

Knowledge Graphs: A Love Affair

Posted by **Sam Chance** on Mar 12, 2021 9:24:43 AM

Find me on:

Share

This article describes the concept of the knowledge graph, how knowledge graphs support data interoperability and some of their benefits. The reader will gain an understanding of what a knowledge graph is, its role in data interoperability, and be more prepared to understand their role in data integration and access.

A Knowledge Graph is a connected graph of data and associated metadata applied to model, integrate and access an organization's information assets. The knowledge graph represents real-world entities, facts, concepts, and events as well as the relationships between them. Knowledge graphs yield a more accurate and comprehensive representation of an organization's data.

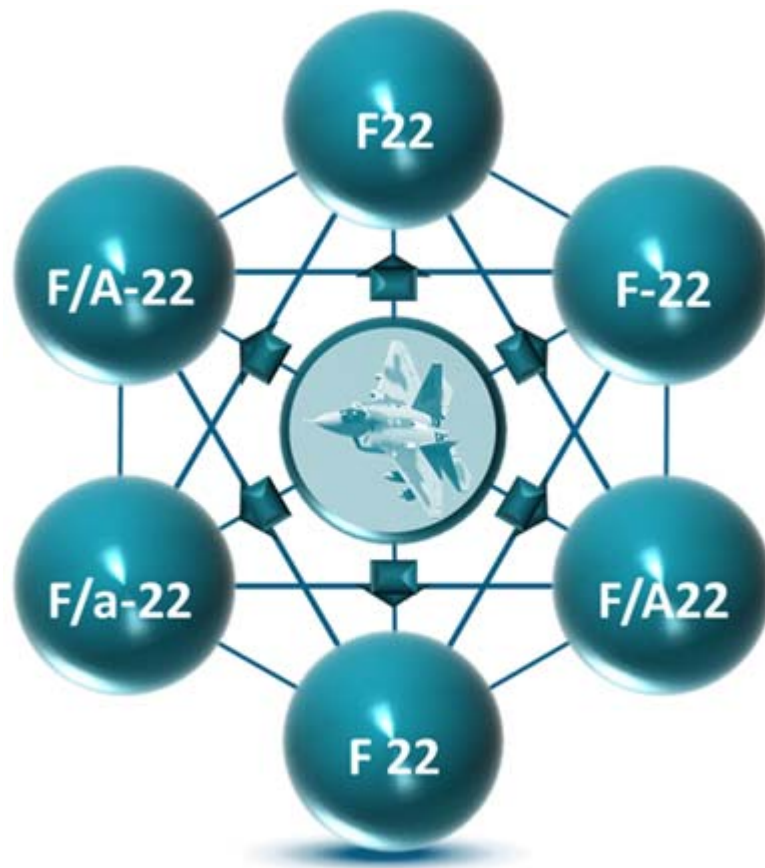


Data rich organizations continually seek to improve their analytics and decision making capabilities. This unending quest has produced a consensus view that *data* is a strategic asset — the key to unlocking insights and to making timely and informed decisions lies in data. For example, “data-driven” decision making has become essential to many organizations. Unfortunately, data reside in myriad formats, structures and syntaxes. This requires organizations to allocate significant resources to data integration activities which impede new insights and superior decision making. For example, when a user needs to ask new questions, IT must discover, select and prepare the data and create a new application to support asking the new questions. This time consuming and manually intensive process is so common, it's considered “normal.”

To improve analysis and decision support, contemporary data management architectures often employ Data Warehousing (DW), Master Data Management (MDM), and Data Lakes to integrate data. But, these solutions produce mixed results and fall short of enabling on demand access to information that is ready for consumption — much less for unanticipated questions.

The Data Lake, a fairly recent advance in data architecture, is a central repository for all types of raw data, whether structured or unstructured, from multiple sources. Nevertheless, collocating data does little for data integration. Organizations continue to dedicate significant resources to data preparation, custom coding and tool stitching for each analytic use case.

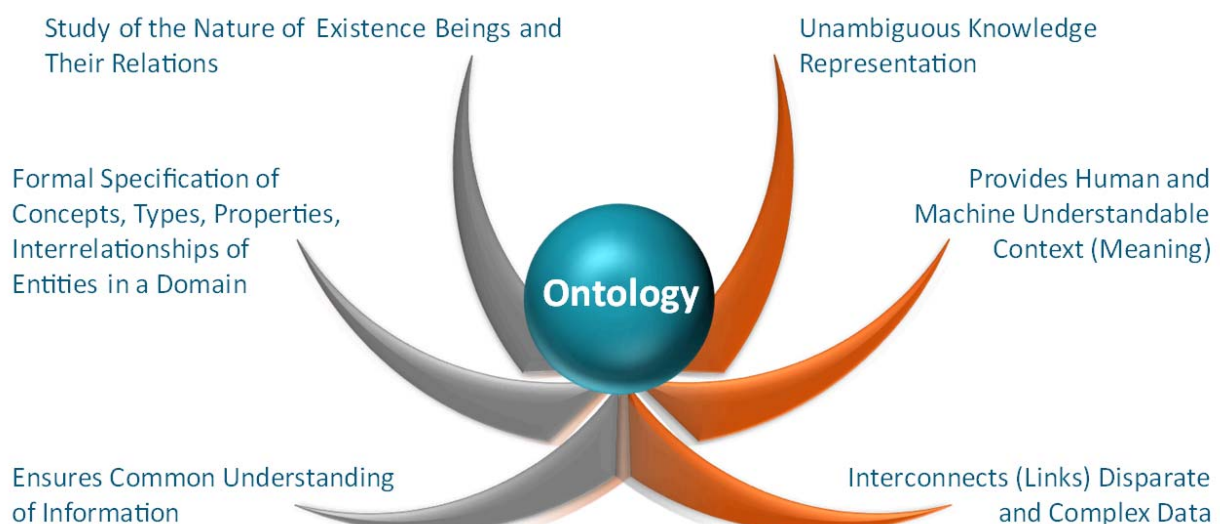
Common to these data integration approaches, only syntactic and structural integration occurs, at best. Data *semantics*, which represent the “meaning” of entities and their important relationships, are left open to interpretation. Lack of semantic integration results in a combinatorial explosion of mediations to facilitate data interpretation and analysis — which means delay, deferred decision making and incompleteness (i.e., conjecture).



Lack of Semantic Integration Hinders Communications and Perpetuates Brittle Data Architectures.

Semantic data integration using formal ontology models represents data with unambiguous, shared meaning, which is human-readable, machine understandable, and interoperable. The key enabler of semantic interoperability is the knowledge graph. For our purposes, knowledge graphs are built on **W3C OWL** ontologies to provide unambiguous knowledge representation; and the data are normalized to **W3C RDF**. So we say, “OWL ontologies provide the ‘knowledge’ and RDF — of which OWL is an extension — provides the ‘graph,’ and together we get the ‘knowledge graph.’”

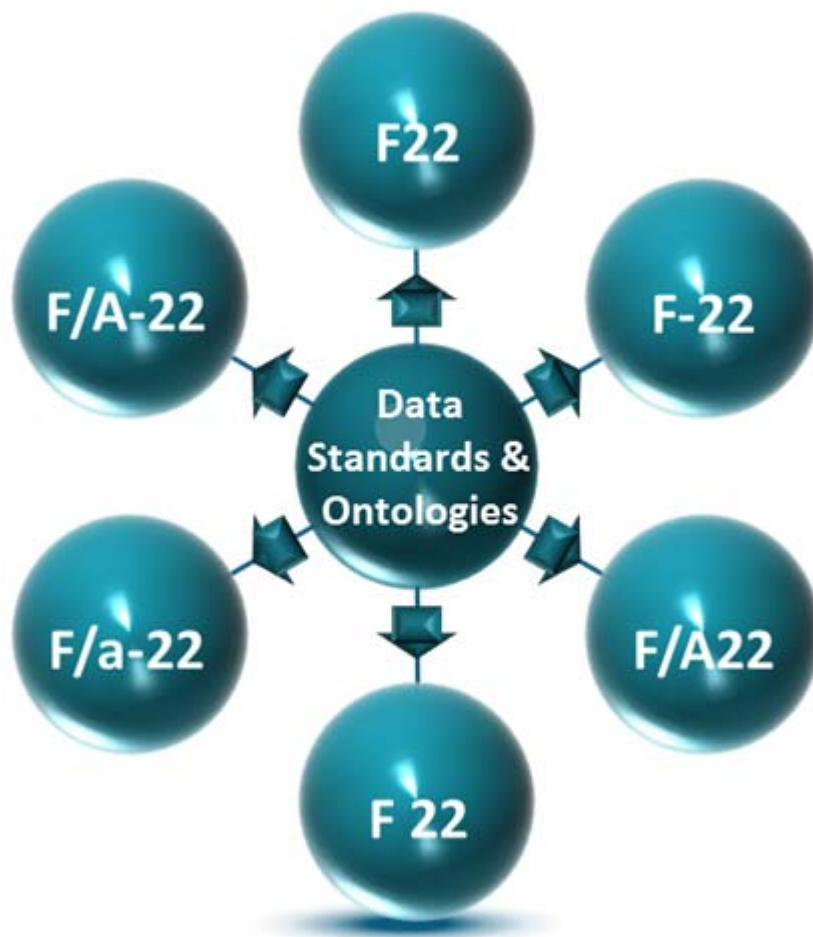
Knowledge graphs link disparate and complex data. They normalize the myriad formats and structures to the [RDF] graph, and harmonize the semantics using [OWL] ontologies. An important new capability that emerges is that knowledge graphs allow automated reasoning services, which widens the path to machine-to-machine autonomy, and operationalization of artificial intelligence (AI) initiatives. A word to the wise, however, I recommend caution when considering extensive use of reasoning services. I’ll elaborate in a future article.



Knowledge Graphs Built Using Ontology Promote Reuse, Modularity, Extensibility, Maintainability and Flexibility

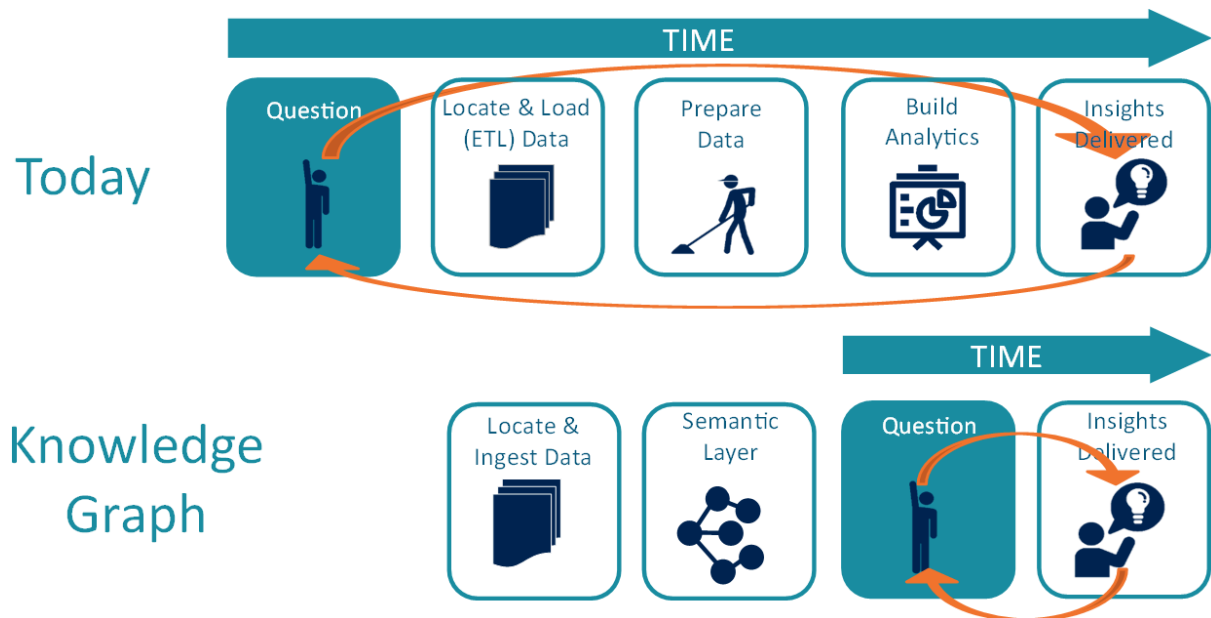
Knowledge graphs allow reuse of foundational concepts that can be applied across domains, organizations and applications. Knowledge graphs are modular which allows separation and recombination — avoiding the need to create a single, shared model (i.e., ‘one ring to rule them all.’). Knowledge graphs can be linked, potentially across a community of interest or ecosystem. Extensibility of knowledge graphs allows growth for new applications. Maintainability of knowledge graphs facilitates the process of identifying and correcting defects, accommodates new requirements, and copes with changes in domain semantics. Knowledge graphs are flexible to changes in underlying technologies and enable separation of design and implementation concerns.

Knowledge graph-based data systems, such as a “**Data Fabric**,” assume one “never has all the facts.” This assumption results in unprecedented flexibility and adaptability. For example, in practice, models *do change*; we learn new information that changes our understanding of some area of interest. Changing knowledge models is easy, and we can incorporate new data sources on the fly. An important outcome is users can ask unanticipated questions, something that is not feasible with traditional data architectures. Semantics enable on-demand data marts in ways traditional data integration technologies simply have failed to do. Importantly, Knowledge graphs naturally cope with the “same concept, different terms” dilemma, which is an inherent characteristic of virtually every data-rich environment.



Knowledge graphs naturally cope with “same concept, different terms,” an inherent characteristic of data ecosystems.

Knowledge graphs are self-describing and schema-less: data describes itself in a uniform machine-processable form. Knowledge graphs are intuitive to users, managers, and other stakeholders. Users can ask ad hoc questions of the entire knowledge graph. The “gap” between technology and business users narrows and time from idea to answer (decision cycle) shrinks.



Creating the semantic layer using knowledge graphs compresses the decision cycle and enables ad hoc questions.

Knowledge graphs create a “semantic layer” of contextualized, harmonized and meaningful data that draws and abstracts from systems, sources and silos to empower users with immediate and interactive access to data from all required sources. Knowledge graphs enable customers to adapt more readily to evolving industry standards, regulatory requirements, and new requirements. Knowledge graphs can retain provenance of ingested data, an indispensable capability for validating critical decisions relying on integrated data.

Knowledge graphs using open standards are portable across data storage and management systems. In other words, the graphs interoperate across graph stores and analytics. Portability reduces vendor lock-in and increases flexibility.

In today’s global market, effectiveness is strongly dictated by the ability to collaborate with other enterprises and to cope with change. Conventional solutions, such as Data Warehouses, Master Data Management, and Data Lakes, insufficiently support current and new demands, exposing the need for more adaptable and decentralized approaches.

Knowledge graphs feature unsurpassed flexibility, enable common understanding, and are a powerful means to achieve seamless interoperability — within the boundaries of an enterprise and across ecosystems. Use of knowledge graphs is essential for interoperability to improve communication and cope gracefully with complexity and volatility.

Cambridge Semantics’ Anzo represents a great “reference implementation” of a **knowledge graph platform**. An important design aspect of Anzo, in my opinion, is that it is **built on** W3C RDF and OWL to manage W3C RDF and OWL. This makes it a “self-similar” design, which is an enterprise architecture strategic advantage that promotes consistency and coherence, and helps systems cope more effectively with complexity.