MDA Journal

David S. Frankel
Lead Standards Architect – Model Driven Systems
SAP Labs

David.Frankel@SAP.com
Mr. Frankel's SAP WebLog is:
https://www.sdn.sap.com/fr/sdn/weblogs?blog=/pub/u/55914

Author:
Model Driven Architecture:
Applying MDA to Enterprise
Computing

Industrial Convergence and Semantic Interoperability

Up to now, the software industry has managed to avoid facing up to the low level of semantic interoperability in our data processing systems. Current macro economic trends requiring more sharing of data across vertical markets threaten to push our systems to the breaking point if we do not address this shortcoming soon. This article takes a close look at this problem, and explains why there is reason to be cautiously optimistic that we can get a handle on it.

Syntactic Interoperability

In order to explain what semantic interoperability is all about, I find it useful to first explain the notion of syntactic interoperability, because syntactic interoperability is a necessary precondition for semantic interoperability. Syntactic interoperability is the ability of a set of components to understand the syntax of the communication that flows among them.

For example, consider what happens when a client sends a message to a general ledger service, requesting that a debit of $200 be posted to account number 4601. Syntactic interoperability guarantees that, when the client specifies that 200 is the dollar amount and that 4601 is the account number, the general ledger system does not misinterpret the message and think that 4601 is the dollar amount and 200 is the account number. Syntactic interoperability also makes it possible for the client and the service to agree that 256.77 is not a valid account number, because the account number must be an integer.

The computer industry has made good progress achieving syntactic interoperability. Various distributed computing protocols, such as Web Services, CORBA, and so on, make it possible for components written in different programming languages and running over different operating systems on different machines to communicate with a shared understanding of the syntax of the data that they exchange during the course of their communications.

From Simple to Complex

However, we don't always have the luxury of dealing with messages as simple as our example, containing just an account number and amount. Industry groups and standards bodies define messages containing considerably more numerous and complex data elements.

For example, in the financial services industry, an industrial strength debit transfer message contains multiple blocks of data, including a block of data usually called something like *Debtor Party*. The Debtor Party block includes multiple data elements, including one that is named something along the lines of *Identification_Issuer_Name_Proprietary.*[1] Conceptually, this data element represents the name of the issuer of a proprietary identification for a debtor party. The element incorporates a number of semantic concepts reflected by the terms *identification*, *issuer*, *name*, *proprietary*, *debtor,* and *party*. In addition to those fine-grained concepts, it incorporates the following coarser-grained concepts that combine the more granular concepts:

- Debtor party
- Proprietary identification
- Issuer name
- Identification Issuer

Thus, the data element in question is based on a combination of these semantic concepts.

Where is the Semantic Knowledge?

In current practice, software developers hard-wire their understanding of these semantic concepts into programs that construct and deconstruct a debit transfer message. This semantic knowledge is buried deep in the code and is not manifest in any machine-readable metadata.

Of course, there are multiple formats for debit transfers in use within the financial services industry. Issuing new message formats does not mean that system managers rip out and replace applications that use older formats, which is exceedingly costly. Furthermore, various domains within the finance industry deal with debits, and each of their associated standards bodies defines different message formats. Thus, there are multiple message definitions that come into play in a single financial transaction. As messages cross subsystem boundaries while moving through the financial supply chain, a debit transfer message encoded in one format often has to be translated to a message in another format. So the old problem arises: How do we map one format to another?

The task of mapping one message format to another falls to a human analyst. The formats that the analyst must map are often lengthy and complicated. Some messages have scores of complex fields. The human analyst does the best she can. But the amount of help that our computer systems provide to help her figure out what the mapping should be is quite limited.

State of the art data mapping tools display both formats on the screen and allow the analyst to graphically draw connections and write expressions to indicate what should map to what according to what rules. These tools also are good at generating translation code, once the analyst has figured out what the mapping should be and has entered the rules into the tool. But as far as these mapping tools are concerned, our example complex data element simply has an opaque name (*Identification_Issuer_Name_Proprietary*), a data type (*Text*), and a cardinality *(zero to one*). There is no metadata about the underlying semantic concepts that the mapping tool can grab onto in order to provide guidance as to what should map to what.

If you think that it should be easy for a person to work out how to map a field in one complex format to another just by visually inspecting the formats, you probably have never tried to do it. Not only is it enormously time consuming, but it also is error prone. Subtle mistakes that occur in

[1] This example is loosely based on one of the ISO 20022 payment messages.

data transformations at subsystem boundaries within the financial supply chain are common and cost the involved parties serious money. Even if our erstwhile analyst is fortunate enough to have access to good documentation of the message formats written in English or in some human language in which she is fluent, the sheer size and complexity of the formats means the process is fraught with opportunities for honest mistakes.

Semantic Interoperability

These mistakes are a consequence of a low degree of semantic interoperability. Semantic interoperability is the ability to coordinate the functioning of a set of components to coordinate their functioning based on a shared understanding of the meaning of the communications that flow among them. In our debit transfer example, the communications involve the passing of messages, as well as the transformation of those messages from one format to another. The level of semantic interoperability is based on the degree to which components at various points in the financial supply chain have a common understanding of the meaning of the data that flows through the chain.

The reason that syntactic interoperability is a necessary precondition for semantic interoperability is that it is not possible to agree on the meaning of a piece of data if the parties to a financial transaction cannot even agree on whether the data is a dollar amount, an integer. or a piece of text.

As I have pointed out, message definitions currently in use in industry today have little if any machine readable metadata that manifests the underlying semantic concepts. Therefore, whatever degree of semantic interoperability that our data processing systems possess relies on hard-wired code and on the best efforts of human data mapping analysts who must wade through the complexity largely unassisted.

The lower the degree of semantic interoperability, the more friction in the system. Studies indicate that a one percent reduction in such friction in electronic payment systems could add a few percent to global GDP.[2] Even if such projections are too optimistic by several fold, it should still be evident that quite modest gains in semantic interoperability could produce substantial returns.

Thus, in attacking this problem, we don't have to strive for fully automated semantic interoperability. Complete automation of all mapping decisions is probably not attainable, certainly not in the foreseeable future. Humans will have to be involved in these decisions. Moreover, even a goal of, say, 50 percent automation may be too high to shoot for at this time, given the technical challenges and the resulting high cost of achieving such a target.

But we can certainly do better than we are doing today, and the evidence indicates that moderate improvements will be well worth the effort. If message definitions for electronic commerce include the right kind of metadata, then each data element essentially carries a semantic map of itself that reveals, in a structured way, the underlying concepts on which the element is based. Tools can use this metadata to provide a reasonable degree of assistance to analysts trying to work out the right data mappings. Such tools can analyze the elements' internal semantic maps and suggest (not dictate) to the analyst what the proper mappings to other elements might be. This approach semi-automates the mapping. It also is helpful to the analyst to be able to read an element's internal semantic map as she carries out the non-automated aspects of the mapping process.

This approach is all the more effective to the extent that data elements' underlying semantic concepts are reused from libraries of semantic concepts that have been defined and registered by a duly constituted registration authority. The more that the same concepts start showing up in different elements' internal semantic maps, the more that tools can find probable mapping

[2] Joseph N. Bugajski, *Response to Payments RFI*, Visa International Payments Association, August 22, 2004, OMG document finance/04-08-07.

relationships. Of course, there are issues with getting different communities aligned around common concepts; that is one of the key reasons why, realistically, we expect only modest results.

Incidentally, service-oriented systems benefit as much as message-oriented systems from the semi-automation because they pass messages around too. Moreover, the electronic payment domain on which my examples focus is only one of many e-commerce domains suffering from friction that could be partially alleviated.

Technology That Can Help

The technologies exist to make limited yet useful advances in semantic interoperability. From the standpoint of technical feasibility, therefore, it is now a matter of engineering to work out how to use these technologies together in a way that scales and provides practical improvements. The UN/CEFACT Core Components (CCTS) initiative and Semantic Web technologies, along with generic modeling tools, give us the basic toolkit we need.[3]

The tougher challenge has been getting industry groups who define electronic commerce interoperability standards to recognize the semantic interoperability problem. However, progress is in the air, due to the pressure of inexorable economic forces.

Industrial Convergence

Low levels of semantic interoperability have generated substantial economic friction for some time now, but the increasing overlap among vertical markets is severely exacerbating the problem to the point where a number of industry standards groups are starting to genuinely understand that we can no longer afford to ignore the issue. It is bad enough when there is only a low degree of semantic interoperability among a single standards body's own published messages, where all of those messages target the same vertical market. It is of an order of magnitude more difficult to deal with a paucity of semantic interoperability when messages devised for different vertical markets need to work together, because it is more difficult to find analysts who understand multiple markets.

The economy is rife with examples of the breakdown of traditional borderlines among vertical markets. Here are just a few sample data points:

- Today, 33 percent of the components in an automobile are high tech components. So if you think that RosettaNet messages for high tech have no bearing on the automotive industry, think again.
- IBM recently spent $745 million to acquire Telelogic. Telelogic's model-driven development tools target the embedded systems space. The walls are coming down between the embedded systems and enterprise systems worlds, and IBM is voting that way with its wallet.
- HIIMMS, the Healthcare Information and Management Systems Society, is spearheading an initiative to put together a health information exchange in Southeast Michigan. Major stakeholders include the Big 3 automobile companies, which are feeling a particularly acute amount of pain due to health care costs. Financial institutions that are directly involved in healthcare payments are also stakeholders. This initiative could have ramifications beyond the Detroit area.

In an economy that is gradually blurring the barriers between vertical industries, the demands for interoperability increase dramatically. Keeping up with the message formats for multiple markets

[3] For those who are familiar with these technologies: Some ontology experts have expressed reservations about certain aspects of CCTS. However, there are some new developments already in motion that I believe can overcome these problems. I will say more about that in future editions of *MDA Journal*.

and understanding how they interrelate is daunting. This specter is putting semantic interoperability on industry radar screens. It is too early to declare victory, but the pain level has reached a threshold such that we who have been pushing for better semantic interoperability are getting a hearing. Quite a number of vertical market groups have embarked upon or are seriously considering semantic interoperability initiatives. Now it is time for us to execute.

Afterword

In future issues of MDA Journal, I will examine the technologies that hold out hope of modest but significant improvements in semantic interoperability, looking at their strengths and weaknesses and how best to use them. I will also describe some of the semantic interoperability initiatives in progress in industry. Additionally, I will consider the impact of semantic interoperability on Business Process Management.

David Frankel is Lead Standards Architect for Model-Driven Systems at SAP Labs. He has over 25 years of experience as a programmer, architect, and technical strategist. He is the author of the book, "Model-Driven Architecture®: Applying MDA® to Enterprise Computing." He also is lead editor of the book "The MDA Journal." He served several terms as a member of the Architecture Board of the Object Management Group (OMG), the body that manages the MDA standards, and he has co-authored a number of industry standards. Recently he has been publishing and speaking about the role of model-driven systems in enterprise SOA and has been promoting the Business Process Expert community at www.sap.bpx.com.