

The Case for Enterprise Ontology

Dave McComb, Semantic Arts, March 2025

I was asked by one of our senior staff why someone might want an enterprise ontology. From my perspective, there are three main categories of value for integrating all your enterprise's data into a single core:

- Economy
- Cross Domain Use Cases
- Serendipity

Economy

For many of our clients there is an opportunity that stems from simple rationalization and elimination of duplication. Every replicated data set incurs costs. It incurs costs in the creation and maintenance of the processes that generate it. But the far bigger costs are associated with data reconciliation. Inevitably each extract and population create variation. These variations add up, triggering additional research to find out why there are slight differences between these datasets.

Even with ontological based systems, these difference creep in. We know that many of our clients ontological based domains contain an inventory (or a sub inventory).

Employees are a good example. These sub-directories show up all over the place. There is a very good chance each domain has their own feed from HR. They may be fed from the same system, but as is often the case, each was directed to a warehouse or a different system for their source. Even if they came from the same source - the pipeline, IRI assignment and transformation are all likely different.

Here's an illustration from a large bank associated with records retention within their legal department. One part of this project involved getting a full directory of all the employees into the graph. Later on we were working with another group on the technical infrastructure, and they wanted to get their own feed from HR to convert into triples. Fortunately we were able to divert them by pointing out that there was already a feed that provided curated employee triples.

They accepted our justification but asked ... "can we have a copy of those triples to conform to our needs." This gave us the opportunity to explain there is no conforming. Each triple is an individual asserted fact with its own provenance. You either accept it or ignore it. There really isn't anything to conform. There is no need to restructure.

At first glance all their sub domains seemed to stand alone, but the truth is there is a surprising amount of overlap between them. There were many similar but not identical definitions of “business units.” There were several incompatible ways to describe geographic aggregation. Many different divisions dealt with the same counterparties or with the same products. And it is only when the domains are unified that most of these differences come to light.

Just unifying and integrating duplicate data sets provided economic justification for the project. We know of another company that justified their whole graph undertaking simply from the rationalization and reduction of subscriptions to the same or similar datasets from different parts of the business.

The good news is that harmonizing ontologically based systems is an order of magnitude cheaper than traditional systems.

Cross Domain Use Cases

Reuse of concepts is one of the most compelling reasons for an enterprise ontology. Some of the obvious cross-domain use cases from some of our pharmaceutical clients include:

- Translation of manufacturing process from bench to trial to full scale
- Integration of Real-World Evidence and Adverse events
- Collapsing submission time for regulatory reporting
- Clinical trial recruiting
- Cross channel customer integration

Some of the best opportunities come from combining previously separate sub domains. Sometimes you can know this going into a project. But sometimes you don’t discover the opportunity until you are well into the project. Those are the ones that fall into the serendipity category.

Serendipity

I’ve recently come to the realization that the most important use cases for unification might in fact be serendipity. That is, the power might be in unanticipated use cases. I’ll give some examples and then we’ll point you to a video from one of Amazon’s lead ontologists who came to the same conclusion.

Schneider-Electric

We did a project for Schneider-Electric ([see case study](#)). We constructed the scaffolding of their enterprise ontology and then drilled in on their product catalog and

offering. Our initial goal was to get their 1 million parts into a knowledge graph and demonstrate that it was as complete and as detailed as their incumbent system. At the end of the project we had all their products in a knowledge graph, with all their physical, electrical, thermal and many other characteristics defined and classified.

Serendipity 1: Inherent Product Compatibility

We interviewed product designers to find out the nature of product compatibility. It was easy to write a different type of rule (using SPARQL) with our greatly simplified ontology that persisted the “inherent” compatibility of parts into the catalog. By doing this it reversed the sequence of events. Previously, because the compatibility process was difficult and time-consuming, they would wait until they were ready to sell a line of products in a new market before beginning the compatibility studies. Not knowing the compatibility added months into their time-to-market. In the new approach, the graph knew which products were compatible before the decision to offer them to new markets.

Serendipity 2: Standards Alignment

Schneider were interested in aligning their product offerings with the standard called eCl@ss which has over 15,000 classes and thousands of attributes. It is a complex mapping process, which had been attempted before but abandoned. By starting with the extreme simplification of the ontology (46 classes and 36 properties out of the several hundred in the enterprise ontology), working toward the standard was far easier and we had an initial map completed in about two months.

Serendipity 3: Integrating Acquisitions

Schneider had acquired another electrical part manufacturer, Clipsal. They asked if we could integrate the Clipsal catalogue with the new graph catalogue. Clipsal also had a complex product catalogue. It was not as complex as Schneider’s, but it was complex and structured quite differently.

Rather than reverse engineering the Clipsal catalogue we just asked for their data engineers to point us to where the 46 classes and 36 properties were in the catalogue. Once we’d extracted all that we asked if we were missing anything. Turns out there were a few items, which we added to the model.

The whole exercise took about six weeks. At the end of the project we were reviewing the Schneider-Electric page in Wikipedia and found that they had acquired Clipsal over ten years prior. When we asked why they hadn’t integrated their catalogue in all the time they responded that it was “too hard.”

All three of these use cases are of interest, because they weren't the use cases we were hired to solve but only manifested when the data was integrated into a simple model.

Amazon Story of Serendipity

This video of Ora Lassila is excellent and inspiring.

https://videlectures.net/videos/iswc2024_lassila_web_and_ai

If you don't have time to watch to the whole thing, skip into minute 14:40 where he describes the "inventory graph" for tracking packages in the Amazon ecosystem. They have 1 Trillion triples in the graph and the query response is far better than it was in their previous systems. At minute 23:20 he makes the case for serendipity.