

CICA Executive Summary

Copyright © 2002 Data Interchange Standards Association

A reference model recommends an architecture and guidelines for the design of further standards. For ANSI ASC X12, the *X12 XML Reference Model* outlines a method for building and interpreting electronic business documents that provides a predictable structure with the flexibility required by different business environments. It also provides recommendations and best practices for XML syntax in the design of schemas and XML messages, called instance documents.

To conduct business electronically with trading partners has meant for the past two decades using Electronic Data Interchange (EDI). Over the past five years, commerce of all kinds over the Internet has also boomed, with many organizations now using EDI or Internet exchanges (or both), to reduce overhead, better manage inventories, open new markets, and improve customer service.

While many larger enterprises may benefit from these technologies, most smaller companies have been left out. Small companies (and not just small companies) have found EDI too expensive, too complex for their limited IT resources, and too inflexible when new opportunities arise. Much of the EDI expense comes from implementation, with the manual intensive process of developing, interpreting and incorporating implementation details, requiring EDI experts. As a result, EDI often cannot provide the agility companies need to respond to new business opportunities or integrate data with business partners. Extending e-business to the vast majority of smaller businesses requires a different approach, but one that provides similar benefits and safeguards enjoyed by larger enterprises.

Much of the burden for alleviating these problems has fallen on the e-business standards community, and recent advances show promise in removing the barriers. The eXtensible Markup Language (XML) offers the ability to exchange structured data over Internet protocols. The Electronic Business XML or ebXML specifications builds on the XML specifications to provide a basic set of e-business capabilities, including business process definitions, reliable messaging, models for registries for specifications, models for company profiles, models for technical trading-partner agreements, and semantic interoperability through a neutral set of core components. Emerging Web services standards based on XML also offer capabilities for interactive messaging, security, and discovery and descriptions of commercial services and business processes.

While these are indeed important advances, they still lack important features needed to make e-business feasible for smaller organizations. One critical missing feature is a systematic way of assembling and interpreting business messages that meets the needs of specific industries, but can still interact across industries when companies need to do business outside of their normal business relationships. This capability makes it easier to take advantage of new business opportunities, as well as transact business with financial, transportation, or government (and other) services that cut across industry boundaries and often have their own terminology.

Most businesses also need their basic, off-the-shelf business software to send, receive, and process transactions directly. Smaller companies, like their larger enterprise counterparts, need to integrate the data exchanged with business partners into their normal management and control software, often relying on common accounting packages for this purpose. Providing more capability to interact with small-business software will go a long way in making e-business work for the millions of companies using this software.

The *X12 XML Reference Model* uses a Context Inspired Component Architecture or CICA that presents seven levels of semantic granularity, from the entire document at the

CICA	X12 EDI
Document	Implementation Guide
Template/ Linked Template	Transaction Set
Module	Implementation Guide Major Loop
Assembly	Minor, generic loop
Block	Moderate segment
Component	Single element or composite container
Primitive	Single piece of data

Table 1

top, down to primitives that contain one discrete piece of data. The over arching difference between CICA and EDI is that CICA encompasses both implementation and traditional EDI standards. This fundamental difference is what motivates the attention to sizing, organizing and classification of architecture elements [granularity model], and the when and how to put them together [document assembly model]. In spite of these differences, some comparisons can be made between existing EDI and CICA, shown in Table 1.

With CICA's scope including both standards and implementation contents, management features are designed for both 'big picture' and each individual use levels. The key is in the Template, illustrated in Figure 1. Templates are created for each business process need for a document, Invoice, Order, etc., containing Slots for each significant Party, Event, Location and Resource. Slots are empty 'place holders' specifying a business use with details specified separately with Modules. In the example, two Templates are shown, Invoice and Order, each containing a set of Slots. Buyer and Seller Slots appear in both Templates, but in this example a Slot for \$'s is in the Invoice, but not in the Order.

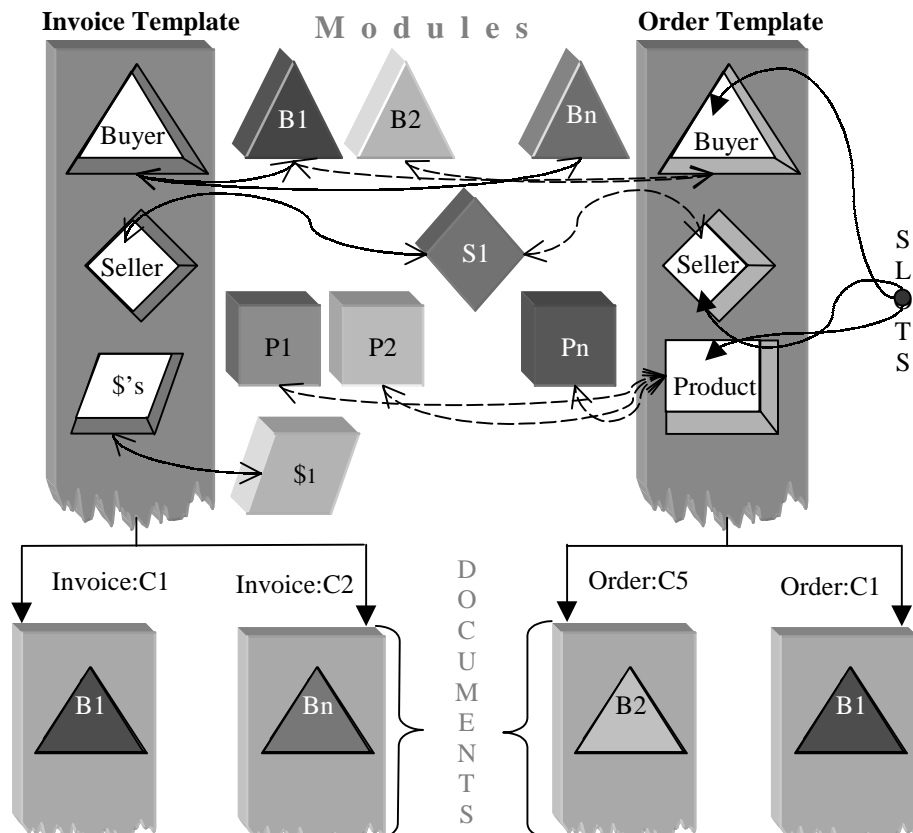


Figure 1

Modules, shown in the middle in Figure 1, are created and linked as needed to the Templates, using Context to specify when the link applies. This simple construct enables maximum reuse and flexibility, with a minimum of complexity. Some kinds of Modules cover subjects where significant variability must be enabled, such as with Product where diversity ranges from office supplies to visiting nurse services. In contrast, specifying Payment has little variability. In these extreme cases, and those in between, CICA's simple approach equally supports the business requirements with simplicity. Finally, Documents are derived from the Template, based on context decisions that choose Modules to occupy the Slots, shown in the lower portion of Figure 1. This simple example illustrates the basics of the CICA architecture. For a complete list of the layers of the CICA architecture with definitions and examples, refer to table 2. This modular approach supports the need for flexibility required by industry, while at the same time placing a priority on organization to meet the needs of end-user packaged application software vendors.

CICA takes advantage of the groundbreaking work done by UN/CEFACT on ebXML core components, building on the ideas of semantically neutral information entities and the critical influence of context. CICA adds to these ideas by providing more levels of granularity, by identifying roles for core components as CICA assemblies, blocks, and components. Likewise, CICA gives a role for business information entities or BIEs, as modules in CICA that combine business context with the semantically neutral blocks.

The reference model includes rules to determine the semantic level at which information components are related, to give a quantitative basis for deciding when data items are equivalent or the extent to which they differ. The model uses the concepts of form, fit and function. Form refers to the physical structure

and content, fit covers the item's intent or meaning, and function concerns the purpose or how the item is used. When all three conditions are equivalent, the items are considered semantically equal, for example a company called seller in one industry and a supplier in another.

The Technical Report takes the concepts in CICA and spells out the metadata for building XML schemas and instance documents, presenting seven layers, one for each level of granularity in the architecture. The reference model provides guidance on XML design issues, covering nuts-and-bolts concerns such as naming conventions, as well as technical issues involving namespaces, XML Schema usage, and XLink/XPointer applications.

Layer	Definition	Example
<i>Document</i>	complete processable message containing data combined with the business context needed by business partners	Invoice from a manufacturer to a customer
<i>Template</i>	the framework of the document with slots or placeholders for modules in the message	An invoice with Slots for Buyer, Seller, Line Product, Financial Obligation, etc.
<i>Module</i>	adds business context, such as special industry terminology or business process requirements or legal constraints, with the neutral data in the assembly or Block into meaningful pieces of information to the business partners.	Buyer, with all of the component parts necessary to specify the Buyer role including Address details
<i>Assembly</i>	links sets of blocks into coherent collections of data that businesses can reuse as needed	An Organization with an address
<i>Block</i>	specifies single parties, resources, events, or locations, composed of combinations of identity and characteristics components	An Organization Party
<i>Component</i>	Finest level of detail that indicates the identity or characteristics of business data to describe parties, resources, events, or locations in a document	ID number used to specify an Organization Party
<i>Primitive</i>	one discrete piece of data within a component	ID Value, an indicator specifying type of ID number

Table 2 CICA Architecture Layers