



The Value of XBRL in an XML World

XBRL offers enhanced data quality, comparability, and transparency in information exchange

Executive Summary

This white paper is intended for those familiar with the use of XML in today's sophisticated business reporting systems. The goal of this white paper is to show how XML-based systems can leverage XBRL, which provides key additional benefits for information exchange applications.

Table of Contents

Introduction	1
Key XBRL Benefits	2
XBRL Leverages XML	8
From XML to XBRL	13
Conclusion.....	14

Introduction

Today, the ability to exchange information effectively is more critical than ever. In the growing complexity of global economics, demand for data quality and transparency are forefront initiatives of government and corporate entities.

Traditional information exchange systems must evolve to meet these new data quality and transparency initiatives, yet remain flexible in the requirements to enrich information models, write data quality rules, and identify new metrics for data analysis with minimal system impact.

Many applications and information models that are well adapted to information exchange have been implemented in XML. At the same time, developers of these applications and information models are discovering XBRL (eXtensible Business Reporting Language) is particularly effective in the realm of addressing issues involving data quality, timeliness, and comparability in business reporting. Current XBRL-based systems have demonstrated greater flexibility in data modeling, collection, and analysis. XBRL-based systems have been especially popular in financial reporting sectors such as banking, securities, stock exchanges, and credit risk assessment. Regulators in several capital markets, including China, Japan, and the United States, have already mandated XBRL-based reporting along with regulators in banking markets in Europe and country-wide reporting efforts in the Netherlands, the UK, and Australia.

Companies that are already using XML or are considering moving to an XML information model should strongly consider XBRL. Migrating from XML to XBRL is not a redesign; rather, XBRL complements the existing XML data model by providing key features such as content validation, semantic expression, extensibility, and improved rendering.

This paper offers readers the opportunity to gain a better understanding of XBRL by providing an overview of the benefits of XBRL in information exchange by describing the similarities and differences between XBRL and XML; and concludes with some thoughts on how an information exchange system currently implemented in XML could leverage the benefits of XBRL.

Key XBRL Benefits

As an XML-based information model, XBRL shares many similarities with XML. However XBRL provides additional benefits to the information exchange application developer.

XBRL Expresses Richer Business Semantics

XBRL's fundamental goal is to express business meaning by providing semantics such as concepts and multiple key relationships between concepts. XML provides the syntax for content and a single hierarchical relationship. XBRL makes it possible to express additional specific domain knowledge around these concepts. For example, one can express simple mathematical relationships (e.g. Shareholder Equity = Assets – Liabilities), more complicated mathematical relationships (e.g. assert whether Non Performing Loans > 20% of Total Portfolio) or presentation information (e.g. this report consists of these concepts presented in a specific ordering). The key benefit is that the semantics are captured in the information model, not in software that is deeply embedded in the system.

Real world usage - Expressing Richer Business Semantics

IFRS (International Financial Reporting Standards), US-GAAP (U.S. Generally Accepted Accounting Principles), EDINET (Financial Services Agency, Japan) and the BASEL II Accords are widely used reporting standards that have been expressed in XBRL. The XBRL taxonomies (how concepts and their relationships are expressed in XBRL) for these reporting standards encapsulate not only the concepts reported, but also the principles and regulations in the respective standards, thus allowing automated processes to validate and assess the quality of incoming data.

XBRL Can Validate Business Semantics of Content

XBRL documents can be consumed and syntactically validated by an XML parser. Because XBRL is expressed in XSD and XML files, an XML validator can catch errors and violations of XSD constraints; however, an XBRL processing engine provides further data validation, ensuring content meets constraints set forth by XBRL validation rules and XBRL Formulae (in addition to XML validation).

```
<xs:element id='us-gAAP_Assets'  
  name='Assets'  
  nillable='true'  
  substitutionGroup='xbrli:item'  
  type='xbrli:monetaryItemType'  
  xbrli:balance='debit'  
  xbrli:periodType='instant' />
```

Figure 1

For example, the <xs:element> tag in Figure 1 ascribes additional XBRL semantics to the concept of 'Assets' against which data can be validated and quality checked:

- The value must be a monetary unit of measure (dataType=xbri:monetaryItemType)
- The value is a debit value in calculation relationships (xbri:balance=debit)
- The concept must be associated with a context that has a single date declaration, not a range of dates (xbri:periodType=instant)

The debit (or credit) constraint dictates how the relationship is modeled in the taxonomy and validation occurs in both the instance document and the taxonomy.

Taxonomy linkbases define relationships between concepts or additional information about concepts), providing additional semantic information for validation, such as:

- Total Assets = Current Assets + NonCurrent Assets – modeled in the calculation linkbase. Mathematical relationships are automatically verified whenever all values are reported in an instance. For example, if Total Assets=100, Current Assets=30, and NonCurrent Assets=80, the relationship flags a calculation error.
- Sales must be associated with a sales region –modeled in the definition linkbase as dimensional information. The Regions dimension contains a set of explicit values (e.g. America, Europe, Asia) and any reported sales fact must be associated with a region.
- If Merger=true, then Merger Description must be reported – modeled in the formula linkbase. Formula linkbases contain validation rules that can be used to generate new information from existing values. Formulas are useful for enforcing filing requirements, trend analysis, and data comparison. Formulas can also use external sources to gather or generate information.

These standard XBRL validation rules are modeled once in a taxonomy. The XBRL processing engine validates not only the syntax of an instance document, but applies the validation rules built into the taxonomy against the data and returns errors, warnings, and/or informational text.

Real world usage - Validating Business Semantics of Content

The Federal Financial Institutions Examination Council (FFIEC) comprises three federal agencies, Federal Deposit Insurance Corporation (FDIC), the Federal Reserve Board (FRB), and the Office of the Comptroller of the Currency (OCC). These agencies are required to analyze and validate quarterly reports

comprising more than 3,000 data elements for quality and consistency from more than 8,300 financial institutions.

In the quarter ending March 2003, more than 17,000 errors and inconsistencies were identified and the full reporting cycle for collecting, validating, processing, and publishing data was 65 days.

In 2004, the FFIEC began to implement an XBRL-driven Call Report Modernization project to enable a more flexible and timely publication of information for market consumption. They defined an XBRL taxonomy that modeled the report, capturing all 3,000+ data elements and defining 1,800 business rules to enforce data validity and consistency. Business rules articulate data requirements in a machine-readable manner, providing automated warnings to filers of potential errors and inconsistencies.

With the new XBRL system in place, the FFIEC reduced the reporting cycle from 65 days to 2 days, and errors from 17,000 to 0.

XBRL Provides Organized, Prescriptive Extensibility

One of XML's greatest strengths is its extensibility, but too much flexibility introduces a factor of unpredictability. Systems built to handle one XML schema may have difficulty handling extensions, and XML extensions (often) require software modifications.

XBRL is designed to be extended in a prescriptive, predictable manner. All elements are defined as top-level, so it is unnecessary to modify parent-child hierarchies; and element attributes cannot be modified by an extending taxonomy; rather, the extending taxonomy must create a new element with a new data type.

Briefly, an XBRL taxonomy extension can be summarized as:

- Adding new elements and their relationships to other elements
- Adding/modifying/removing relationships between elements
- Modifying element labels and references, such as for localization

Thus, all XBRL taxonomies have the same predictable "shape" that can be handled by any generic XBRL processor without software modification.

Real world usage - Organized, Prescriptive Extensibility

Since 2005, the International Accounting Standards Board (IASB) has developed and maintained an XBRL taxonomy to represent the International Financial Reporting Standards (IFRS). The intent of the IFRS taxonomy is to provide companies, regulators, investors, and other users of IFRS-based reports easier comparison of financial data.

The Central European Banking Supervisors (CEBS) in turn created its Financial Reporting Framework (FINREP taxonomy) as an extension of the IFRS taxonomy. While leveraging as much as the IFRS content as possible, the FINREP taxonomy includes additional concepts, introduces multi-dimensionality, and creates new relationship hierarchies. FINREP has been distributed to all European countries that have decided to use XBRL for financial reporting. Currently, France, Belgium, Norway, and Poland, have been firm adopters of FINREP, and each has developed its own localized extension taxonomy which includes localization of labels and references.

All the regulators in the various countries use commercial off-the-shelf processing engines and have not written any additional code to handle the country-specific taxonomy extensions.

XBRL Provides a Multidimensional Model

Once the reported data has been validated, business reporting systems often migrate information to data warehouses and business intelligence applications where the data is modeled in multidimensional cubes (OLAP cubes) for benchmarking, trend analysis, data comparison, and other analytics.

XBRL's multidimensional modeling capabilities provide a seamless reporting process. An XBRL taxonomy is often used to model hypercubes consisting of measures and dimensions exemplified in US-GAAP, IFRS, FINREP, and COREP taxonomies. A multi-dimensional XBRL taxonomy allows both submitters and receivers to leverage the benefits of a dimensional model to correlate the information in their own internal systems. Dimensional data can be directly queried from XBRL in realtime without the need for a separate data warehouse.

For example, Sales is the **primary item** that is associated with a **hypercube** containing a single **dimension** called Regions. The dimension contains the **domain** of Total Regions that consists of three **domain members**: Americas, Europe, and Asia. This dimensional modeling information is defined in the definition linkbase. The instance document retains its flat shape. Dimensional information enriches the context, and the fact values are unchanged.

In the instance document, a fact value with dimensions looks like:

```
<ci:Sales contextRef="D-2008-Europe" unitRef="U-Monetary" decimals="INF">11901000</ci:Sales>
```

Figure 2

Its corresponding context in the instance document looks like:

```
<context id="D-2008-Europe">
  <entity>
    <identifier scheme="http://www.SampleCompany.com">SAMP</identifier>
    <segment>
      <xbrldi:explicitMember dimension="company:Regions">company:Europe</xbrldi:explicitMember>
    </segment>
  </entity>
  <period>
    <startDate>2008-01-01</startDate>
    <endDate>2008-12-31</endDate>
  </period>
</context>
```

Figure 3

The context includes a <segment> tag that shows the fact value belongs to the Europe member of the Regions dimension for the period 2008.

Real world usage - Multidimensional Model

The U.S. Securities and Exchange Commission mandates that public companies use the US GAAP taxonomy for reporting financial information in the 10-Q and 10-K submission. The US GAAP taxonomy supports the use of dimensions. In the case of Microsoft, dimensions are used report the revenue and income for its divisions.

Welcome !!		MSFT 2008-09-30	
EDD			
Segment Reporting Information, Revenue, Total		1,814	1,929
Segment Reporting Information, Operating Income (Loss), Total		134	139
MBD			
Segment Reporting Information, Revenue, Total		4,947	4,113
Segment Reporting Information, Operating Income (Loss), Total		3,260	2,714
OSB			
Segment Reporting Information, Revenue, Total		770	665
Segment Reporting Information, Operating Income (Loss), Total		(521)	(268)
ST			
Segment Reporting Information, Revenue, Total		3,405	2,903
Segment Reporting Information, Operating Income (Loss), Total		1,081	880

Figure 4

The domain members of the dimension include EDD (Entertainment and Devices Division), MBD (Microsoft Business Division), OSB (Online Services Business) and ST (Server and Tools).

XBRL Provides Additional Rendering Capabilities

Although machine-readable data models are an optimal way to submit, process, and validate data, they typically do not offer basic user readability. Often information exchange systems require rendering capabilities to display data in human readable format (such as HTML, Forms, PDF, or Excel). Submitters need forms or spreadsheets to prepare data through a user-friendly interface. Reviewers (administrators, analysts) need submitted data to be rendered into HTML, PDF, or Excel. These rendering requirements are sent to developers who

fall into the trap of hard-coding labels, ordering rows, and columns directly into their tools, leading to extremely rigid frameworks with high maintenance costs.

XML developers know a better approach is to use a transformation language to convert from a machine-readable language (such as XBRL or XML) to a user-friendly format. Transformation retains the content model as the source of data, and XSLT can be used to generate alternative formats as needed.

XBRL supports XSLT, with an additional benefit of leveraging the taxonomy's presentation linkbase for rendering purposes. This linkbase may be used to generate a single or multiple reports (e.g. Balance Sheet or Income Statement). Often, the presentation hierarchy is already a direct representation of a report, and is the vertical axis for rendering labeling information and documentation regarding each concept in the taxonomy. This information can also be leveraged by a report generation tool.

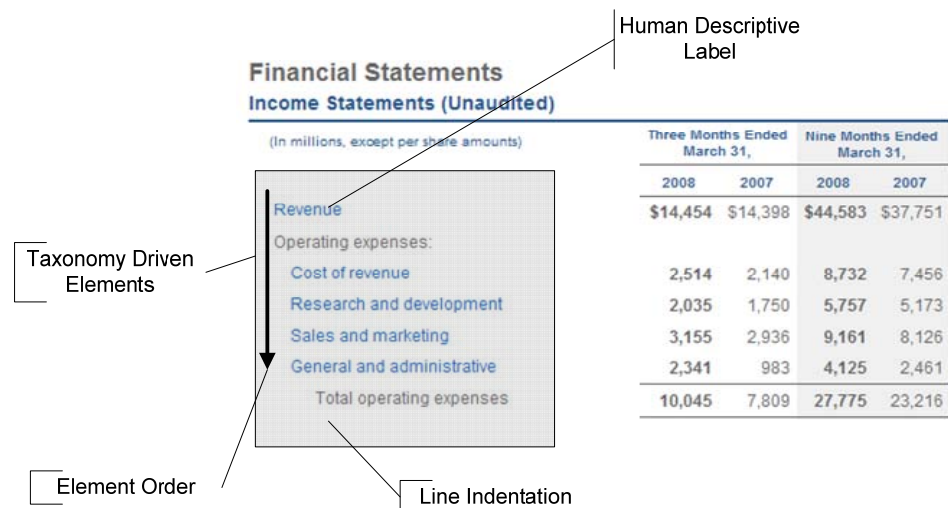


Figure 5

A single XSLT written for XBRL report generation can be reused across several taxonomies. Because it uses generic functions, such as getLabel(), the XSLT query requires minimum maintenance. To add a new row to a table or to modify a label, the taxonomy changes, not the XSLT. Thus any business user can design an XBRL report, using the same XSLT file without developer assistance once the initial XSLT is developed.

Real world usage - Additional Rendering Capabilities

In March 2008, the Italian Chamber of Commerce mandated that Italian companies file annual financial statements in XBRL. More than 850,000 companies were affected by this change, and Wolter Kluwers Italy (WKI), a leading provider of accounting software, needed an XBRL solution to assist those companies with the new filing requirements. By integrating the UBmatrix XBRL processing engine into WKI Balance Sheet software, clients were able to

generate 100% error free submissions using on-the-fly XBRL validation. Because the instance document and taxonomy remain in memory in the processing engine, filers were able to capture a snapshot of their XBRL submissions in a PDF report at any point in the process. A sample PDF snapshot is shown in Figure 6.

Stato patrimoniale (schema civilistico)	31/12/2004	31/12/2003
Attivo		
B) Immobilizzazioni		
I - Immobilizzazioni immateriali		
2) costi di ricerca, di sviluppo e di pubblicità	314	629
3) diritti di brevetto industriale e diritti di utilizzazione delle opere dell'ingegno	1.870	9.403
7) altre	4.248	5.137
Totale immobilizzazioni immateriali	6.432	15.169
II - Immobilizzazioni materiali		
1) terreni e fabbricati	629.325	631.145
2) impianti e macchinario	82.714	260.607
3) attrezzature industriali e commerciali	52.264	14.072
4) altri beni	60.151	85.154
5) immobilizzazioni in corso e acconti	558	558
Totale immobilizzazioni materiali	825.012	991.536
III - Immobilizzazioni finanziarie		
1) partecipazioni in		

Figure 6

XBRL Leverages XML

This section takes a closer look at the components of XBRL and discusses how XBRL both leverages and distinguishes itself from XML.

The Content Model

XBRL and XML both allow developers to create and use a content model that defines data elements, their attributes and relationships, and surrounding rules. Data must comply with the set of constraints in this content model. In XML, this model is called a **schema definition**, often generated as an XSD file. In XBRL, this model is called a **taxonomy**, represented by one or more XSD files and zero or more XML files.

Elements

In both technologies, an XSD file is used to define data elements and their properties. Figure 7 shows an XBRL element, or **taxonomy concept**, which is an XML schema element that has extra attributes. The 'xbrli' namespace prefix

indicates that this element uses XBRL-schema defined attribute names and values.

```
<xs:element id='us-gaap_Assets'  
            name='Assets'  
            nillable='true'  
            substitutionGroup='xbrli:item'  
            type='xbrli:monetaryItemType'  
            xbrli:balance='debit'  
            xbrli:periodType='instant' />
```

Figure 7

An XBRL processor uses XBRL schema attributes during document validation (see previous section - **XBRL Can Validate Business Semantics of Content**).

Relationships

XML and XBRL begin to differ in regards to the relationships in the content model. An XML schema defines the hierarchy that associates elements within one another. This would tell us, for example, that 'Total Assets' is made up of 'Current Assets' and 'Non-Current Assets'. In XBRL, the XSD file defines all elements as top-level elements with no relationships.

In XBRL an element can have multiple types of relationships with different semantic meanings. For example, *Total Assets* is the sum of *Current Assets* and *Non-Current Assets*, so it would make sense to model *Total Assets* as a parent with two child nodes. It would also make sense to model these as three sibling nodes, with *Total Assets* at the bottom of the balance sheet for generating a report. In XBRL, these two perspectives are expressed as the Presentation View and Calculation View, as shown in Figure 8.

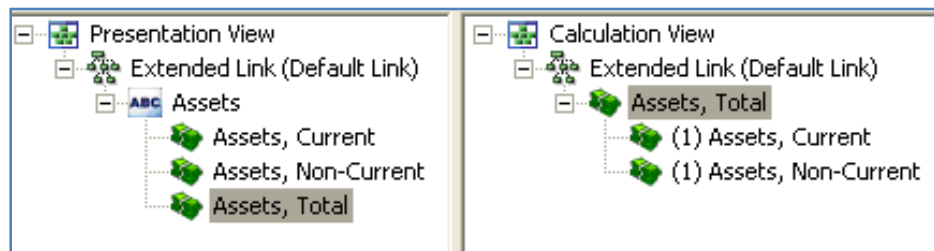


Figure 8

In XBRL, relationships (among elements) are stored in XML files that are referenced directly by the taxonomy XSD file. The XSD file is referred to as the **taxonomy entry-point** and the corresponding XML files are referred to as **linkbases**, because they use XLink language notation to describe the relationships between concepts. Together, the XSD and XML files constitute an XBRL taxonomy, or content model.

XBRL supports many types of linkbases, including:

- **Presentation** – describes a logical hierarchy of concepts that a user would recognize and understand. This enhances readability and also provides an opportunity for automated report or form generation, if desired.
- **Calculation** – describes the mathematical relationships among elements, particularly additions and subtractions that roll up to subtotals and final total amounts. For more complex calculations, such as ratios and multiplications, see the formula linkbase.
- **Definition** – models multi-dimensional data, as described in the XBRL Dimensions 1.0 specification.
- **Label** – A concept may be assigned to multiple labels, each with its own label type. For example, a concept could have a standard label, a documentation label, and/or custom labels, as defined by the user. All labels are localizable via the *xml:lang* attribute. Multi-lingual Label Linkbases allow automated multi-lingual translation
- **Reference** – A concept may also be associated with zero or more external references. For example, Total Assets may have a relationship to a reference called “FASB Statement of Financial Accounting Concepts, Ref. No. 6, Paragraph 25.”
- **Formula** – contains user-defined business rules used for data validation and analysis and to gather statistical data.
- **Generic** – enables users to define custom relationship types in a taxonomy. A formula linkbase is an example of a generic linkbase that has recently been adopted as an official XBRL specification.

Tradeoffs in Complexity

With this variety of possible relationships, it’s easy to imagine how an XBRL taxonomy can be larger and more complex than its XML counterpart. After all, some taxonomies have thousands of concepts, each with multiple relationships. XML may have the same thousands of concepts, but it is limited to one type of relationship to handle those concepts: parent-to-child relationships in a single tree structure.

At a minimum, a valid XBRL taxonomy consists of a single XSD file. Additional XSD files and all linkbases, or relationships, are optional.

Why use linkbases?

Linkbases are useful for two reasons:

1. **Ease of use** – Presentation, label, and reference linkbases provide (optional) information to add meaning to the data model. For

taxonomies that are shared among hundreds of consumers, it's critical that the data model is conveyed in a straightforward and readable manner. Financial information can be displayed in a presentation view, just as it would appear in a financial statement--familiar territory for users.

Labels provide descriptions of concepts that can be customized or localized.

References link to external sources that further clarify meaning.

2. **Validation** - The more detailed the information in the data model, the greater the opportunities to ensure data quality. Calculation, definition, and formula linkbases enable taxonomy authors to model domain knowledge in equations that can be processed by a computer. For example, the *Total Assets = Current Assets + Non-Current Assets* calculation in Figure 8 allows any XBRL-aware system to validate the fact values of these concepts.

Thus, XBRL taxonomies can be as simple or complex as the information you want to capture in your data model. XBRL users consider the taxonomy the "single source of truth," a single location that encapsulates domain knowledge, data quality checks, business rules, rendering information, and any other desired information.

Although these types of relationships can be modeled in XML, XBRL provides a standardized language for modeling the relationships and software vendors offer supporting implementations.

Instance Data

In XML, the XSD file is the content model and the XML file contains the content. In XBRL, a taxonomy (XSD and XML files) is the content model and an **instance document** (XML or XBRL file) contains the content and a reference back to the taxonomy entry-point.

Fact Values

Because XBRL is designed for business reporting, an instance document must follow the structure that is defined in the taxonomy. To report Assets (defined in Figure 9) of \$1,000,000 for the year the **fact value** the instance document looks like this:

```
<us-gaap:Assets contextRef="ABC-2008-Instant"
                decimals="-2"
                unitRef="USD">1000000</us-gaap:Assets>
```

Figure 9

This looks like XML, with a start and end tag, some attributes, and some content. However, XBRL has no element-specific attributes.

Let's take a look at a larger snippet of an instance document:

```
<us-gaap:Assets contextRef="ABC-2008-Instant" decimals="-2" unitRef="USD">1000000</us-gaap:Assets>
<us-gaap:AssetsCurrent contextRef="ABC-2008-Instant" decimals="-2" unitRef="USD">700000</us-gaap:AssetsCurrent>
<us-gaap:AssetsNoncurrent contextRef="ABC-2008-Instant" decimals="-2" unitRef="USD">300000</us-gaap:AssetsNoncur
<us-gaap:StockholdersEquity contextRef="ABC-2008-Instant" decimals="-2" unitRef="USD">2850</us-gaap:StockholdersE
<us-gaap:StockBasedCompensation contextRef="ABC-2008-Duration" decimals="-2" unitRef="USD">10430</us-gaap:StockB
<us-gaap:ScheduleOfGoodwillBySegmentTextBlock contextRef="ABC-2008-Duration">
  The changes in the carrying amount of goodwill, by reportable segment, for the year ended December 31, 2008,
</us-gaap:ScheduleOfGoodwillBySegmentTextBlock>
```

Figure 10

Notice the similarity in attribute names and values and that elements are not nested. As previously described, the relationships are defined in the taxonomy's linkbases.

Contexts and Units

Every fact value must use a *contextRef* attribute, and every numeric fact value must also include a *unitRef* attribute and *precision* attribute (often thought of as *decimals*). The *contextRef* and *unitRef* attributes refer to the **contexts** and **units**, which are defined separately in the instance document.

Here's what a context could look like in an instance document:

```
<context id="ABC-2008-Instant">
  <entity>
    <identifier scheme="http://www.sec.gov/CIK">CompanyABC</identifier>
  </entity>
  <period>
    <instant>2008-12-31</instant>
  </period>
</context>
```

Figure 11

Notice that the context *id* matches the *contextRef* from the fact value. This particular context tells us that the information is for *Company ABC* and the reported value is valid for the date *December 31, 2008*.

Here's what a unit could look like in an instance document:

```
<unit id="USD">
  <measure>iso4217:USD</measure>
</unit>
```

Figure 12

Notice that the unit *id* matches the *unitRef* from the fact value. This particular unit tells us that the reported value is in US Dollars, as defined by the ISO standard for currency codes.

Thus, putting this information together, Figure 9 tells us that *Company ABC* is reporting *Assets = 1,000,000* in *US dollars* on *December 31, 2008*, with a precision in the *hundreds* (`decimals="-2"`).

Tradeoffs in Complexity

Unlike a typical XML document, an XBRL instance document has a very well-defined structure or “shape” made of three types of components: fact values, contexts, and units. Regardless of which taxonomy (content model) is being used, the instance document is very consistent in its representation. This consistency allows for document rendering, parsing, and ease of querying in a taxonomy-agnostic manner.

Consider a query, such as “Find all companies that reported Assets greater than \$1,000,000USD since 2007.” This type of information is extremely simple to retrieve because all XBRL elements are top-level tags. In addition, this query may be reused for any taxonomy. To make this query generic, simply parameterize the value 1000000, the monetary unit ‘USD’, the concept ‘Assets’, and the year ‘2007.’ The underlying query implementation remains valid for all instance documents, regardless of the taxonomy.

Alternatively, in regular XML files, the structure of the document is extremely dependent on the constraints set forth in the taxonomy. The element hierarchy may consist of any number of nested levels and the attribute names and values are dependent on the element being declared. Thus, it is difficult to provide an taxonomy-agnostic manner of generating XML files and virtually impossible to create meaningful queries that can be used with different XML files.

From XML to XBRL

If you already have an XML data model, creating an XBRL taxonomy is quite simple. The basic things to consider are:

- Elements and element attributes such as ‘name’, ‘id’, ‘substitutionGroup’, ‘type’ are modeled the same way in an XSD file for both XML and XBRL.
- XBRL requires some additional element attributes. For example `xbrli:period` is an attribute to indicate the period of time that an element is valid.
- In XBRL, certain element-specific attributes are not used. In the simplest case, these attributes become additional elements.
- Element relationships in an XML schema reside in an XBRL linkbase. Moving these to a presentation linkbase is a typical solution.

With these changes, an XML schema has been transformed to an XBRL taxonomy and can be run through an XBRL processing engine for validation.

However the real benefits of XBRL come from extending this basic model, enabling users to define relationships in linkbases that apply domain knowledge through calculations, presentations, formulas and dimensions

The more domain knowledge captured in an XBRL taxonomy, the more benefits that can be gained.

Conclusion

As an XML-based technology, XBRL fits well into many XML processing pipelines. In particular, projects that demonstrate any of the following characteristics are often good candidates for XBRL:

- ***The information model is complex and somewhat flexible, or requires periodic changes.***

An XBRL taxonomy enables even the most complex data models to be represented, offering extensibility such that domain knowledge can be shared and customized for specific usages.

- ***Information components have complex relationships that must be enforced in an automated, efficient manner.***

An XBRL taxonomy offers the flexibility for users to express multiple types of relationships among components. Relationships may be used to model presentation formatting, mathematical operations, multidimensional characteristics, business rules, and other custom associations.

- ***Management and control of metadata needs to be handled at the level of a business user.***

XBRL allows business users to model information, describe data quality requirements, flag potential issues, and extrapolate new data for further analysis. Because information is encapsulated in the taxonomy, business users can make modifications whenever necessary with minimal to zero changes to the underlying application or system code.

- ***Information is presented in many different ways to many different users.***

Reports can generated directly from XBRL data into any desired rendering format (including HTML, Excel, and PDF). Predefined reports can be described using the XBRL taxonomy's presentation linkbase as a guide, and ad hoc reporting is also available. An XBRL database makes it easy to do trend analysis and data comparisons on-the-fly using real-time data in a performant and scalable manner.

For more information on the UBmatrix software solutions, visit <http://www.ubmatrix.com>. For inquiries, 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

August 2009