engr. Review*, vol. 8, no. 1, pp. 49-85, 1993.
- [11] Thomas H. Hinke and Harry S. Delugach, "AERIE: An Inference Modeling and Detection Approach For Databases," in *Database Security, VI: Status and Prospects*, vol. A-21, *IFIP Transactions*, B. W. Thuraisingham and C. E. Landwehr, eds. Amsterdam, Elsevier Science Publ. (North-Holland), 1993.
- [12] Thomas H. Hinke and Harry S. Delugach, "Security-Oriented Database Inference Detection," presented at Proc. 17th Natl. Comp. Security Conf., Baltimore, MD, 1994.
- [13] Thomas H. Hinke, Harry S. Delugach, and Asha Chandrasekhar, "A Fast Algorithm For Finding Second Paths in Database Inference Analysis," *Jour. Computer Security*, vol. 3, no. 2/3, 1995/1996.
- [14] Thomas H. Hinke, Harry S. Delugach, and Randall P. Wolf, "A Framework For Inference Directed Data Mining," in *Database Security Volume X: Status and Prospects*, Pierangela Samarati and Ravi S. Sandhu, eds., Chapman and Hall for the International Federation of Information Processing (IFIP), 1997, pp. 229-239.

- [15] Ryszard Raban and Harry S. Delugach, "Animating Conceptual Graphs," in *Conceptual Structures: Fulfilling Peirce's Dream*, vol. 1257, *Lecture Notes in Artificial Intelligence*, Dickson Lukose, Harry S. Delugach, Mary Keeler, Leroy Searle, and John F. Sowa, eds., Springer-Verlag, 1997, pp. 431-445.
- [16] Randall P. Wolf, "Knowledge Acquisition via the Integration of Personal Constructs and Conceptual Graphs," in *Computer Science Department, Ph.D. dissertation*. Huntsville, AL, USA: University of Alabama in Huntsville, 1998.
- [17] Randy P. Wolf and Harry S. Delugach, "Knowledge Acquisition via the Integration of Repertory Grids and Conceptual Graphs," presented at ICCS '96, 4th Intl. Conf. on Conceptual Structures, University of New South Wales, Sydney, Australia, 1996.
- [18] Randall P. Wolf and Harry S. Delugach, "Knowledge Acquisition via Tracked Repertory Grids," Computer Science Dept., Univ. Alabama in Huntsville, Technical Report TR-UAH-CS-1996-02, 1996.