

# THE VALUE OF USING KNOWLEDGE GRAPHS IN SOME COMMON USE CASES

*By: Dave McComb*

*February 20, 2019*



We've been asked to comment on the applicability of Knowledge Graphs and Semantic Technology in service of a couple of common use cases. We will draw on our own experience with client projects as well as some examples we have come to from networking with our peers.

The two use cases are:

- Customer 360 View
- Compliance

We'll organize this with a brief review of why these two use cases are difficult for traditional technologies, then a very brief summary of some of the capabilities that these new technologies bring to bear, and finally a discussion of some case studies that have successfully used graph and semantic technology to address these areas.

## Why is This Hard?

In general, traditional technologies encourage complexity, and they encourage it through ad-hoc introduction of new data structures. When you are solving an immediate problem at hand, introducing a new data structure (a new set of tables, a new json data structure, a new message, a new API, whatever) seems to be an expedient. What is rarely noticed is the accumulated effect of many, many small decisions

taken this way. We were at a healthcare client who admitted (they were almost bragging about it) that they had patient data in 4,000 tables in their various systems. This pretty much guarantees you have no hope of getting a complete picture of a patient's health and circumstances. There is no human that could write a 4,000 table join and no systems that could process it even if it were able to be written.

This shows up everywhere we look. Every enterprise application we have looked at in detail is 10-100 times more complex than it needs to be to solve the problem at hand. Systems of systems (that is the sum total of the thousands of

application systems managed by a firm) are 100- 10,000 times more complex than they need to be. This complexity shows up for users who have to consume information (so many systems to interrogate, each arbitrarily different) and developers and integrators who fight a read guard action to keep the whole at least partially integrated.

*Knowledge Graphs and Graph Databases have gained a lot of mind share recently as it has become known that most of the very valuable digital native firms have a knowledge graph at their core.*

Two other factors contribute to the problem:

- Acquisition – acquiring new companies inevitably brings another ecosystem of applications that must be dealt with.
- Unstructured information – a vast amount of important information is still represented in unstructured (text) or semi-structured forms (XML, Json, HTML). Up until now it has been virtually impossible to meaningfully combine this knowledge with the structured information businesses run on.

Let's look at how these play out in the customer 360 view and compliance.

## Customer 360

Eventually, most firms decide that it would be of great strategic value to provide a view of everything that is known about their customers. There are several reasons this is harder than it looks. We summarize a few here:

- Customer data is all over the place. Every system that places an order, or provides service, has its own, often locally persisted set of data about “customers.”
- Customer data is multi-formatted. Email and customer support calls represent some of the richest interactions most companies have with their clients; however, these companies find data from such calls difficult to combine with the transactional data about customers.
- Customers are identified differently in different systems. Every system that deals with customers assigns them some sort of customer ID. Some of the systems share these identifiers. Many do not. Eventually someone proposes a “universal identifier” so that each customer has exactly one ID. This almost never works. In 40 years of consulting I've never seen one of these projects succeed. It is too easy to underestimate how hard it will be to change all the legacy systems that are maintaining customer data. And as the next bullet suggests, it may not be logically possible.

- The very concept of “customer” varies widely from system to system. In some systems the customer is an individual contact; in other, a firm; in another a role; in yet another, a household. For some it is a bank account (I know how weird that sounds but we’ve seen it).
- Each system needs to keep different data about customers in order to achieve their specific function. Centralizing this puts a burden of gathering a great deal of data at customer on-boarding time that may not be used by anyone.

## Compliance

The primary reason that compliance related systems are complex is that what you are complying with is a vast network of laws and regulations written exclusively in text and spanning a vast array of overlapping jurisdictions. These laws and regulations are changing constantly and are always being re-interpreted through findings, audits, and court cases.

The general approach is to carve off some small scope, read up as much as you can, and build bespoke systems to support them. The first difficulty is that there are humans in the loop all throughout the process. All documents need to be interpreted, and for that interpretation to be operationalized it generally has to be through a hand-crafted system.

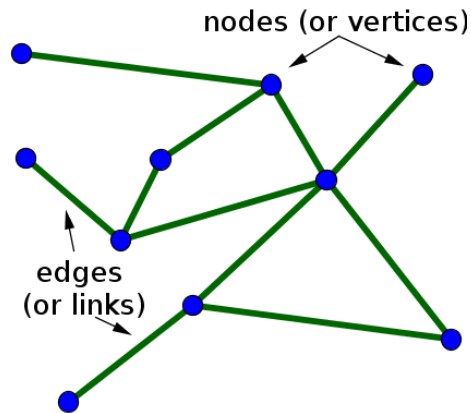
## A Brief Word on Knowledge Graphs and Semantic Technology

Knowledge Graphs and Graph Databases have gained a lot of mind share recently as it has become known that most of the very valuable digital native firms have a knowledge graph at their core:

- Google – the google knowledge graph is what has made their answering capability so much better than the key word search that launched their first offering. It also powers their targeted ad placement.
- LinkedIn, Facebook, Twitter – all are able to scale and flex because they are built on graph databases.
- Most Large Financial Institutions – almost all major financial institutions have some form of Knowledge Graph or Graph Database initiative in the works.

## Graph Databases

A graph database expresses all its information in a single, simple relationship structure: two “nodes” are connected by an “edge.”



A node is some identifiable thing. It could be a person or a place or an email or a transaction. An “edge” is the relationship between two nodes. It could represent where someone lives, that they sent or received an email, or that they were a party to a transaction.

A graph database does not need to have the equivalent of a relational table structure set up before any data can be stored, and you don’t need to know the whole structure of the database and all its metadata to use a graph database. You can just add new edges and nodes to existing nodes as soon as you discover them. The network (the graph) grows organically.

The most common use case for graph databases are analytic. There are a whole class of analytics that make use of network properties (i.e., how closely x is connected to y, what the shortest route is from a to b).

## Knowledge Graphs

Most graph databases focus on low level data: transactions, communications, and the like. If you add a knowledge layer onto this, most people refer to this as a knowledge graph. The domain of medical knowledge (diseases, symptoms, drug/drug interaction, and even the entire human genome) has been converted to knowledge graphs to better understand and explore the interconnected nature of health and disease.

Often the knowledge in a knowledge graph has been harvested from documents and converted to the graph structure. When you combine a knowledge graph with specific data in a graph database the combination is very powerful.

## Semantic Technology

Semantic Technology is the open standards approach to knowledge graphs and graph databases. (Google, Facebook, LinkedIn and Twitter all started with open source approaches, but have built their own proprietary versions of these technologies.) For most firms we recommend going with open standards. There are many open source and vendor supported products at every level of the stack, and a great deal of accumulated knowledge as to how to solve problems with these technologies.

Semantic technologies implement an alphabet soup of standards, including: RDF, RDFS, OWL, SPARQL, SHACL, R2RML, JSON-LD, and PROV-O. If you're unfamiliar with these it sounds like a bunch of techno-babble. The rap against semantic technology has been that it is complicated. It is, especially if you have to embrace and understand it all at once. But we have been using this technology for almost 20 years and have figured out how to help people adapt by using carefully curated subsets of each of the standards and leading through example to drastically reduce the learning curve.

While there is still some residual complexity, we think it is well worth the investment in time. The semantic technologies stack has solved a large number of problems that graph databases and knowledge graphs have to solve on their own, on a piecemeal basis. Some of these capabilities are:

- Schema – graph databases and even knowledge graphs have no standard schema, and if you wish to introduce one you have to implement the capability yourself. The semantic technologies have a very rich schema language that allows you to define classes based on what they mean in the real world. We have found that disciplined use of this formal schema language creates enterprise models that are understandable, simple, and yet cover all the requisite detail.

- Global Identifiers – semantic technology uses URIs (the Unicode version of which is called an IRI) to identify all nodes and arcs. A URI looks a lot like a URL, and best practice is to build them based on a domain name you own. It is these global identifiers that allow the graphs to “self-assemble” (there is no writing of joins in semantic technology, the data is already joined by the system).
- Identity Management – semantic technology has several approaches that make living with the fact that you have assigned multiple identifiers to the same person or product or place. One of the main ones is called “sameAs” and allows the system to know that ‘n’ different URIs (which were produced from data in ‘n’ different systems, with ‘n’ different local IDs) all represent the same real-world item, and all information attached to any of those URIs is available to all consumers of the data (subject to security, of course).
- Resource Resolution – some systems have globally unique identifiers (you’ve seen those 48-character strings of numbers and letters that come with software licenses, and the like), but these are not very useful, unless you have a special means for finding out what any of them are or mean. Because semantic technology best practice says to base your URIs on a domain name that you own, you have the option for providing a means for people to find out what the URI “means” and what it is connected to.
- Inference – with semantic technology you do not have to express everything explicitly as you do in traditional systems. There is a great deal of information that can be inferred based on the formal definitions in the knowledge graph as part of the semantic schema and combined with the detailed data assertions.
- Constraint Management – most graph databases and knowledge graphs were not built for online interactive end user update access. Because of their flexibility it is hard to enforce integrity management. Semantic technology has a model driven constraint manager that can ensure the integrity of a database is maintained.

- Provenance – one key use case in semantic technology is combining data from many different sources. This creates a new requirement when looking at data that has come from many sources you often need to know: Where did this particular bit of data come from? Semantic Technologies have solved this in a general way that can go down to individual data assertions.
- Relational and Big Data Integration – you won't be storing all of your data in a graph database (semantic, or otherwise). Often you will want to combine data in your graph with data in your existing systems. Semantic technology has provided standards, and there are vendors that have implemented these standards, such that you can write a query that combines information in the graph with that in a relational database or a big data store.

It is hard to cover a topic as broad as this in a page, but hopefully this establishes some of what the approach provides.

## Applying Graph Technology

So how do these technologies deliver capability to some more common business problems?

### Customer 360

We worked with a bank that was migrating to the cloud. As part of the migration they wanted to unify their view of their customers. They brought together a task force from all the divisions to create a single definition of a customer. This was essentially an impossible task. For some divisions (Investment Banking) a customer was a company, for others (Credit Card processing) it was usually a person. Not only were there differences in type, all the data that they wanted and were required to have in these different contexts was different. Further one group (corporate) espoused a very broad definition of customer that included anyone that they could potentially contact. Needless to say, the “Know Your Customer” group couldn't abide this definition as every new customer obligates them to perform a prescribed set of activities.

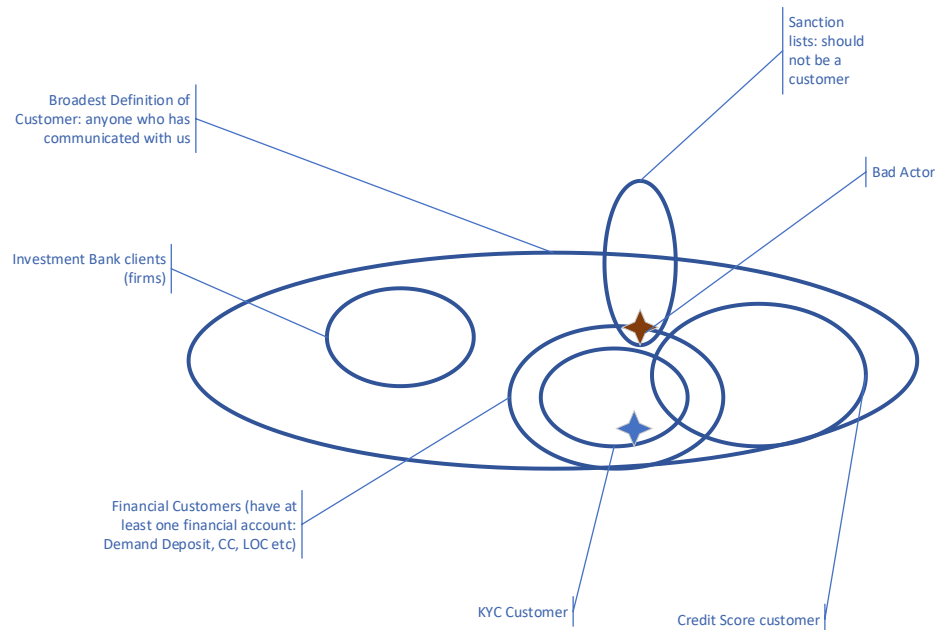
What we have discovered time and again is that if you start with a term (say, “Customer”) and try to define it, you will be deeply disappointed. On the other hand, if you start with formal definitions (one of which for “Customer” might be, “a Person who is an owner or beneficiary on a financial account” (and of course financial account has to be formally



defined)), it is not hard to get agreement on what the concept means and what the set of people in this case would be. From there it is not hard to get to an agreed name for each concept.

In this case we ended up creating a set of formal, semantic definitions for all the customer related concepts. At first blush it might sound like we had just capitulated to letting everyone have their own definition of what a “Customer” was. While there are multiple definitions of “Customer” in the model, they are completely integrated in a way that any individual could be automatically categorized and simultaneously in multiple definitions of “Customer” (which is usually the case).

The picture shown below, which mercifully omits a lot of the implementation detail, captures the essence of the idea. Each oval represents a definition of “Customer.”



In the lower right is the set of people who have signed up for a free credit rating service. These are people who have an “Account’ (the credit reporting account), but it is an account without financial obligation (there is no balance, you cannot draw against it, etc.). The Know Your Customer (KYC) requirements only kick in for people with Financial Accounts. The overlap suggests some people have financial accounts and non-financial accounts. The blue star represents a financial customer that also falls under the guidelines of KYC. Finally, the tall oval at the top represents the set of people and organizations that are not to be customers, the so-called “Sanctions lists.” You might

think that these two ovals should not overlap, but with the sanctions continually changing and our knowledge of customer relations constantly changing, it is quite possible that we discover after the fact that a current customer is on the sanctions list. We've represented this as a brown star that is simultaneously a financial customer and someone who should not be a customer.

We think this approach uniquely deals with the complexity inherent in large companies' relationships with their customers.

In another engagement we used a similar approach to find customers who were also vendors, which is often of interest, and typically hard to detect consistently.

## **Compliance**

Compliance also is a natural for solving with Knowledge Graphs.

### ***Next Angles***

Mphasis' project "Next Angles" converts regulatory text into triples conforming to an ontology, which they can then use to evaluate particular situations (we've worked with them in the past on a semantic project). In this white paper they outline how it has been used to streamline the process of detecting money laundering: <http://ceur-ws.org/Vol-1963/paper498.pdf>.

### ***Legal and Regulatory Information Provider***

Another similar project that we worked on was with a major provider of legal and regulatory information. The firm ingests several million documents a day, mostly court proceedings but also all changes to laws and regulation. For many years these documents were tagged by a combination of scripts and off shore human taggers. Gradually the relevance and accuracy of their tagging began to fall behind that of their rivals.

They employed us to help them develop an ontology and knowledge graph; they employed the firm netOWL to perform the computational linguistics to extract data from documents and conform it to the ontology. We have heard from third parties that the relevance of their concept-based search is now considerably ahead of their competitors.

They recently contacted us as they are beginning work on a next generation system, one that takes this base question to the next level: Is it possible to infer new information in search by leveraging the knowledge graph they have plus a deeper modeling of meaning?

### ***Investment Bank***

We are working in the Legal and Compliance Division for a major investment bank. Our initial remit was to help with compliance to records retention laws. There is complexity at both ends of this domain. On one end there are hundreds of jurisdictions promulgating and changing laws and regulations continually. On the other end are the billions of documents and databases that must be classified consistently before they can be managed properly.

We built a knowledge graph that captured all the contextual information surrounding a document or repository. This included who authored it, who put it there, what department were they in, what cost code they charged, etc., etc. Each bit of this contextual data had textual data available. We were able to add some simple natural language processing that allowed them to accurately classify about 25% of the data under management. While 25% is hardly a complete solution, this compares to ½ of 1% that had been classified correctly up to that point. Starting from this they have launched a project with more sophisticated NLP and Machine Learning to create an end user “classification wizard” that can be used by all repository managers.

We have moved on to other related compliance issues, which includes managing legal holds, operation risk, and a more comprehensive approach to all compliance.

## **Summary**

Knowledge Graphs and Semantic Technology are the preferred approach to complex business problems, especially those that require the deep integration of information that was previously hard to align, such as customer-related and compliance-related data.

11 Old Town Square  
Suite 200  
Fort Collins, CO 80524

970-490-2224  
305-425-2224  
info@semanticarts.com

© Semantic Arts, Inc.