

A Brief Introduction to the gist Semantic Model

Phil Blackwood, Semantic Arts (2020)

It's no secret that most companies have silos of data and continue to create new silos. Data that has the same meaning is often represented hundreds or thousands of different ways as new data models are introduced with every new software application, resulting in a high cost of integration. By contrast, the data-centric approach starts with the common meaning of the data to address the root cause of data silos:

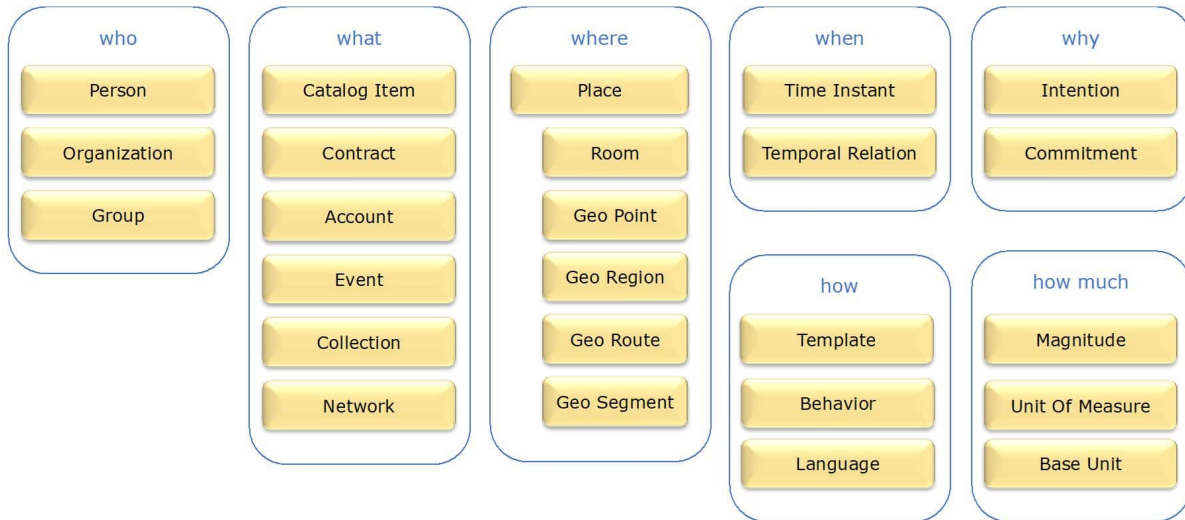
An enterprise is data-centric to the extent that all application functionality is based on a single, simple, extensible, federate-able data model.

An early step along the way to becoming data-centric is to establish a semantic model of the common concepts used across your business. This might sound like a huge undertaking, and perhaps it will be if you start from scratch. A better option is to adopt an existing core semantic model that has been designed for businesses and has a track record of success, such as gist.

[Gist is an open source semantic model created by Semantic Arts.](#) It is the result of more than a decade of refinement based on data-centric projects done with major corporations in a variety of lines of business. Semantic Arts describes gist as "... designed to have the maximum coverage of typical business ontology concepts with the fewest number of primitives and the least amount of ambiguity." The Wikipedia entry for upper ontologies compares gist to other ontologies and gives a sense of why it is a match for corporate data management.

This blog post introduces gist by examining how some of the major Classes and Properties can be used. We will not go into much detail; just enough to convey the general idea.

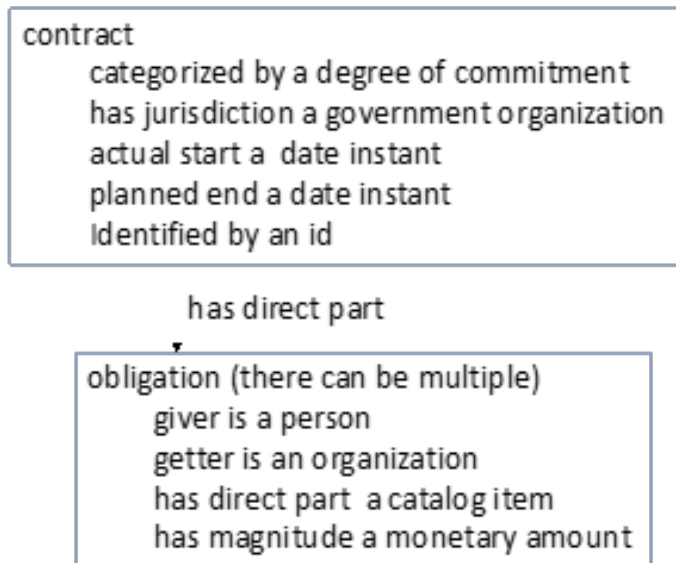
Everyone in your company would probably agree that running the business involves products, services, agreements, and events like payments and deliveries. In turn, agreements and events involve "who, what, where, when, and why", all of which are included in the gist model. Gist includes about 150 Classes (types of things), and different parts of the business can often be modeled by adding sub-classes. Here are a few of the major Classes in gist:



gist also includes about 100 standard ways things can be related to each other (Object Properties), such as:

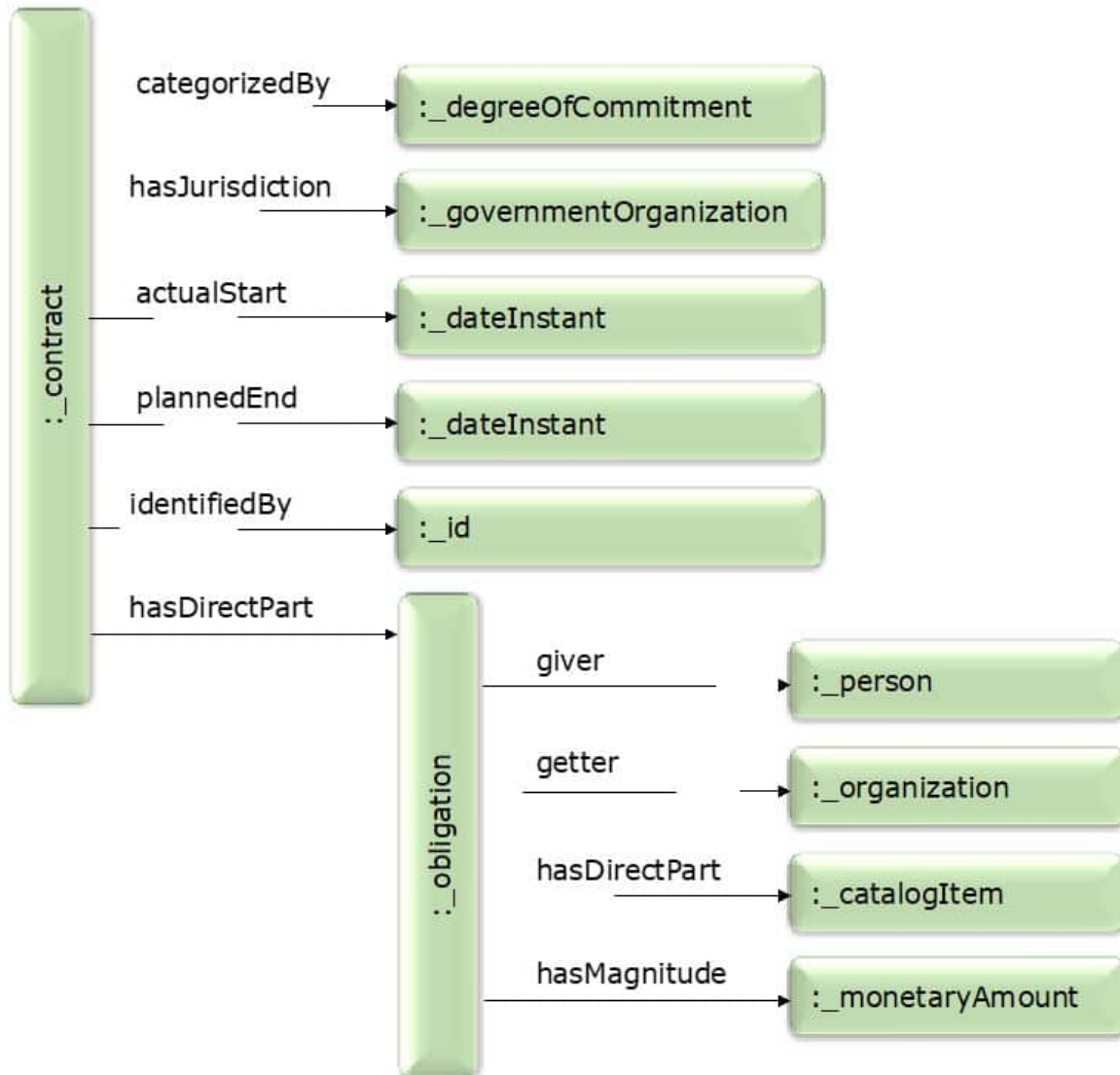
- owns
- produces
- governs
- requires, prevents, or allows
- based on
- categorized by
- part of
- triggered by
- occurs at (some place)
- start time, end time
- has physical location
- has party (e.g. party to an agreement)

For example, the data representing a contract between a person and your company might include things like:



In gist, a Contract is a legally binding Agreement, and an Agreement is a Commitment involving two or more parties. It's clear and simple. It's also expressed in a way that is machine-readable to support automated inferences, Machine Learning, and Artificial Intelligence.

The items and relationships of the contract can be loaded into a knowledge graph, where each “thing” is a node and each relationship is an edge. Existing data can be mapped to this standard representation to make it possible to view all of your contracts through a single lens of terminology. The knowledge graph for an individual contract as sketched out above would look like:



Note that this example is just a starting point. In practice, every node in the diagram would have additional properties (arrows out) providing more detail. For example, the ID would link to a text string and to the party that allocated the ID (e.g. the state government that allocated a driver’s license ID). The CatalogItem would be a detailed Product or Service Specification. In the knowledge graph, there would be a single Person entry representing a given individual, and if two entries were later discovered to represent the same person, they could be linked with a `sameAs` relationship.

Relationships in gist (Properties) are first class citizens that have a meaning independent of the things they link, making them highly re-usable. For example, `identifiedBy` is not limited to contracts, but can be used anywhere, something has an ID. Note that the Properties in gist are used to define relationships between instances rather than Classes; there are also a few standard relationships between Classes such as `subClassOf` and `equivalentTo`.

The `categorizedBy` relationship is a powerful one, because it allows the meaning of an item to be specified by linking to a taxonomy rather than by creating new Classes. This pattern contributes to extensibility; adding new characteristics becomes comparable to adding valid values to a relational semantic model instead of adding new attributes.

Unlike traditional data models, the gist semantic model can be loaded into a knowledge graph and then the data is loaded into the same knowledge graph as an extension to the model. There is no separation between conceptual, logical, and physical models. Similar queries can be used to discover the model or to view the data.

Gist uses the W3C OWL standard (Web Ontology Language), and you will need to understand OWL to get the most value out of gist. To get started with OWL for corporate data management, check out the book [Demystifying OWL for the Enterprise](#), by Michael Uschold. There's also a brief introduction to OWL and the way it uses set theory [here](#).

The technology stack that supports OWL is well-established and has minimal vendor lock-in because of the simple standard data representation. A semantic model created in one knowledge graph (triple store) can generally be ported to another tool without too much trouble.

To explore gist in more detail, you can download an ontology editor such as Protégé and then select File > Open From URL and enter: <https://ontologies.semanticarts.com/o/gistCore9.4.0> Once you have the gist model loaded, select Entities and then review the descriptions of Classes, Object Properties (relationships between things), and Data Properties (which point to string or numeric values with no additional properties). If you want to investigate gist in an orderly sequence, I'd suggest viewing items in groups of "who, what, when, where, and how."

Take a look at gist. It's worth your time, because having a standard set of common terms like gist is a significant step toward reversing the trend toward more and more expensive data silos.