# Tackling the Semantic Interoperability of Modern Manufacturing Systems

Steven R. Ray
Email: ray@nist.gov
National Institute of Standards and Technology
100 Bureau Drive, STOP 8260
Gaithersburg, MD 20899-8260

## Abstract

The need for increased precision in information standards coupled with the desire to automate parts of the system integration process has led us to the use of a formal semantic approach to systems integration. This paper provides a brief overview of some of the relevant semantics-based work underway within the Manufacturing Systems Integration Division at NIST.

## Background

The National Institute of Standards and Technology[1] supports the definition and implementation of normative standards for information exchange through development, testing and deployment activities in industrial and governmental applications. One important lesson from our experience is that ambiguity in the definition of terms within a specification leads to interoperability problems during implementation. This is a major reason that we are strong proponents of the rigor offered by semantic technology in the domain of information standards.

## Evolution of standards

Early on, information standards were in the form of protocols for moving information around, such as the ASCII[2] standard for encoding letters and symbols used by teletype machines, or even the pervasive TCP/IP[3] networking protocol standard that drives the Internet itself. These standards typically focus on the way in which information is to be encoded (the syntax) and only peripherally describe the nature of the information being standardized (the content). Traditionally, such standards are specified in terms of English prose. Specifications supporting ISO (the International Organization for Standardization) standards are typically highly structured texts intended for a human reader[4].

---

[1] For more information on NIST and its work, http://www.nist.gov/.
[2] The American Standard Code for Information Interchange. See also ISO-14962-1997 and ANSI-X3.4-1986 (R1997).
[3] TCP/IP stands for Transmission Control Protocol/Internet Protocol. For a clear and concise description of TCP/IP, see http://pclt.cis.yale.edu/pclt/COMM/TCPIP.HTM
[4] See, for example, ISO Directive, Part 3: 1997 3rd edition, "Rules for the structure and drafting of international standards" at http://www.iso.ch/iso/en/ISOOnline.openerpage

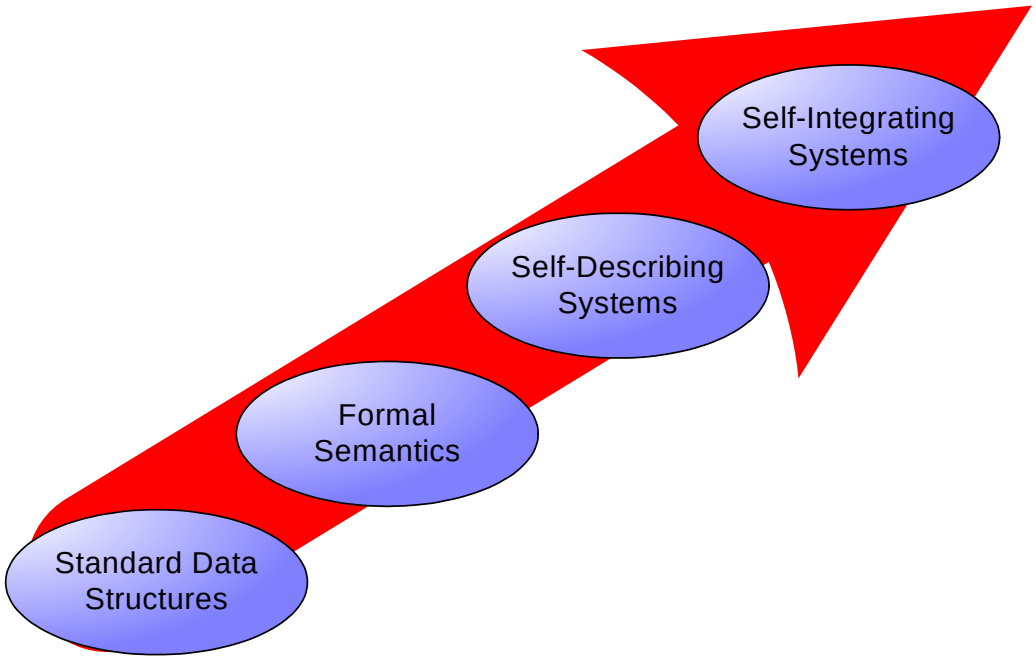


Figure 1. Evolution of information exchange.

More recently, industrial standards are being defined in a more computer-readable form, most notably in XML[5]. This has a number of advantages for developers and implementers, because these specifications can be compiled by computers, databases can be automatically built, and certain kinds of testing can be performed more easily. However, some groups have used XML markups as a substitute for modeling the information – a dangerous shortcut that only works in communities that already share a common understanding of the meaning and usage of terms. A far better approach is to adopt one of the emerging semantic technologies, such as OWL[6], or first order logic[7]. This evolution of representational power toward formal semantics, and the systems integration capabilities that could follow are shown in Figure 1.

At the lower left we see the current state of the art, with XML-based standards as an example. The next bubble shows a generation of standards with definitions of terms (or tags) that are formal enough for a computer to be able to use that definition. This will allow what Tim Berners-Lee calls "self-describing" systems[8], shown in the third bubble– software components that can, on request, provide a formal description of their interfaces, e.g in XML Schema or WSDL[9], along with a formal semantic model of what the tags

[5] See eXtensible Markup Language, http://www.w3.org/XML/
[6] See Web Ontology Language, http://www.w3.org/TR/owl-features/
[7] The Common Logic project (ISO 24707, http://cl.tamu.edu) is standardizing a first-order knowledge representation language that incorporates KIF and Conceptual Graphs. Some of these ideas are also being applied in the Joint US/EUad hoc Agent Markup Language Committee (http://www.daml.org/committee/).
[8] "Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web," by Tim Berners-Lee, Mark Fischetti (Contributor), Michael L. Dertouzos, Harperbusiness, 2000, page 190.
[9] See Web Services Description Language, http://www.w3.org/2002/ws/desc/

mean. It requires a formal, semantic-definition language that is rigorous enough to support logical inference. Finally, in the fourth bubble, is the prize that will unleash the Web for computers — self-integrating systems, i.e. software components that can ask others for a description of their interfaces and then determine how to adjust their own interfaces to be able to communicate with the others. From a systems integration perspective, incorporation of semantic technologies thus opens the possibility of automating the systems integration task. Systems integration is an expensive and tedious task that is consuming an ever-larger slice of budgets within government and industry alike. The possibility of self-describing, and ultimately self-integrating systems can be seen as a generalization of the "plug and play" approach used in today's personal computers, and promises to save tremendous time and effort as well as improve the reliability of integrated systems.

## Automated reasoning about shared concepts

One example of a semantic approach to systems integration is a tool suite developed at NIST called the "Twenty Questions Tool[10]." This tool suite uses the concept of model-theoretic invariants of an ontology to characterize that ontology and to specify the mappings between the ontologies underlying any two applications. Designed to work with the fully axiomatized Process Specification Language (PSL) standard[11], one can generate pairs of precise semantic definitions to characterize how two applications use concepts related to the notion of "process," then generate a concise listing of concepts that are shared between those two applications as well as a listing of exactly how similar concepts differ between the two[12]. Such a tool would be impossible to build without the rigor and expressiveness provided by the first order logic description of PSL.

## A framework for automated integration

The holy grail of systems integration, as depicted in Figure 1, is that of "self-integrating systems" – that is, systems that possess the reasoning capability to inquire about and negotiate with other applications to determine how to exchange information with it and to build the necessary syntactic translators for this purpose. While full realization of this is clearly beyond our reach, one can nevertheless imagine a framework designed to support such a capability.

This has been done at NIST and is described in NIST Interagency Report 7101[13]. This framework calls for explicit conceptual models of each application that is to be integrated, along with the mappings to what are called the engineering models of their interfaces. In the report, the conceptual model captures, with semantic rigor, the functions of the application, the entities it acts on, and the information units it shares. An

---

[10] http://www.nist.gov/psl/20questions.html
[11] http://www.tc184-sc4.org/SC4_Open/SC4_Work_Products_Documents/PSL_(18629)/, and http://www.nist.gov/psl
[12] See also "Semantic Integration Through Invariants" by Gruninger and Kopena, to appear in AI Magazine later in 2004.
[13] "The AMIS Approach to Systems Integration," Libes et al., NISTIR 7101, May 2004, http://www.nist.gov/msidlibrary/doc/nistir7101.pdf

engineering model of an application interface is defined as a specification for that interface in terms of the specific information technologies used to implement it. In particular, an engineering model captures the communication mechanisms to be used – messages, files, databases, procedure calls, object operations – and may capture detailed requirements for their use. It also captures the organization and representation of the data elements that carry the information. To perform an integration, the conceptual models are mapped to a "joint action model" that captures the intersecting semantics of the two applications in the context of a desired interaction. It is the set of semantic mappings that ultimately drives the syntactic translation and reorganization of information as it flows from one application to another (see Figure 2). Parts of this framework have been instantiated – for example, given the semantic models and mappings, we have been able to dynamically generate the syntactic translator[14].

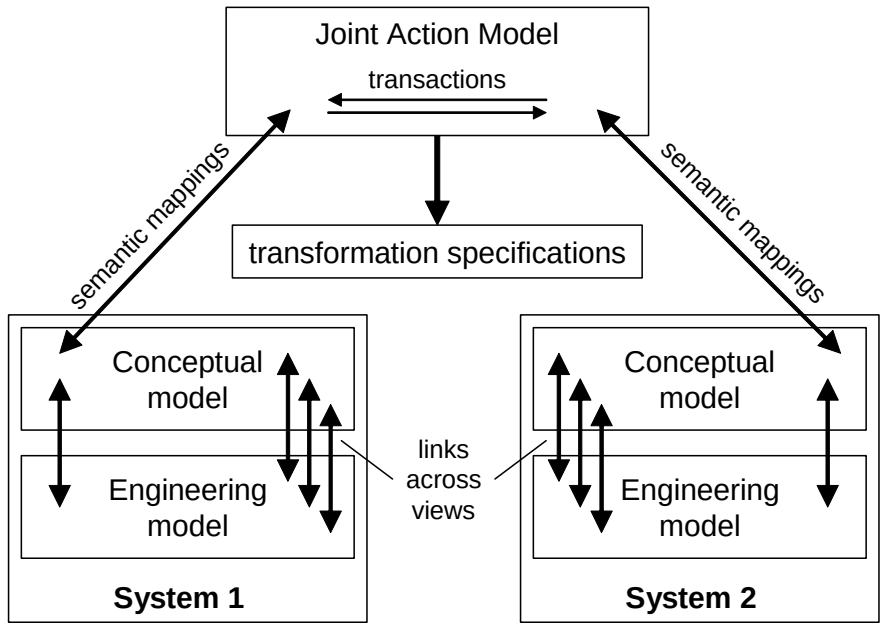


Figure 2. Semantic integration framework, from NIST Interagency Report 7101.

In the future we hope that software applications will be designed and built with explicit semantic models, in addition to the formal engineering models that are the emerging practice. Recognizing the need for such models, we are preparing to extend our own ontology-building activities with an eye toward reusable ontology modules called for in the context of manufacturing, such as schedules, activities, facilities, parties, roles and their relationships to transactions and processes. These ontology modules will be mapped to axiomatic theories where possible, such as those within PSL.

[14] "Automated Composition of Conversion Software," Flater, NISTIR 7099, March 2004, http://www.nist.gov/msidlibrary/doc/nistir7099.pdf

## Conclusion

The use of formal semantics in information exchange standards offers two important benefits. The formality of first order logic provides a degree of rigor and expressiveness not found in other modeling or representational languages. Further, such an approach supports reasoning by software applications in support of dynamic systems integration. It is only with this degree of formality that the full benefits of a semantic approach to information can be realized. We remain committed to the use of formal semantics in our ongoing work in developing interoperability standards.