A BPTrends Column October 2007

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

Semantic Interoperability and Convergence in the Finance Industry

Introduction

In my previous MDA Journal (July, 2007), I outlined the problems that low levels of semantic interoperability pose for industry. In this installment, I examine an initial attempt to deal with this problem for the financial services industry.

While interoperability among software components ultimately requires technical message formats such as XML schemas with well-defined data types, the lack of interoperability at the semantic level is more important for reducing integration costs. Semantic interoperability problems cause errors when data traverses organizational and application boundaries (that is, B2B and A2A boundaries). Agreement among trading partners on the technical syntax of a message does not guarantee a common understanding of the semantics of the information being exchanged, and, in fact, can mask a lack of such understanding.

A primary focus for efforts to improve semantic interoperability is achieving a clear understanding of the meaning of information contained in messages that parties exchange across organizational and application boundaries.

ISO 20022 (UNIFI) 1.x

Multiple standards organizations in the finance sector are converging around a standard called *ISO 20022*, also known as the *UNIversal FInancial Industry message scheme (UNIFI).* UNIFI lays out a message definition methodology that attempts to address the semantic interoperability problem.

The UNIFI standard does not define financial messages. Instead, UNIFI codifies a *process* for defining such messages. Any community of interest can use the process to define messages for the electronic exchange of data and can apply to the UNIFI Registration Authority to have the

messages registered in the UNIFI Repository.

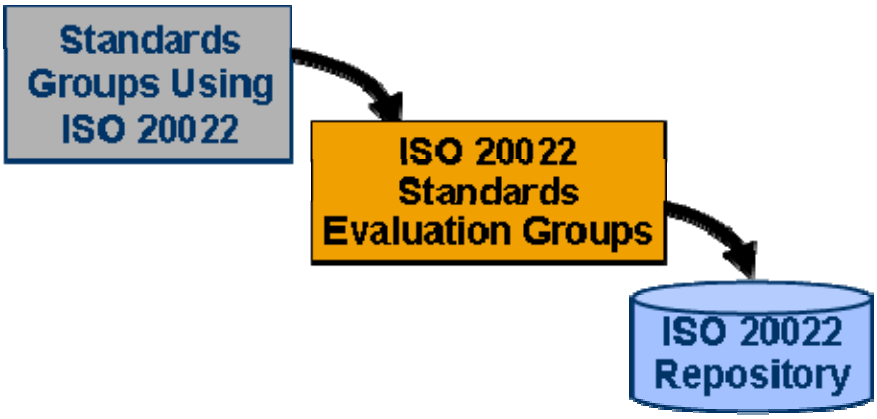

Figure 1. Standards Groups Submit Content to the ISO 20022 Repository

The UNIFI process is model-driven and component-based. Both of these characteristics flow from the desire to address the semantic interoperability problem.

Why UNIFI is Model-Driven

UNIFI's model-driven approach means that semantic specifications are carefully separated from definitions of XML wire formats. In fact, UNIFI sees XML schemas as details of interest only to the machine, and therefore it defines rules for auto-generating the schemas from the models. The models are expressed by means of a UML profile that the standard defines.

Why UNIFI is Component-Based

UNIFI's component-basis is reflected by the fact that the UML profile models a message as an assembly of reusable components that live in a common repository. The idea is that, if two schemas from different message sets appear in the same financial supply chain, and have parts that trace back to the same reusable components, then a mapping tool could suggest to an integration analyst that these parts map to each other.

Mapping tools could thus make it easier for analysts to map one UNIFI-based message to another by taking advantage of the fact that both are created with the same methodology and from common components. Mapping is very labor-intensive and thus generates enormous costs, making it a major cause of high overall integration costs. Mapping costs are so high, that even small improvements in integration analysts' mapping productivity would yield significant monetary benefits.

The models are divided into models of general business entities called *Business Components,* which are essentially business objects. A message is modeled as an assembly of *Message Components*. Message Models include trace relationships that link Message Components to the Business Components from which they are derived. This approach is supposed to ensure that there is a way to trace how an element of a message relates to the business entities that inhabit the environment in which the message is used, thus giving a smart mapping tool some more semantic information that it can use to help a mapping analyst be more productive.

Business Process Models

The UNIFI methodology also requires that message definitions submitted to the repository include business process models that define specific scenarios in which the messages are used. The methodology uses UML activity models to describe process flow, UML sequence models to describe message exchange choreographies, and UML use cases as well.

Extent of UNIFI Adoption

A number of finance industry standards organizations and other financial institutions collaborated to define UNIFI and set up the common repository. In the three years since ISO finalized the UNIFI 1.0 standard, this convergence has gained UNIFI a good deal of traction.

The first community of interest to use the methodology and to register messages in the ISO 20022 Repository was the International Standards Team Harmonization (ISTH) initiative, a consortium of standards bodies that enlisted the assistance of major banks and software vendors. ISTH defined a set of electronic payment messages called the *payment kernel* that is critical to the plans for implementing the Single European Payment Area (SEPA). Other communities have followed with additional contributions to the repository.

A list of messages already registered in the UNIFI Repository is available at http://www.iso20022.org/index.cfm?item_id=42790. A list of new messages in the pipeline for submission to the repository is available at http://www.iso20022.org/index.cfm?item_id=42936.

Message Definitions and Service Definitions

There is a great deal of discussion in the software industry about Service-Oriented Architecture. UNIFI does not explicitly cover the notion of a service. It focuses on defining messages. However, a critical part of the definition of a service is the definition of the input and output messages that are exchanged when invoking operations that are part of the service.

Limitations of UNIFI 1.x

While the separation of semantic definitions from technical formats has worked quite well, problems have arisen with the way that UNIFI 1.x structures the reusable components.

The reuse of common Message Components across multiple message sets is not materializing, because the components are generally not reusable. Each component consists of multiple data elements – it is modeled as a UML class with a number of elements (attributes and associations). In practice, that precise combination of elements seldom recurs, except possibly in closely related messages from the same message set. So each submission to the repository is defining a whole new set of components, and the size of the repository is starting to explode, filling up with UML components whose reuse potential is quite limited.

Another limitation is that the trace links from the Message Components to the Business Components are weak because they lack information that defines how the message elements are derived from the business elements. The trace links are semantic-free relationships that simply show that a message element somehow traces back to a business element.

The fact that the Message Components are not reusable and that the trace links from Message Components to Business Components are weak means that the methodology does not, in fact, bring about substantive improvements in the ability to map messages from different message sets to each other when messages from different sets show up in the same financial supply chain. This limits UNIFI's ability to cut integration costs.

Addressing UNIFI's Limitations

ISO's UNIFI expert group is drawing up plans for a UNIFI Version 2.0 that will address UNIFI 1.x's limitations. This will be the subject of the next installment of MDA Journal.

David Frankel is Lead Standards Architect for Model-Driven Systems at SAP Labs. He is a member of the panel of experts that ISO has assembled to review and update the ISO 20022

standard. 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.