DEMYSTIFYING INTEGRATION

THIS FRAMEWORK HELPS NAVIGATE THE STANDARDS MAZE TO DEVELOP A FLEXIBLE APPLICATIONS INFRASTRUCTURE THAT ADVANCES AN ORGANIZATION'S STRATEGIC BUSINESS NEEDS.

By Kaushal Chari and Saravanan Seshadri

Driven by emerging opportunities in the marketplace, e-commerce organizations must rapidly and continuously roll out new products and services to support evolving business models. But agility is possible only when an organization's enterprisewide applications infrastructure permits changes and the quick deployment of new functionality. In practice, the infrastructure consists of many integrated software application systems, each designed to meet a particular business objective. These systems might also include interfaces to external systems to facilitate business-to-business (B2B) interactions. Organizations implementing an enterprisewide applications infrastructure to meet immediate business needs often pursue unplanned and ad hoc application systems integration—an undisciplined approach that too often leads to large isolated monolithic application systems involving proprietary architectures, business semantics, and technologies. The resulting brittle infrastructure is unable to adapt to changes as readily as it should.

The ad hoc approach runs counter to a coordinated integration strategy based on a global view of current and future standards, technologies, and business opportunities. Predicting future business opportunities is clearly difficult, but adopting standards-based integration solutions is the most promising way to reduce the long-term costs of integration and facilitate a flexible applications infrastructure capable of responding to business changes.

Adopting a standards-based approach to applications integration is difficult for several reasons. First, standards and specifications are often incompatible, incomplete, or involve overlapping scopes that are not mutually exclusive. This lack of congruence complicates the process of selecting a set of standards covering all aspects of integration for an organization's multiple applications. Second, standards are not always available; for example, the only one available for message-oriented middleware (MOM) is the Java Messaging Service specification, which is restricted to Java-based integration solutions. This opens the door for many proprietary implementations of MOM. Finally, stan-

ILLUSTRATION BY GIANPAOLO PAGNI

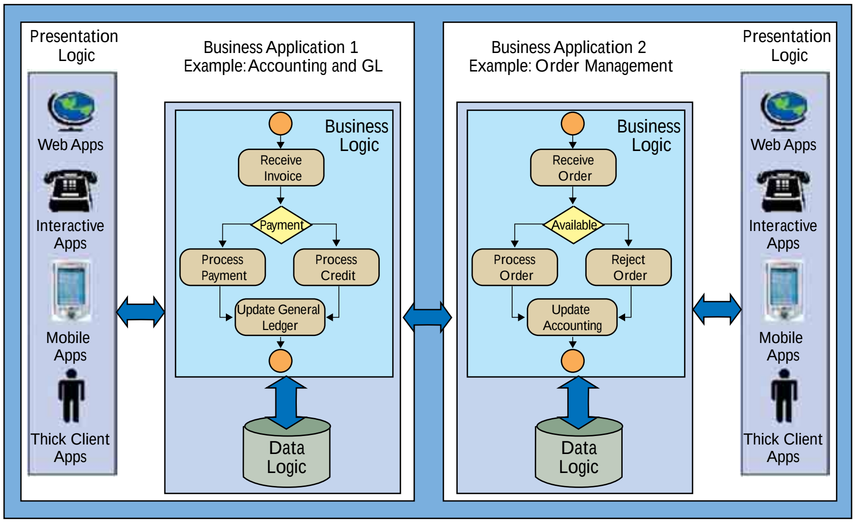

Figure 1. Application components and integration levels.

dards and specifications evolve, further limiting their use in integrated business applications.

Though frameworks available in the literature view systems integration at high levels of abstraction, no existing framework to our knowledge provides a global view of existing standards and specifications when it comes to applications integration (see the sidebar "Sources for Standards and Specifications"). Such a framework would be useful in providing a much-needed perspective for system architects weighing the scope and usefulness of standards and specifications. Here, we propose such a framework. Apart from providing the necessary foundation and perspectives, it also helps identify gaps in the coverage offered by the existing standards and specifications.

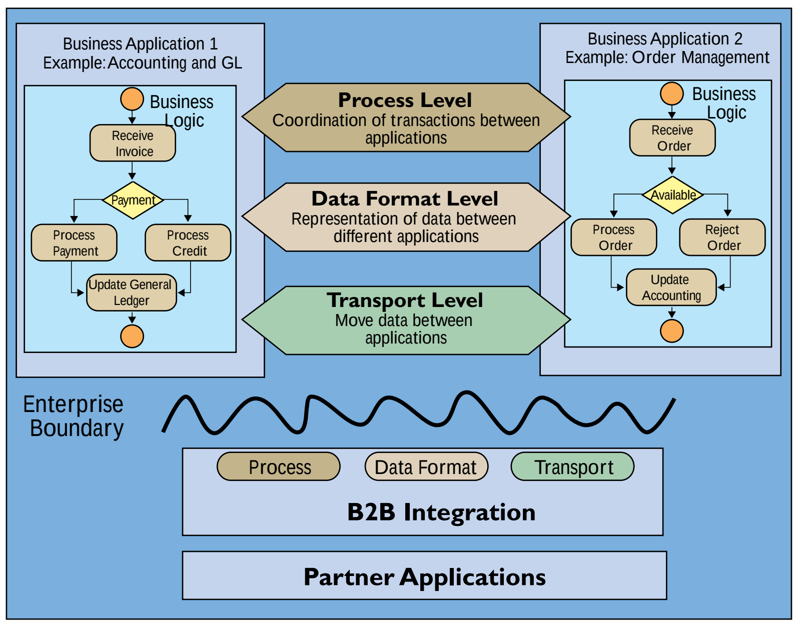

Figure 2. Dimensions of the integration framework.

Common Aspects of Applications

The main objective of the proposed framework is to help practitioners and researchers alike analyze and navigate the maze of standards and specifications available for application integration. Standards and specifications have been developed for respective problem domains covering various components of applications and facets of integration. When building enterprise applications, a number of methodologies and technologies can be used. However, certain aspects of application integration are common in most applications and can be classified into three categories:

The application itself. An application's architecture can be viewed as a collection of three functional components: *presentation* for presenting data and providing the user interface; *business logic* for encapsulating the application's processing logic; and *data logic* for managing persistent data;

Application integration requirements. Integrating an application with other applications affects one or more application components and involves one or more levels of integration, including: *transport* for dealing with the movement of data between integrated applications; *data format* for enabling consistent representation of data between applications; and *process* for coordinating business events between applications; and

Domain independence, domain dependence. Domain-independent standards and specifications are generic in nature, applicable in a variety of domains, while domain-dependent standards and specifications are specific to a vertical industry domain.

In order to facilitate a coordinated integration strategy using a global view of the existing standards, any holistic framework must incorporate all the characteristics of the application-integration environment. These characteristics are the basis for defining the dimensions of the proposed framework.

Adopting standards-based integration solutions is the most promising way to reduce the long-term costs of integration and facilitate a flexible infrastructure.

Dimensions of the Framework

Application-integration standards and specifications can be organized along the lines of three orthogonal dimensions:

Applications architecture. This dimension represents an application's three functional components, as outlined in Figure 1:

Presentation logic. Provides the ability to manage the interactions between an application system and its various presentation interfaces, including Web browsers, interactive voice recognition, mobile computing devices, fat clients, and dumb terminals;

Business logic. Implements the business rules that represent the business processes of an application system; an example is the logic that encodes the rules needed to process an invoice and accounts payable; and

Data logic. Provides the ability to access and map data into a form that can be processed by business logic; an example is accounts data retrieved from a database and encapsulated as a Java bean for use by the business logic.

Integration level. This dimension represents the level of integration (there are three), as outlined in Figure 2:

Transport. Provides the infrastructure and abstraction over the communication protocols needed to move data between similar and dissimilar entities in a transparent manner. Examples of tasks involved at this level are transport protocol bindings, routing, and reliable transport of messages across applications using asynchronous and synchronous mechanisms;

Data. Facilitates integration of business applications by addressing the representation data elements in different systems, packaging data elements into formatted messages, and associated transformation rules. Typical tasks at this level are encoding data, formatting and specifying data fields based on an ontology, and packaging messages by including data, as well as metadata, for encoding, message content, security, and message delivery; and

Process. Pertains to the integration of at least two different entities (such as application systems, lines of business, and enterprises based on business rules and events); tasks at this level include orchestrating process interactions based on business rules and events providing process context, and handling process exceptions.

Industry domain specificity. This dimension represents the scope of the integration standards (there are two types), that is, whether they are generic and applicable to a variety of industries or specific to a particular industry:

Industry domain independence. Standards are

Sources for Standards and Specifications

Recognized standards bodies include the International Standards Organization (ISO), International Telecommunications Union (ITU), Institute of Electrical and Electronics Engineers (IEEE), Internet Engineering Task Force (IETF), and American National Standards Institute (ANSI). Specifications approved by these organizations after a formal process are known as de jure standards. Mature specifications in the pipeline for approval are known as standards-in-making.

Consortia include the World Wide Web Consortium (W3C), Object Management Group (OMG), Open Applications Group, Inc. (OAGI), and Organization for the Advancement of Structured Information Standards (OASIS). Consortia standards may be submitted to recognized standards bodies for approval as de jure standards.

Finally, companies or groups of companies may publish specifications and submit them to recognized standards bodies for approval as de facto standards or to consortia for approval as consortia standards.

Standards and specifications not approved by recognized standards bodies but that are widely adopted are known as de facto standards. C generic, providing capabilities across multiple industry domains; examples are HTTP 1.1 and SQL; and

Industry domain dependence. Standards are specific to a particular vertical industry domain; examples are RosettaNet and HL7.

The three dimensions of the integration framework—applications architecture, integration levels, and industry-domain specificity—help practitioners identify the integration standards in Tables 1 and 2. For example, HTTP 1.1 is included in the cell "Domain Independent, Data Logic, Transport Level," since it is used to access information from a server and transport it to a client. However, HTTP 1.1 is not included in the cell "Domain Independent, Business Logic, Transport Level," since nothing inherent about HTTP 1.1 pertains to business logic, even though HTTP 1.1 could be used as a transport protocol for business logic.

In some cases, standards with broader scope are placed in more than one cell. For example, RosettaNet Implementation Framework 2.0 covers transport-level and data-format-level issues for data, as well as for business logic, and is therefore in four cells. The presence of a standard in a cell in Tables 1 and 2 does not indicate that it encompasses all the features of that cell. Tables 1 and 2 list only notable nonexhaustive standards, as it is not possible to provide a complete list. The color codes in the table legend identify the type of standard.

Table 1. Classification of domain-independent standards and specifications for application integration [1].

Industry Domain	Applications Architecture	Integration Level	Standard
Domain Independent	Data Logic	Transport	FTP
			OSI-RPC
			SMTP
			SQL/CLI
			HTTP 1.1
			CORBA/IIOP 3.0.2
			ebXML-Message Service 1.0
			OAMAS 1.0
			JDBC 3.0
			JMS 1.1
			ODBC 3.51
			WS-Routing
		Data Format	SQL
			MIME
			S/MIME 3.0
			ebXML-Message Service 1.0
			ebXML Core Components (Document Assembly) 1.04
			OAGIS-BOD 8.0
			SOAP 1.2
			XML–DTD 1.1
			XML Schema
			XQuery 1.0
			JDBC 3.0
			ODBC 3.51
		Process	OAGIS-BOD 8.0
			UML 1.5
			WSDL 2.0
	Business Logic	Transport	ebXML-CPP 2.0
			OAMAS 1.0
		Data Format	EDIFACT
			ANSI X.12
			X.12
			ebXML Core Components (Document Assembly) 1.04
			OAGIS-BOD 8.0
			SOAP 1.2
			WFMC Interfaces
			BizTalk Framework 2.0
		Process	ebXML-BPSS 1.01
			ebXML-CPA 2.0
			OAGIS-BOD 8.0
			OAGIS-Integration Scenarios 8.0
			UML 1.5
			WFMC Reference Model and Interfaces
			BPML 1.0
			BizTalk Framework 2.0
			BPEL4WS 1.1
	Presentation Logic	Transport	ISO/IEC 6429:1992
			Telnet
			HTTP 1.1
			TN 3270E
			WAP 2.0 (lower layers)
		Data Format	HTML 4.01
			XSL 1.0
			WAP–WML 2.0
			JSP 2.0
		Process	UML 1.5

Observations

Based on the framework presented here, a number of observations can be made regarding standards and their coverage and usability in enterprise applications:

- Many nondomain-specific transport-level standards are available for use in domain-specific higher-level integration standards. As these transport-level standards have matured, it has become easier for many domain-specific standards (such as RosettaNet) to focus on higher domain-specific layers;
- Nondomain-specific standards have tended to

EVEN NEW STANDARD INITIATIVES DO NOT ADDRESS THE MIGRATION STRATEGIES ORGANIZATIONS MIGHT CHOOSE TO FOLLOW IN ADOPTING THE NEW STANDARDS.

Industry Domain	Applications Architecture	Integration Level	Standard
Domain Dependent	Data Logic	Transport	ACORD XML Business Message Specification for P&C Insurance and Surety 1.4.1
			RosettaNet Implementation Framework 2.0
		Data Format	ANSI X12N
			HL7 2.5
			HL7 CCOW 1.3
			ISO 15022
			ACORD Automation Level 3
			ACORD OlifE 2.6.00
			ACORD XML Business Message Specification for P&C Insurance and Surety 1.4.1
			RosettaNet Dictionaries
			RosettaNet Implementation Framework 2.0
			SPEC2000
		Process	HL7 2.5
			HL7 CCOW 1.3
			RosettaNet Partner Interface Process
			SPEC2000
	Business Logic	Transport	ACORD XML Business Message Specification for P&C Insurance and Surety 1.4.1
			RosettaNet Implementation Framework 2.0
		Data Format	ANSI X12N
			HL7 2.5
			ISO 15022
			ACORD Automation Level 3
			ACORD OLifE 2.6.00
			ACORD XML Business Message Specification for P&C Insurance and Surety 1.4.1
			RosettaNet Dictionaries
			RosettaNet Implementation Framework 2.0
			SPEC2000
		Process	HL7 2.5
			RosettaNet Partner Interface Process
			SPEC2000
	Presentation Logic	Transport	
		Data Format	HL7 CCOW 1.3
		Process	HL7 CCOW 1.3

Table 2. Classification of domain-dependent standards and specifications for application integration [1].

Legend

- De jure standard
- De jure standard-in-making
- Consortia-based standard
- Consortia-based standard-in-making
- Specifications

Legend. Color codes for standards and specifications in Tables 1 and 2.

focus on technology-specific efforts in a bottom-up manner without mapping the needs of vertical industries;

- The Business Process Modeling Language, the Business Process Execution Language for Web Services, and other standards are emerging to address business-level integration, ultimately facilitating integration to be carried out at the business-process level rather than at the technology level;
- Like X.500 for names, no standards exist for representing metadata information about integration that is accessible in common integration repositories. Perhaps the Universal Description, Discovery and Integration standard and the Web Services Description Language could be used, though both seem more focused on B2B and Web-based services;
- Domain-specific standards tend to focus on B2B integration;
- Domain-specific standards do not generally address all facets of integration, relying instead on domain-independent standards or proprietary solutions for bridging the gap;
- There is a dearth of domain-specific standards for addressing process-level integration. Most of the domain-specific effort seems to stop at defining data-format integration;
- There are no standards for facilitating common business services across vertical domains that enable quicker integration. For example, human resources management and financial transactions are common to all vertical domains, yet no service standards are related to these common corporate administrative functions;
- Creating common presentation standards within vertical domains does not seem to be a priority; and
- Even new standard initiatives do not address the migration strategies organizations might choose to follow in adopting the new standards.

The framework and these observations reinforce the need for practitioners and organizations to create their own evolving standards-based strategy for managing application integration. Moreover, the observations highlight the need for enhancing existing standards for effective integration. **C**

References

1. Chari, K. and Seshadri, S. *A Glossary of Standards and Specifications for Application Integration*, 2004; see coba.usf.edu/chari/glossary.html.
2. Fingar, P. Component-based frameworks for e-commerce. *Commun. ACM 43*, 10 (Oct. 2000), 61–66.
3. Hasselbring, W. Information system integration. *Commun. ACM 43*, 6 (June 2000), 33–38.

Kaushal Chari (kchari@coba.usf.edu) is an associate professor in the Department of Information Systems and Decision Sciences in the College of Business Administration at the University of South Florida in Tampa.

Saravanan Seshadri (saru@ultramatics.com) is founder and president of Ultramatics, Inc. in Tampa, FL.