
# *Specialty Technology for Relationship Analytics*

**A Monash Information Services Bulletin**

**by**

**Curt A. Monash, Ph.D.**

**November, 2006**

**Sponsored by:**

**Cogito**

# Specialty Technology for Relationship Analytics

*Conventional analytics explores tabular arrays of numbers.*

Conventional analytic technology does a good job of analyzing numerical relationships. Data mining uncovers subtle yet important correlations. Spreadsheet, planning, and budgeting tools allow consequences to be inferred from known relationships. And conventional business intelligence (BI) allows more casual explorations.

*Real-world relationships are hard to model in this way.*

Some kinds of real-world relationships, however, cannot be expressed so well in numbers. Yet these need to be explored too. Terrorists may connect via friends, financiers, or shared residences. Fraudsters connect in similar ways, or perhaps via service providers. Web pages are connected by complex hyperlink structures. Life-saving drug possibilities may be found in intricate biological pathways.

*The goal is to find relationship patterns.*

In each of these cases, the challenge is to find a pattern of relationships. Is there an innocent-seeming hub from which terrorist support flows? Are a group of insurance claimants improbably interlinked? Is one website really being recommended by many independent referrers? Often, there simply isn't the kind of numerical array needed for conventional analytic techniques to work.

Graph *models are the key.*

Even so, computerized analysis can help in such pursuits. The natural mathematical model for such problems is well-known to computer scientists: A *directed graph.* Nodes can consist of people, places, companies, and schools. Or genes and proteins. Or web pages. Edges (aka *arcs*) could be phone calls, blood relationships, or known attendance. Or hyperlinks. Or chemical interactions.

*One can find, count, and visualize myriad relationship paths.*

Once such information is stored as a logical graph, there's a lot one can do with it. Much of this analysis is based on finding, counting, and/or visualizing *paths* – i.e., connections, usually with intermediate links. One can look to see if there are any paths at all connecting two nodes, thus perhaps finding a valid biological process, or the means to commit a crime. Or one can check which node has the shortest average connection to a group of others – thus finding a good candidate for the hub of a terrorist network, or the influencer of a set of consumers. And if there are any groups of nodes with high degrees of mutual interconnection – well, then maybe that's a fraud network.

*Cogito's technology does* relationship analytics.

Last year, I coined a phrase for this kind of analysis – *relationship analytics.* It was during my first briefing from Cogito, a startup company focusing on this kind of data model and application. They address

applications of this kind – and if you have that kind of problem, it may be worthwhile hearing what Cogito has to say about solving it.

*The RDBMS alternative works mainly for simpler graphs.*

The problem Cogito tries to solve is this: Relational DBMS have difficulties with complex graph-theoretical analysis. It's easy to manage a graph where every path has length 1; that's just a three-column table (node, edge, node). And for paths of length two, you can join the edge table to itself and materialize the resulting view (Oracle uses this strategy). But for longer paths, the relational approach breaks down. It's hard to optimize performance relationally when you're targeting graphs of longer path-lengths. And as SQL guru Joe Celko has pointed out, it's extremely hard to write SQL for graph analysis if the path lengths are long, variable, or not known in advance.

*Cogito has a very interesting approach.*

Cogito's answer is to offer a data manager and associated tools that are graph-oriented to the core. Key elements include:

- A proprietary data manipulation language, GQL (Graph Query Language), along with an SDK, that make graphical queries straightforward.
- A processing engine for graphical queries, with hybrid disk-based/memory-centric operation.
- Support in the SDK for a range of graph-theoretic analytic metrics calculations, organized around the concept of "centrality."
- A variety of visual tools for graphical query generation, filtering, visualization, etc.
- Other necessary capabilities, such as data integration.

*Its data structures mimic graph topology.*

The basic idea of the processing engine is to simply store node-edge-node triples as atomic data, along with pointers to adjacent nodes. The data can then be clustered according to node proxmity. In principle, how well this works would seem to depend on the topology of the graph, but favorable results have been reported in a variety of application scenarios. For example, based on what we've heard from more than one text mining company, it seems that there's some Cogito-based processing of security-related, text-extracted (or perhaps voice-sourced) information, in or on behalf of US national security.

*These can beat relational alternatives.*

In another example, MyFamily.com – owner of Ancestry.com – wanted to let visitors find relationships between famous people and themselves. This involves looking for paths 10-20+ edges long, on a graph with over 200 million nodes and 1 billion edges. Query rates are on the order of 20/second, or 2 million/day. They didn't find an affordable relational solution. But with Cogito they have a working system.

*It comes down to* path length.

In deciding between conventional DBMS and specialty graph-oriented tools such as Cogito's, there's one key criterion: *Path length.* If path lengths are short and predictable, there's a good chance that relational DBMS and their forthcoming extensions can do the job. In complex graphs with longer paths, however, relational approaches may not scale well. In such cases, specialty technologies warrant serious consideration.

## About the Author

For a quarter-century, Curt Monash has been a leading analyst of and strategic advisor to the software industry. Praised by Lawrence J. Ellison for his "unmatched insight into technology and marketplace trends," Curt was the software/services industry's #1 ranked stock analyst while at PaineWebber, Inc., where he served as a First Vice President until 1987. Since 1990 he has owned and operated Monash Information Services, a highly acclaimed technology analysis firm focused on enterprise software. He has been extensively published and quoted in the technology and general business press, and has been a regular columnist for *Application Development Trends, Software Magazine,* and *Computerworld.*

Prior to his business career, Curt earned a Ph.D. in Mathematics (Game Theory) from Harvard University at the age of 19. He has held faculty positions in mathematics, economics and public policy at Harvard, Yale, and Suffolk Universities. For more information please see www.monash.com.

## About the Sponsor

Cogito, Inc. is a pioneer in the development of Graph-based Relationship Analytics, improving data structuring, modeling, definition and analysis. This approach to data analytics gives organizations new insight into complex and distant relationships previously hidden within massive data stores.

Cogito has spent many man-years researching and modeling data structures, knowledge management algorithms, and data visualization techniques. The resulting product—the Cogito Knowledge Center—represents a compelling alternative to relational database managers. Where standard relational databases deal with sets of data elements, Cogito represents data in a mathematical graph. This structure can be far more flexible for users who need to analyze relationships between data elements.

The company can be reached via www.cogitoinc.com .