



Implementing XBRL-based Applications – The Benefits of Using a Processing Engine

Executive Summary

This white paper is intended for software developers and architects who have an interest in implementing XBRL support within their products and services, and for IT managers who are evaluating software which they will use to implement XBRL within their enterprises. The document starts with a brief overview of the basic concepts and terminology of XBRL and then provides a short overview of XBRL processing engine functionality. The paper concludes with a summary of the benefits developers can realize by utilizing an XBRL processing engine as opposed to hand-coding applications.

Introduction

XBRL (eXtensible Business Reporting Language) is one of a family of XML (eXtensible Markup Languages) which has become a standard means of communicating information between businesses and on the Internet. XBRL provides major benefits to organizations of all sizes in the preparation, analysis, and communication of business information. It offers cost savings, greater efficiency, improved accuracy, and reliability to all those involved in supplying or using financial data.

Adoption of XBRL is rapidly gaining momentum worldwide. Examples of XBRL-based systems include:

- Since 2005, the United States Federal Deposit Insurance Corporation (FDIC) has required the use of XBRL when collecting quarterly call reports from 8,000 financial institutions.
- European banking supervisors, including Bank de France, National Bank of Poland, and Kredittilsynet (The Financial Supervisory Authority of Norway), are all collecting Basel II information from reporting banks in XBRL.
- The Dutch Ministry of Finance is implementing a countrywide effort to use an XBRL-based system to collect census and tax information from all businesses in the Netherlands.
- The United States Securities and Exchange Commission has been piloting the use of XBRL for public company filings for two years and is expected to mandate the use of XBRL this year.
- Japan's Financial Services Agency announced that disclosure statements submitted to EDINET (Electronic Disclosure for Investors NETwork – Japan's EDGAR system) must be in XBRL format beginning in the year 2008.
- China was the first country to require mandatory filing of data in XBRL in 2004 when it made the filing of financial reports in XBRL mandatory for all companies listed on the Shanghai Stock Exchange.

Deploying these systems requires applications that can create, validate, and process XBRL documents. Corporate software developers, systems vendors, and integrators will need to make decisions about how to address these requirements to generate and utilize XBRL. In addition, end users will have to evaluate the software they procure to ensure that it can meet all new XBRL reporting requirements.

A Brief Overview of XBRL

The XBRL standard is managed by XBRL International, a not-for-profit consortium of approximately 550 companies and agencies working together to develop the XBRL standard and promote and support its adoption. This collaborative effort began in 1998 and has produced a variety of specifications and taxonomies to support the goal of providing a standard, XML-based language for digitizing business reports in accordance with the rules of accounting in each country or with other reporting regimes, such as banking regulation or performance benchmarking.

XBRL is an open standard – free of license fees. It enables unique identifying tags to be applied to items of financial data, such as “net profit.” However, these tags are more than just simple identifiers. They provide a range of infor-

mation about the item, such as whether it is a monetary item, a percentage, or a fraction. XBRL allows labels in any language to be applied to items, as well as accounting references or other subsidiary information.

XBRL can also define how items are related to one another. For example, it can represent items that are calculated from one another by being subtotaled or totaled. With the upcoming XBRL Formula Specification, it will allow formulas which can create business rules (e.g., raise a flag if cost of sales is great than 50% of sales). It can also identify whether items fall into particular groupings for organizational or presentational purposes. Most importantly, XBRL is extensible in a structured way which allows companies and other organizations to customize existing tag libraries to meet a variety of special requirements.

The rich and powerful structure of XBRL allows very efficient handling of business data by computer software. It supports all the standard tasks involved in compiling, storing, and using business data. Such information can be converted into XBRL by suitable mapping processes, or generated in XBRL by software processing engines. It can then be searched, selected, exchanged, analyzed by computer, or published for ordinary viewing.

Requirements for Submitters and Collectors

There are two participants in the exchange of information, the submitters of information and collectors who gather and analyze information – each with specific requirements to process XBRL documents:

- Submitters are responsible for creating and reporting information, such as banks or regulated companies. These organizations need to create and validate XBRL documents on a yearly, quarterly, monthly, weekly, or even daily basis. Submitters may need to gather information from multiple sources (e.g., ERP systems, market risk systems) and consolidate the information into a valid XBRL instance document. The creation of the instance document requires a full understanding of the taxonomy being used (concepts, relationships, calculation and formula information).
- Collectors are responsible for gathering XBRL documents, including regulators such as the FDIC or information hubs such as the MIX. These organizations need to build enterprise-class applications that support XBRL to validate and process hundreds, thousands, or even millions of XBRL documents each reporting cycle. Collector organizations not only need the ability to process XBRL documents – they also need software to manage the workflow surrounding the reporting process.

KEY DEFINITIONS

XBRL taxonomies are the dictionaries or schemas which the language uses which define the specific tags for individual items of data (such as “net profit”). National jurisdictions have different accounting regulations, so each may have its own taxonomy for financial reporting. Many different organizations – including regulators, specific industries, or even companies – may also require additional taxonomies to cover their own business reporting needs. Taxonomies in widespread public use include IFRS (International Financial Reporting Standards), US-GAAP (United States Generally Accepted Accounting Principles), FINREP (financial reporting), and (COREP) COmmon solvency ratio RE-Porting framework) for Basel II reporting.

XBRL instance documents are business reports in an electronic format created according to the rules of XBRL. These documents contain facts that are defined by the elements in the taxonomy they refer to, together with their values, units, and an explanation of the context in which they are placed.

XBRL Processing Engines

An XBRL processing engine is the essential component of any application that plans to read, write, validate, or process XBRL. Much like a database engine that uses SQL to read and write data to a set of relational tables (or schema), an XBRL processing engine reads and writes XBRL based upon the information, such as presentation, calculations, dimensions, and business rules defined in an XBRL taxonomy.

The UBmatrix Processing Engine is the ideal platform for the development and deployment of XBRL-based applications. It is designed for high volume, large scale business processing. The engine provides taxonomy and instance document validation and conversion modules which allow for transformation of XBRL from and into Excel, plain text (CVS), XML or HTML.

UBmatrix Processing Engine is built on an extensible object-oriented architecture and enables developers to easily integrate XBRL processing into either Java or .Net applications. The processing engine fully conforms to the latest XBRL standards and also supports formulas in anticipation of the upcoming XBRL International standard. With the UBmatrix Processing Engine, software developers can:

- Easily XBRL-enable existing technology stacks and applications;
- Support full XBRL validation, including calculation checks and validation against the taxonomy, which verifies not only the syntax, but also exercises basic business rules to ensure valid content;
- Support Discoverable Taxonomy Set (DTS)-aware XML Stylesheet Transformations (XSLT) including custom defined extensions, and XML Path Language 2.0 (X-Path) for taxonomy and instance document transformation, and rendering XBRL data into a multitude of formats readable by both computers and people;
- Validate full compliance with the XBRL 2.1 specification; and
- Fully support Financial Reporting Taxonomies Architecture (FRTA 1.0) validation.

XBRL Processing Engines The Benefits of Creating, Reading, or Processing XBRL Using a Processing Engine

By utilizing an XBRL processing engine, XML developers and ISVs obtain a high-performance processor for building and deploying applications for business reporting, data interchange, and service-oriented architectures (SOA). The engine can save developers from spending a lot of time hand-coding their own solutions. Deploying XBRL-enabled business and financial applications with an XBRL processing engine also enables organizations to streamline the process of extracting XBRL data automatically and validating calculation links to ensure data accuracy.

XBRL processing engines integrate into the existing IT infrastructure to facilitate the validation and transformation of XBRL data into and out of enterprise applications such as ERP, CRM, financials, spreadsheets, analytics, and others. Organizations that collect and process XBRL data will benefit from a highly scalable and powerful processing engine that facilitates transformation of XBRL data into any format.

In summary, an XBRL processing engine:

- Provides increased flexibility – It makes it easier to maintain software – especially if the taxonomy changes over time.
- Cuts development costs and speeds time to market – XBRL is complex standard. A processing engine makes it easier and quicker to build more feature-rich software, with advanced capabilities like calculation trace and rendering. It would take multiple man-years to hand-code all of the functionality provided by a processing engine.
- Enables easier validation – A processing engine provides the ability to validate XBRL computations and business rules, either for submitters or collectors of information.
- Reduces risks and eliminates pitfalls in XBRL implementations – By leveraging a tested solution, enterprises can reduce risks and leverage best practices in XBRL programming.

KEY DEFINITIONS CONTINUED

A Discoverable Taxonomy Set (DTS) is a set of related taxonomy schemas and linkbases.

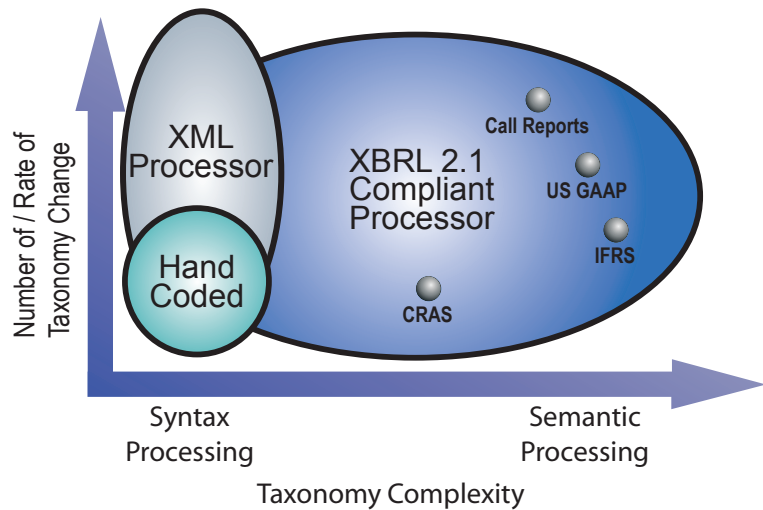
XML Stylesheet Transformations (XSLT) is a language for transforming source XML documents into other document formats using XML Path Language (XPath) as a query language.

XPath 2.0 is an expression language that allows the processing of values conforming to the data model defined in [XQuery/XPath Data Model (XDM)]. <http://www.w3.org/TR/xpath20/>

XBRL 2.1 specification can be found at: <http://www.xbrl.org/SpecRecommendations/>

FRTA provides the means by which disclosures are made. The recommended architecture establishes rules and conventions that assist in comprehension, usage, and performance among different financial reporting taxonomies. <http://www.xbrl.org/technical/guidance/FRTA-RECOMMENDATION-2005-04-25.htm>

- Facilitates XBRL compliance – By using fully conforming XBRL processing engine, developers can ensure that their applications will meet complex XBRL compliance regulations.
- Shifts developer focus to more strategic projects – A processing engine can eliminate the time required to hand-code application, enabling developers to work on the company's core products/competencies and concentrate on more strategic development projects



Conclusion

Hand-coding XBRL functionality into an application can be an adequate method if the taxonomies never change. In these few, static cases, software developers might be able to create a basic level of functionality without using processing engine. But even with simple XBRL programming, IT management should evaluate how much time is being spent to create and maintain that code, the efforts required to ensure compliance with a changing taxonomy and evolving XBRL specifications, and what functionality is lacking by not using an advanced XBRL processing engine.

To fully leverage the power and functionality provided by XBRL, it is best to utilize an XBRL processing engine. By allowing an XBRL processing engine to do the creation, validation, and collection of documents, developers don't have to create or continue to support new application features as they can leverage XBRL. In addition, they can exchange those business rules with other organizations as they are in a global, open standard format, not the company's internal proprietary format.

The decision on when to leverage an XBRL processing engine should be based on the level of XBRL support the application intends to provide. Enterprises may not need full support for all XBRL implementations, but when they do need it – and don't have it – they could be misjudging the full potential of XBRL and the productivity benefits of using a fully conforming XBRL processing engine. For more information on the UBmatrix Processing Engine or other UBmatrix solutions, visit www.ubmatrix.com. For inquires, please send an email to info@ubmatrix.com.

UBmatrix, Inc.

333 Twin Dolphin Drive
 Redwood City, CA 94065
 USA
 Tel: +1.650.264.4510
 Fax: +1.650.264.4515
 E-mail: info@ubmatrix.com