Semantics Driven Architecture™ (SDA™)

1. The Integration Imperative

Since almost the beginning of the computing world, the hardest problem has always been *integration*. The first programs were self-contained, small, and well-defined; they were typically descriptions of mathematical algorithms, cast into the new language of computing (for the first decades, assembly language, specific to specific computing architectures). As soon as those programs were written, the first legacy systems were in place, and the first integration problems appeared.

Integration of existing application code is critical to the construction of large software systems, the same way that *systems engineering* is about the integration of engineering disciplines; it makes more sense, in the abstract, to paste together existing codes to make larger ones than to "start from scratch." That requires, however, the full understanding of the codes that already exist, or at least the *interfaces* at the edges of those codes and the meaning of those interfaces. Building the world's best password vault is useless if programmers that need to use that vault can't understand how to unlock it.

2. The N+1 Problem

That brings us to the thorniest problem in engineering, a problem we call the "N+1" problem. Any well-trained engineer knows to simplify systems, to make them easier to understand, easier to maintain and easier to integrate by minimizing those "edges" (the interfaces). An obvious cognate in the non-software world is electrical plus ("mains plugs"). As any traveler knows, there are about seven most common designs for connecting electrical equipment to the wall to receive electricity; while most of Europe and many other countries use two round pegs (of different diameters!), North America and Japan and a few other countries use two parallel, flat blades. Meanwhile most countries in the British Commonwealth use some variant of the large British connector, Australia relies on non-parallel flat blades, and most importantly, none of these are interchangeable.

Any good engineer looks at this problem and can propose a single design that will work everywhere, with Japan's 100 volts at 50 or 60 Hertz, with Europe's 220 volts at 50 Hertz, with Canada's 110 volts at 60 Hertz, and be completely useable everywhere. Thus, theoretically, the world would be able to move from about seven primary electrical connectors to one! But of course none of the many billion existing connectors is likely to be replaced, either on the walls or on the devices, so the net effect of this effort would be not to reduce the number of connectors to one, but to increase the number to eight -- N+1.

More importantly, the exercise is pointless, because the world has dealt with this problem successfully already, with *integration devices* that we call *converters* or *adapters.* Since most electrical equipment today is happy to take whatever voltage is coming out of the wall, the only remaining problem is the conversion of the physical connector; we can deal with that with a US$1 device available at any hardware store (or US$20 at any airport!) that converts the connector on the wall to the connector on the device.

We can think of this device as a *translator*, translating one physical connection to another. In the context of the software integration problem, it's about connecting from one edge, one design, *one meaning*, to another.

3. Integrating Systems: Sharing Concepts

This in essence is a simple version of the problem we face when we connect legacy systems; and in fact every system ever built and used is a legacy system that at some point has to be integrated with other systems. In order to make that possible, those legacies must be understood. They've been built to different requirements, with different intended purposes, different architectures and most importantly typically by different designers, and those different designers had different *semantics* in mind.

That is to say, what I mean by *body temperature* might be different than what you mean. This comes not only from the simple issue of different temperature scales (Celsius vs. Fahrenheit), but whether that's a temperature taken orally or axillary, or whether it's an internal core body temperature or an externally-measured temperature, is critically important to understanding whether the body is functioning properly or not. That *semantics* (the specific meaning, in this case, of "body temperature") must be shared by the person capturing and recording the temperature, and by the person deciding on my health status. Getting it wrong can have grave consequences.

The Object Management Group® (OMG®) has approached this problem for more than twenty years with an approach called *Model Driven Architecture®*, an approach that focused on capturing as many details as possible about the "edges" between systems--relationships, parameters, interfaces. But we need more; we need to share the meanings of words, and not just the words.

4. What Do We Mean by Semantics?

So then what do we mean by semantics and what will semantics-driven architecture standards deliver that we don't have now?
Many people "think they know" what the term "semantics" means, but as it turns out, we don't always agree or understand one another. Philosophers talk about the study of

meaning; in linguistics the definition is similar, but implies a formal representation of meaning, using formal logic (e.g., situation semantics), including formal definitions, well-defined relationships and constraints between elements of an expression (words and phrases), mapping rules with respect to binding pronouns, ellipses, and other components of an expression, and so forth. In computer science, people sometimes use the term "semantics" to describe integrity constraints, variable bindings, and other programming and data modeling concepts, while others assume that by semantics one means a formal representation of knowledge that is both human and machine interpretable. The latter understanding is derived from joint work between computer scientists, linguists, philosophers, and psychologists that has gone on for the last 30 years or more, although each discipline wears their own color of glasses when talking about it.

So what is the real business case for semantic integration? Here's an example from the healthcare domain. Healthcare delivery is often referred to anecdotally as "a team sport" in part because so many varied specialties and subspecialties collaborate by necessity as part of routine care delivery. For example, a patient presenting for stomach pain will often require a physical examination, laboratory work, review of recent medical events (lab results), analysis of current medications, personal history, and perhaps a host of other items. It is rare that all of this occurs within one setting, and information sharing challenges exacerbate as each niche community uses its own "native" language for documenting and expressing their findings. Some simple examples:

- A blood pressure reading (120/80) has entirely different meanings if taken with the patient at rest or during exercise
- Laboratory test values may appear to be either normal or abnormal depending upon the reference range of the instrumentation used
- "MI" to a registration clerk in the US means Michigan, but means Myocardial Infarction (heart attack) to a physician.
- When scheduling a medical appointment, there is no shared vernacular to describe visit types

The list can go on and on. Elemental to achieving interoperability is establishment and accurate conveyance of a shared meaning across organizations (or parts of an organization), and the systems that support them. This shared meaning, however, is only one step en route to the real objective of being able to have systems intelligently act on that data.

For example, a human caregiver can review a medication list and equate "penicillin" with "PCN" with "pclln" with "amox" with "axomicillin". Within the context of a current set of medications, these are easily enough human-interpretable. To make this information machine actionable, however, requires more consistent representation, and the ability to "compute on" that information to a desired outcome – for example, checking patient allergies for penicillin class drugs to prevent a harmful drug order. What is needed to make this step a reality is consistent data representation, modeling of interrelationship among concepts (for instance, relationships between drugs and allergic reactions), and the standards infrastructure to allow for the expression, navigation, interaction, and query among this complex knowledge landscape.

That of course is just the beginning. The healthcare industry has to interact closely with transportation and logistics, with workplace management, with efficient building management, and a dozen other disciplines. Imagine the semantic integration problem that results!

5. OMG's Areas of Expertise

At this writing, the OMG is almost three decades old, with a history of developing shared software designs--standards--employed in literally billions of running computing systems. Today's OMG focuses on four general areas:

- the Industrial Internet of Things: leveraging OMG's long experience in middleware and modeling, the application of the Internet of Things (IoT) to industrial systems, whether manufacturing, healthcare, electrical grids or other industrial-scale systems that must *integrate* literally millions of data to know how to optimize or deliver better service. More than middleware, it's about *semantics in vertical markets;* how do we build and integrate tools and devices from multiple manufacturers and users in large-scale systems?
- the System Modeling Environment (SME): leveraging OMG's traditional modeling language experience to create not only shared languages (like UML® and the upcoming SysML® 2), but shared systems models defined in those languages, it's about *models in vertical markets*. Shared, reusable satellite engineering designs, for example, will make it easier to build and launch satellites for purposes from Earth surveillance to communications.
- Sovereignty & Security, a burgeoning area leveraging OMG's strengths in security standards and languages, is rapidly being recognized as a leading problem in computing. Managing *cybersecurity and data residency in vertical markets* must be solved in a world in which nearly every device is software-driven.
- Business Architecture & Modeling will take advantage of OMG's existing standards and relationships in the business modeling space to build and collect not only shared languages like BPMN™, SBVR™ and BMM™ but also *capability models in vertical markets*; that is, shared expressions of business capabilities like "shipping" and "invoicing" that can easily be used and reused in many contexts.

All of these four areas depend on *shared semantics*; and fortunately, that's also a traditional area of expertise for the OMG.

Since the year 2000, OMG's Ontology Platform Special Interest Group (PSIG), Business Modeling and Integration (BMI) Task Force, and other working groups have developed a scaffolding of standards and technologies in the "semantics space." Some of these, such as the Semantics of Business Vocabulary and Business Rules™ (SBVR) specification, are focused primarily on modeling the terminology and linguistics aspects of semantics, while others, such as the Ontology Definition

Metamodel™ (ODM™), Distributed Ontology, Modeling, and Specification Language™ (DOL™), Decision Model and Notation™ (DMN™), and Production Rule Representation™ (PRR™) are designed to support formal knowledge representation and reasoning applications. At the same time, we have also developed content specifications, such as the Common Terminology Service™ (CTS2™), Date Time Vocabulary™ (DTV™), Financial Instrument Global Identifier® (FIGI®) specification, the Financial Industry Business Ontology®* (FIBO®*) family of ontologies, and the Information Exchange Packaging Policy Vocabulary™ (IEPPV™), and have started to develop or revise others. These content specifications define shared semantics, i.e., formal vocabularies, relationships, and rules, using formal knowledge representation methods, to support both human and machine understanding for interoperability, integration, governance, question answering, recommendation systems, and other semantically driven capabilities.

Despite this growing foundation, we've identified a number of gaps with respect to infrastructure standards, at OMG and beyond, in addition to the need for more content that people can agree and depend on. Model-driven software generation and programming-level interfaces form the basis for many long-lived standards. What we are talking about now, however, is interoperability at a higher level of abstraction, enabling exchange and integration of actionable knowledge. We are already working on a handful of standards to create integrated, semantically savvy fabrics to enable data scientists, for example, to integrate best of breed applications and services to solve really tough problems using orders of magnitude more data than ever before. The data they leverage may have been developed by entirely independent groups of people, and now must be repurposed in ways that were not envisioned at the time it was created. The services needed to deliver dynamic and content-heavy solutions must be integrated on demand, without human intervention if at all possible.

These kinds of capabilities are already needed to build an integrated factory where climate changes impact production, or where people can integrate internet-enabled devices in their homes from a variety of manufacturers. They are essential for a wide range of decision support, data management, compliance, governance, and other functions that many enterprises are attempting to cobble together today. And while some facilities to integrate data and services from a single vendor are available, the standards enabling even basic interoperability across classifiers, rule engines, machine learning algorithms, predictive analytics, and other knowledge-heavy services from multiple vendors are lacking.

Creating such a rich, semantically enabled fabric of standards doesn't happen in a vacuum. The biggest breakthroughs over the last 50+ years have been cross-technology, cross-domain, and cross industry, and it takes a community of people that are the best at what they do in their individual companies and domains, coming together to brainstorm on how to cross the chasm to create them--which, by coincidence again, we happen to have at OMG.

This *semantic infrastructure* is the basis of the Semantics Driven Architecture, and the basis of everything OMG does--and frankly, the basis for integration projects worldwide, just as the Model Driven Architecture and model-based development in general swept the world since it was first introduced by OMG in 1999.

** FIBO and Financial Industry Business Ontology are registered trademarks of the EDM-C, used with permission.*