XBRL and its relationship to XML Web Services, ebXML and other infrastructures for e-Business

Walter Hamscher (walter@hamscher.com) 25 May 2002

Purpose

The eXtensible Business Reporting Language (XBRL) is a markup language for financial and business reporting. Although it was designed from the outset to be "Internet native," its scope focuses entirely on the *content* of a business report, and it is completely neutral with respect to the technical means by which e-reporting is accomplished. In other words, how one organization should get its XBRL document to another – how a bank should send its XBRL-formatted quarterly report to a regulator, how a subsidiary gets its monthly XBRL-formatted P&L to corporate headquarters – is not particularly constrained by XBRL. This is as it should be, since that is what makes financial reports expressed in XBRL reusable.

However, at this stage of its maturity, whenever an XBRL taxonomy (in effect, a tag set) is built, that effort tends to go hand-in-hand with the development of a specific XBRL-based application to *use* that taxonomy, so that certain design decisions occur repeatedly.

The goal of this note is to sketch some of the salient issues, business processes, and likely implementation profiles of using XBRL, not only to facilitate implementation, but also to facilitate ongoing discussion with the advocates of various e-business infrastructure standards currently on offer.

Business Reporting Processes

If a constantly changing set of reporting organizations ("producers") is going to transmit business reports to at least one receiving authority ("consumer"), there are certain interactions between those participants that are always going to occur and that must be enabled by the technology in some way. In e-business, these are called not only "business processes" but "interface processes," "transaction sets," "message sets," and so on. Here are some groups of closely related processes:

1. Registering the producer; retrieving and updating the standing data of the producer.
    - Example: getting a new tax identifier for a company, changing its main phone number or address.
2. Registering, revoking and changing the authorized users within the producing organization.
    - Example: registering the officers of a bank that are allowed to submit Call Reports to the national Deposit Insurance Corporation.
3. Registering, revoking and changing the authorized intermediaries that are reporting on behalf of the producer;

  - Example: informing the Inland Revenue that a particular office of a particular accounting firm will be filing their tax forms.

4. Obtaining new versions of the electronic filing form; receiving other change notices;
   - Example: last year's form asked only for the number of employees as of a given date; the new form asks for full-time employees, part-time employees, and employees on leave as three separate figures.
5. Submitting a test report, submitting an actual report; Obtaining an error check or analysis on a report;
   - Example: the compliance officer at a bank, having just installed new reporting software, tests their end-to-end automated reporting thoroughly with test data before "going live."
6. Recalling a previously submitted report; submitting an updated report; modifying individual data items within a report;
   - Example: one of the 1500 separate data items required by a complex annual filing turns out, after the deadline, to have had transposed digits because it was manually entered.
7. Confirming acceptance or rejection of a report; obtaining a transaction history, version history; access log, and detailed audit trail of reports, by organization and/or report.
   - Example: the General Manager of a foreign subsidiary wants to verify that the past two years of business data went to corporate headquarters on time.

These seven groups of processes are in no way unique to XBRL based financial reporting. Those familiar with systems predating XBRL such as the SEC EDGAR system or Canada's SEDAR, or anyone who has electronically filed their taxes from their home computer, will recognize analogues of these processes – it does not even matter whether all of the processes are performed online, much less whether they use the Internet. The point is, they all have to be enabled somehow. And to be successful, it must be easy to add producers, make changes in the information exchanged, maintain an environment where reports once filed are not lost, that return receipts are generated, etc.

Infrastructures for e-Business support e-Reporting

There are several e-business infrastructures capable of supporting business reporting, since the processes listed above are not particularly unique to business reporting. If you were to imagine each one of the listed processes as having nothing to do with business reporting at all, but with transactions -- such as making a plane reservation or ordering 500 kilograms of coffee beans -- it should be clear that *any* business-to-business platform worth its salt must supply all of these capabilities to some degree.

This is where XML Web Services, ebXML, and other e-business infrastructure standards – RosettaNet, ANX, EDIFACT, OFX2, OAGIS – come into play. Every one of these has some answer to such questions as "How does a participant register?" "How does the sender know their report arrived?" and "How does one flag a transaction as a test transaction as opposed to a live one?" They differ widely with respect to many different factors of importance to anyone building an XBRL-enabled solution:

1. Level of maturity and availability of vendor solutions;
2. Ability to leverage the low cost and wide availability of the Internet;
3. Robustness and manageability of privacy, authentication, integrity, etc;
4. Ease with which they scale by adding new participants to a live network;
5. Ease with which new transactions and content payloads can be added;
6. Ability to formally define the "choreography" of the interactions in sufficient detail that tools can generate robust software that implements the process.

The point is that these infrastructures are clearly sufficient for business reporting. They may be more complex than is in fact *necessary* for business reporting, but they can certainly do the job. Also, in XML Web Services[1] and ebXML, the entire payload of any business message (a transaction is made up of individual messages) is XML; since XBRL is defined using XML Schema, it would seem that one of these is a natural fit for business reporting using XBRL.

If the application calls for business transactions to behave differently (to be routed to a different location) based on the content of the message, it is important to be able to define the way the payload is formatted. For example, suppose that an XBRL instance document contains financial information for TLA, Inc., and its three subsidiaries for the five years ending 30 June 2002. Suppose further that the receiving system needs to send the incoming data off to several other systems based on categories such as "large companies" "companies with foreign income" "oil companies." By using either ebXML or XML Web Services, it is easier for the software to examine the contents of the XBRL formatted financial data, and for it to make the processing decision immediately when routing the messages, rather than waiting until the data is deposited in some other system, and then processed later.

Validation of incoming data is critical in a business reporting application (or, more generally, the application of "business rules" to data). These e-business infrastructures simply use a grammar of some type (e.g. XML Schema) to validate data at the level of data types (e.g., the date-of-birth field is dd/mm/yy) or structures (e.g., each line of an invoice must have a quantity, product code, price and discount). Validating that the data obeys co-constraints (e.g., if taxes paid are greater than owed, it must be indicated whether the difference is a cash refund or prepayment) is left to application-specific tools.

XBRL, the Content Payload for Business Reporting

Just as it is important for those designing e-reporting systems to appreciate e-business infrastructure standards, it is important that those who are building e-business infrastructures (portals, marketplaces, trading networks) be aware of XBRL and its

[1] Unlike ebXML, the XML Web Services protocol stack (SOAP, WSDL, UDDI) does not define all of the security management and the entire process infrastructure that would probably be needed for e-reporting. However, it has certain advantages of relative simplicity, industry momentum and widespread familiarity, and does deserve consideration alongside ebXML.

features: financial and business reporting applications need not re-invent business reporting. XBRL is the only open specification in the world that is:

- Optimized for the exchange of historical, archival business reporting data;
- Models data that is hierarchically arranged for drill-down and reported along dimensions of time, entity, and scenario (or context);
- Independent of specific industry, regulatory regime or level of detail;
- Supported by major accounting professional societies worldwide.

The absence of any other financial reporting standard having garnered the support of any of the accounting professional societies means that XBRL currently occupies, and will occupy for a considerable period at least, the unique position of being the worldwide standard for financial reporting that encompasses both internal and external reporting. Key themes distinguish XBRL from other standards within the financial arena:

- Reports, as distinct from transactions. A purchase order, or, more precisely, the sending and acceptance of a purchase order, is a transaction; transactions are the purpose of a whole host of financial standards including IFX, OFX2, ACORD and others. XBRL is for reporting.

- Performance data, as distinct from market data. Market data tends to be ephemeral and real-time, with pricing being always crucial; performance data is archival and records the history of business operations and their results. XBRL is about performance data.

- Entities, as distinct from investment instruments. Equities are financial instruments whose underlying value is based on public company entities; an entity is the business itself. XBRL represents detail about entities -- not only publicly traded companies, but any business or non-profit entity. Equities and other financial instruments are the subject matter of MDDL, FpML and others.

- Reporting metadata, as distinct from document metadata. Metadata – data about data – is, for the most part, data about a document. Standards such as the Dublin Core define the data that describe a documents (which themselves contain data) taken as a whole. XBRL defines how the individual numbers and facts *inside* the financial statements and similar documents relate to one another.

Hence, XBRL business content can be embedded into any other standard that is related to transactions, market data, instruments, or document metadata. For example, if there were a standard ebXML business process for tax reporting, XBRL could be used as part of the “payload” of the tax return itself, since it is used to represent financial statement level data as well as the ledger of business transactions that are classified into different tax treatments.

Conclusion

The synergy between XBRL and other e-business infrastructure standards was deliberate in design, and it is ready to exploit in building e-reporting systems. Publishing a formal definition of generic business reporting using a process definition language would be a useful next step in order to facilitate future e-reporting applications.