

---

# Foreword

What do the following things all have in common: marketing attribution analysis, anti-money laundering (AML) analysis, customer journey modeling, safety incident causal factor analysis, literature-based discovery, fraud network detection, internet search node analysis, map application creation, disease cluster analysis, and analyzing the performance of a William Shakespeare play. As you might have guessed, what these all have in common is the use of graphs, proving that Shakespeare was right when he declared, “All the world’s a graph!”

Okay, the Bard of Avon did not actually write *graph* in that sentence, he wrote *stage*. However, notice that the examples listed above all involve entities and the relationships between them, including both direct and indirect (transitive) relationships. Entities are the nodes in the graph—these can be people, events, objects, concepts, or places. The relationships between the nodes are the edges in the graph. Therefore, isn’t the very essence of a Shakespearean play the active portrayal of entities (the nodes) and their relationships (the edges)? Consequently, maybe Shakespeare could have written *graph* in his famous declaration.

What makes graph algorithms and graph databases so interesting and powerful isn’t the simple relationship between two entities, with A being related to B. After all, the standard relational model of databases instantiated these types of relationships in its foundation decades ago, in the entity relationship diagram (ERD). What makes graphs so remarkably important are directional relationships and transitive relationships. In directional relationships, A may cause B, but not the opposite. In transitive relationships, A can be directly related to B and B can be directly related to C, while A is not directly related to C, so that consequently A is transitively related to C.

With these transitivity relationships—particularly when they are numerous and diverse, with many possible relationship/network patterns and degrees of separation between the entities—the graph model uncovers relationships between entities that otherwise may seem disconnected or unrelated, and are undetected by a relational

database. Hence, the graph model can be applied productively and effectively in many network analysis use cases.

Consider this marketing attribution use case: person A sees the marketing campaign; person A talks about it on social media; person B is connected to person A and sees the comment; and, subsequently, person B buys the product. From the marketing campaign manager's perspective, the standard relational model fails to identify the attribution, since B did not see the campaign and A did not respond to the campaign. The campaign looks like a failure, but its actual success (and positive ROI) is discovered by the graph analytics algorithm through the transitive relationship between the marketing campaign and the final customer purchase, through an intermediary (entity in the middle).

Next, consider an anti-money laundering (AML) analysis case: persons A and C are suspected of illicit trafficking. Any interaction between the two (e.g., a financial transaction in a financial database) would be flagged by the authorities, and heavily scrutinized. However, if A and C never transact business together, but instead conduct financial dealings through safe, respected, and unflagged financial authority B, what could pick up on the transaction? The graph analytics algorithm! The graph engine would discover the transitive relationship between A and C through intermediary B.

In internet searches, major search engines use a hyperlinked network (graph-based) algorithm to find the central authoritative node across the entire internet for any given set of search words. The directionality of the edge is vital in this case, since the authoritative node in the network is the one that many other nodes point at.

With literature-based discovery (LBD)—a knowledge network (graph-based) application enabling significant discoveries across the knowledge base of thousands (or even millions) of research journal articles—"hidden knowledge" is discovered only through the connection between published research results that may have many degrees of separation (transitive relationships) between them. LBD is being applied to cancer research studies, where the massive semantic medical knowledge base of symptoms, diagnoses, treatments, drug interactions, genetic markers, short-term results, and long-term consequences could be "hiding" previously unknown cures or beneficial treatments for the most impenetrable cases. The knowledge could already be in the network, but we need to connect the dots to find it.

Similar descriptions of the power of graphing can be given for the other use cases listed earlier, all examples of network analysis through graph algorithms. Each case deeply involves entities (people, objects, events, actions, concepts, and places) and their relationships (touch points, both causal and simple associations).

When considering the power of graphing, we should keep in mind that perhaps the most powerful node in a graph model for real-world use cases might be "context." Context may include time, location, related events, nearby entities, and more. Incorpor-

porating context into the graph (as nodes and as edges) can thus yield impressive predictive analytics and prescriptive analytics capabilities.

Mark Needham and Amy Hodler's *Graph Algorithms* aims to broaden our knowledge and capabilities around these important types of graph analyses, including algorithms, concepts, and practical machine learning applications of the algorithms. From basic concepts to fundamental algorithms to processing platforms and practical use cases, the authors have compiled an instructive and illustrative guide to the wonderful world of graphs.

— Kirk Borne, PhD  
Principal Data Scientist and Executive Advisor  
Booz Allen Hamilton  
March 2019