Filing financial statements in XBRL: How-to, lessons learned, and best practices

Mandatory compliance for XBRL-based reporting is fast approaching

Skill Level: Introductory

David Newman (dnewman0@us.ibm.com)
Software Engineer
IBM

Dean Ritz (xbrl@deanritz.com)
Independent Consultant

Murali Vridhachalam (mural@us.ibm.com)
IT Architect
IBM

18 Nov 2008

In the past few years, eXtensible Business Reporting Language (XBRL) has emerged to meet increased regulatory and transparency requirements for financial reporting. The global connectivity of the Internet has encouraged the rapid development of XBRL standards. Effective December 15, 2008, the US Securities and Exchange Commission (SEC) will require companies with more than $5 billion in market capitalization to file their financial statements in XBRL. Over the following two years, all publicly traded companies in the US will be required to file using XBRL. In this article, learn the fundamentals of XBRL, the steps in the filing process, and lessons from an actual filing with the SEC.

Introduction

Did you know?
IBM® was one of the first companies to file its financial statements

using XBRL as part of the SEC Voluntary Filing Program. The authors worked with multiple filings of IBM financial statements.

XBRL provides an XML-based data format that enables electronic communication of business and financial information. With XBRL, every piece of information is tagged using standardized vocabulary defined in an XBRL taxonomy. For example, *net revenue* would have its own unique XBRL element.

XBRL International is a nonprofit organization that maintains the XBRL standard. More than 500 companies and government agencies are members, and the organization operates with more than 20 XBRL jurisdictions around the world. There are major government-sponsored XBRL initiatives in the US, Australia, Netherlands, India and many other countries.

In this article, learn about XBRL terminology and the steps for preparing an XBRL filing. From their experience, the authors include both general and specific lessons learned, and best practices, for creating XBRL format filings.

Differences between XML and XBRL

If you're familiar with XML, you might wonder what XBRL has that XML does not have. While it is true that XBRL is built on XML, and that all XBRL documents must also be XML valid, there are major differences between the two. The most important conceptual distinction is that XBRL provides a way to express and validate semantics. For example, *Assets* = *Equity* + *Liabilities* can be expressed using XBRL but not using native XML.

XBRL accommodates change better than XML. For example, XBRL separates the schema from the data. XBRL instance (data) documents are simple, containing normalized data immune to their sequence, and without hierarchy. Users can make changes to the XBRL taxonomy, such as changing parent-child relationships or adding new children, without having to reformat instance documents. Similarly, data in the instance document may be reordered without having an impact on the taxonomy. Normalized data also simplifies database storage.

Another unique feature of XBRL is its native support of multidimensional modeling, which is very similar to Microsoft® Excel pivot tables. Consider the situation where you want a metadata model to support segmentation of product sales by country and also by product line. Assuming a simple example of product sales by three countries (US, Canada, and Mexico) and two product lines (software and hardware), a pure element-based approach would result in 12 elements:

- sales_US_software
- sales_US_hardware

- sales_US_allProducts
- sales_Canada_software
- sales_Canada_hardware
- sales_Canada_allProducts
- sales_Mexico_software
- sales_Mexico_hardware
- sales_Mexico_allProducts
- sales_allCountries_software
- sales_allCountries_hardware
- sales_allCountries_allProducts

Adding another product to the product line adds another four elements. Add another line item, such as cost of goods sold, and that one additional line item would result in another twelve elements. Such an approach quickly gets out of control, and is extremely frustrating to model and manage.

Using XBRL dimensions, there would be the single element of sales, along with the two dimensions of country and product line, as shown in Figure 1. The country dimensions would have children for US, Canada, and Mexico; the product line dimensions would have children for software and hardware.

Figure 1. Country and product line dimensions easily reused for line items

Dimensions

Line Item	Country	Product Line
Sales	Mexico	Hardware
Sales	Mexico	Software
Sales	Mexico	All
Sales	Canada	Hardware
Sales	Canada	Software
Sales	Canada	All
Sales	US	Hardware
Sales	US	Software
Sales	US	All
Sales	All	Hardware
Sales	All	Software
Sales	All	All

The modularity greatly simplifies building the metadata model; the dimensions of country and product line can be reused for other elements, such as cost of goods sold, gross revenue, net revenue, and so on. It also showcases the semantic relationships between the values reported and their segmentation. For the adventurous, XBRL dimensions also support some set operations. The US-GAAP taxonomy (UGT), IFRS, COREP, and FINREP taxonomies make extensive use of dimensions.

The last noteworthy distinction of XBRL is *validation*. While XML validation is limited to syntax, XBRL validation confirms semantic relationships between elements (as defined in the XBRL taxonomy). For example, an XBRL validation engine confirms that the value reported for assets does, in fact, equal the sum of the values reported for equity and liabilities.

XBRL Formulas, a recent addition to the XBRL standard, provides a portable notation for the validation of other, more complex relationships between elements. For example, the FDIC uses XBRL formulas to capture the expertise of its auditors so that reviews can be performed in seconds rather than hours. The nonprofit Microfinance Exchange uses XBRL validation and formulas to examine the financial reports of thousands of microfinance organizations around the world. Using XBRL, analyst work at Microfinance Exchange that once took hours is now performed in seconds.

XBRL terminology

Key XBRL terms are:

Concept
Describes a discrete piece of business information. For example, "Assets, Total" label, documentation, or reference. A concept that defines a business fact, such as "Assets, Total," in XBRL is sometimes called an *element*—a term inherited from XML.

Tuple
A special kind of data type because it binds together a set of concepts, like a row in a relational database where fields in a single row are related. It used to be common usage to associate concepts through tuples.
For example, annual reports require the disclosure of stock holdings of company officers. A tuple would be used to associate each officer's name with age, shares owned, and options held. The individual facts are meaningless without this association; you would not know which age went with which officer! Tuples may also contain other tuples.

Dimensions
XBRL dimensions have now taken the place of tuples for associating concepts. Tuples are no longer permitted in the UGT or in a company's filing taxonomy for two reasons:

- Tuples insert schema information into the instance document. Consequently, changes to tuple definitions in the taxonomy can "break" the instance document.
- Nested tuples are extremely awkward to modify, both in the taxonomy and in the instance document.

Dimensions do not have these faults, and they provide flexibility in data modeling that tuples don't offer.

Name
The name of a concept uniquely identifies it in its namespace. A concept also has an ID formed by the concatenation of the namespace prefix with the concept name. For example, an element named Assets in the UGT taxonomy with the prefix us-gaap has the name us-gaap_Assets. This provides for unique identification of an element within a large, networked taxonomy (such as UGT).

Relation
A connection between two concepts, accomplished using the xlink standard. The relation is always from one concept to another. It is directional, based on

xlink, with "from" and "to" as the endpoints. The name of the relation is its role. A concept may participate in many relations, such as a concept having multiple labels by language.
For example, first you would identify one endpoint of the relation. In this case, it is the concept for a value to be reported.

```
<loc xlink:type="locator";
      xlink:label="IBM_NetAssets";
      xlink:href="ibm.xsd#IBM_NetAssets" />
```

Next, define the labels in different languages. These are other endpoints of the relation.

```
<label xlink:type="resource"
      xlink:label="IBM_NetAssets_lbl"
      xml:lang="en-US"
      xlink:role="http://www.xbrl.org/2003/role/label" >Net
Assets</label>

<label xlink:type="resource"
      xlink:label="IBM_ActivoNeto_lbl"
      xml:lang="es"
      xlink:role="http://www.xbrl.org/2003/role/label">Activo
Neto</label>
```

Establish the relations by creating arcs between the concepts and the two labels.

```
<labelArc xlink:type="arc"
      xlink:arcrole="http://www.xbrl.org/2003/arcrole/concept-label"
      xlink:from="IBM_NetAssets"
      xlink:to="IBM_NetAssets_lbl" />

<labelArc xlink:type="arc"
      xlink:arcrole="http://www.xbrl.org/2003/arcrole/concept-label"
      xlink:from="IBM_NetAssets"
      xlink:to="IBM_ActivoNeto_lbl" />
```

Extended link

A collection of concepts organized for a specific purpose. It contains a network of relations between concepts, such as:
Assets, Total = Assets, Current + Assets, Noncurrent
There are two purposes for extended links:

- They make something big into something smaller and more modular.
- They isolate relations that might otherwise conflict. For example, there are two ways to calculate Trade Receivables, Net:

Trade Receivables,	Trade receivables,

Gross - Allowance for Bad Debt --- = Trade Receivables, Net	Net, Current + Trade receivables, Net, Noncurrent --- = Trade Receivables, Net

To capture both relations (because you want your taxonomy to be as comprehensive as possible), use extended links to isolate them from each other.

Presentation relation

Provides for a hierarchical organization of elements. The relations are from the parent to the child. These relations are stored in a presentation linkbase.

Calculation relation

Provides for summation organization of elements. These relations are stored in a calculation linkbase.

Label

A human-readable label for an element. Business users see labels. Taxonomy authors (taxonomists) see concept names. Labels have attributes for language and purpose. A label is assigned to a particular element through a relation established between the label and the element. These relations are stored in a label linkbase. Figure 2 shows an example.

Figure 2. The label arc links a label to an XBRL concept

Documentation

Describes the meaning of an element in human-readable terms. Consider it a special type of label—a label whose role is for documentation. Documentation should be clear enough for a business user. However, documentation is not authoritative and is not meant to offer an official or exhaustive definition. Instead, references are intended to provide authoritative documentation. Documentation relations are stored in a documentation linkbase.

Reference

Provides a pointer to authoritative literature for an element. For example, the authoritative references for UGT concepts point to SEC and Financial Accounting Standards Board (FASB) rules. These relations are stored in a reference linkbase.

Taxonomy

A collection of elements, relations, labels, references, documentation, and references to other taxonomies. It is like a dictionary or classification system. Discoverable taxonomy set (DTS) is used to define the complete network of all referenced taxonomies and their respective file-based components.

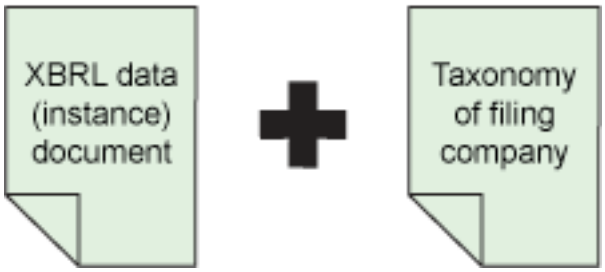
Preparing an XBRL filing

The XBRL filing process has a single, clear goal: To transform an *implicitly* structured financial report into a structured financial report.

Preparing a filing involves making associations between pieces of the stated and implicit data (usually in a spreadsheet or word processing document) and an explicit informational structure described in XBRL. The UGT provides a starting point. The company filing taxonomy (CFT) extends the UGT to meet company-specific reporting needs (new elements, new labels for existing elements, changes in the schema, and so on). For example, the CFT can override the standard UGT label of "Assets, Total" to match what the company uses on its reports, such as "Total Assets."

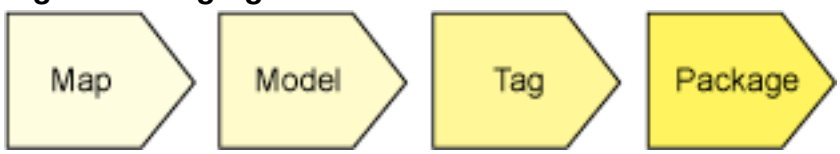
An XBRL filing is composed of two logical components: an XBRL taxonomy, which describes the metadata of the filing company's financial report (built on the UGT), and the XBRL instance document that references the taxonomy.

Figure 3. Two logical components of a filing



The ancillary pieces created during the XBRL filing process are various kinds of project documentation. The documentation is important to both the technical team creating the technical documents, and to the financial reporting specialist concerned with securing auditable information about the creation of a specific filing. Figure 4 shows a broad overview of the process.

Figure 4. Large-grain view



The map step documents the associations between your financial report and the UGT, including changes and additions. Consider this a description of the perfect

filing. The task then becomes implementing the map.

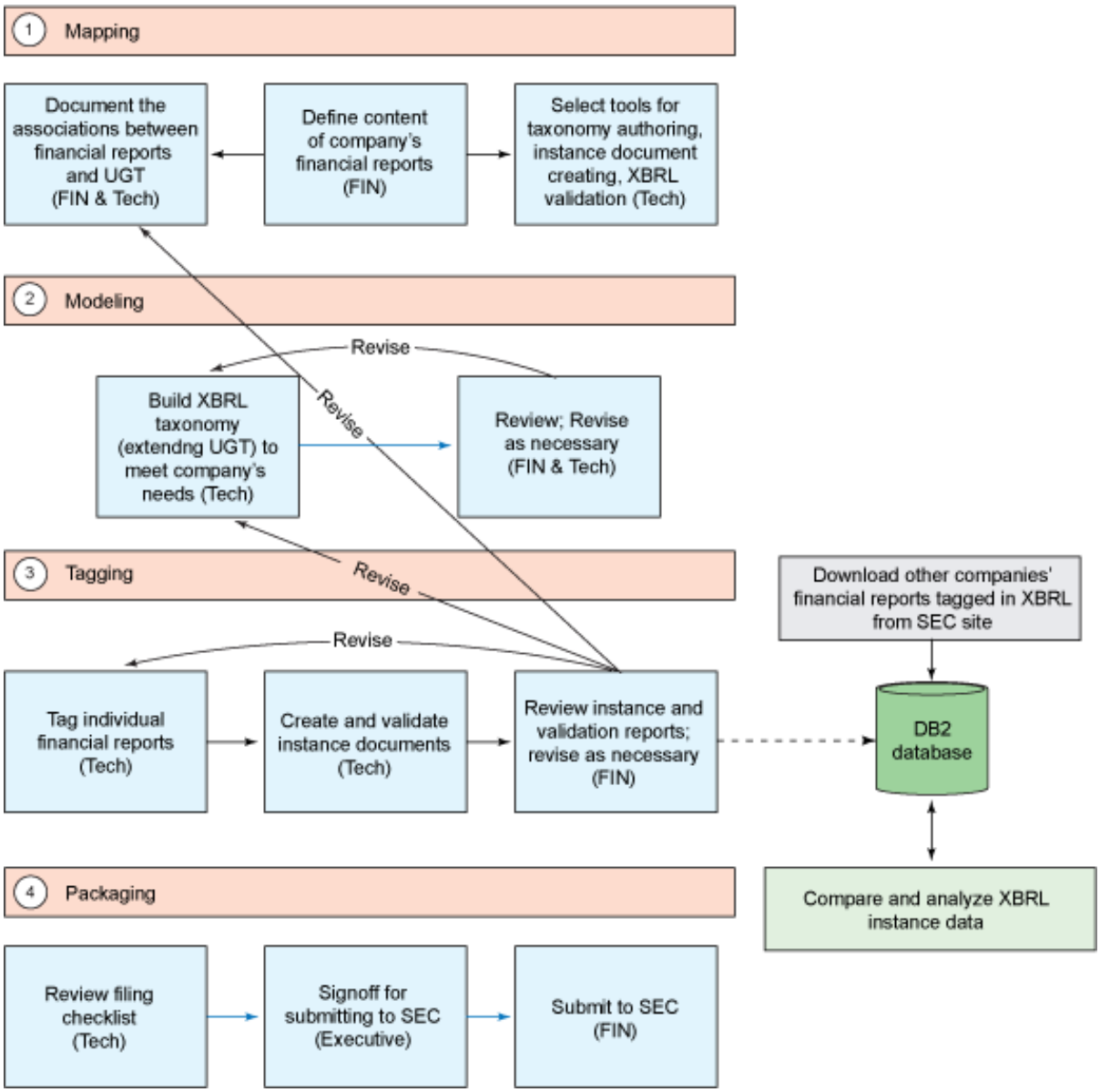
The model step implements your company filing taxonomy.

Tagging is the physical process of linking your financial facts to your company taxonomy.

During packaging, you create your filing package, complete your filing checklist, review the report with the client, and deliver it to the SEC or your financial printer.

Figure 5 shows the iterative nature of the entire filing process. It also shows an optional step where a company may use IBM DB2® 9 to store its XBRL-formatted financial reports, as well as reports from other companies downloaded from the SEC site to allow for comparisons and faster analysis of financials of various companies.

Figure 5. Detailed four-step process for filing



For the Q1 FY2008 filing by IBM, an XBRL descriptive table (XDT) was created to capture the mapping information. The XDT was based on the "neutral format table" work of Charles Hoffman, the acknowledged father of XBRL (see Resources). The neutral format table captures mapping information without regard to implementation technology. The XDT adds information necessary for XBRL implementation.

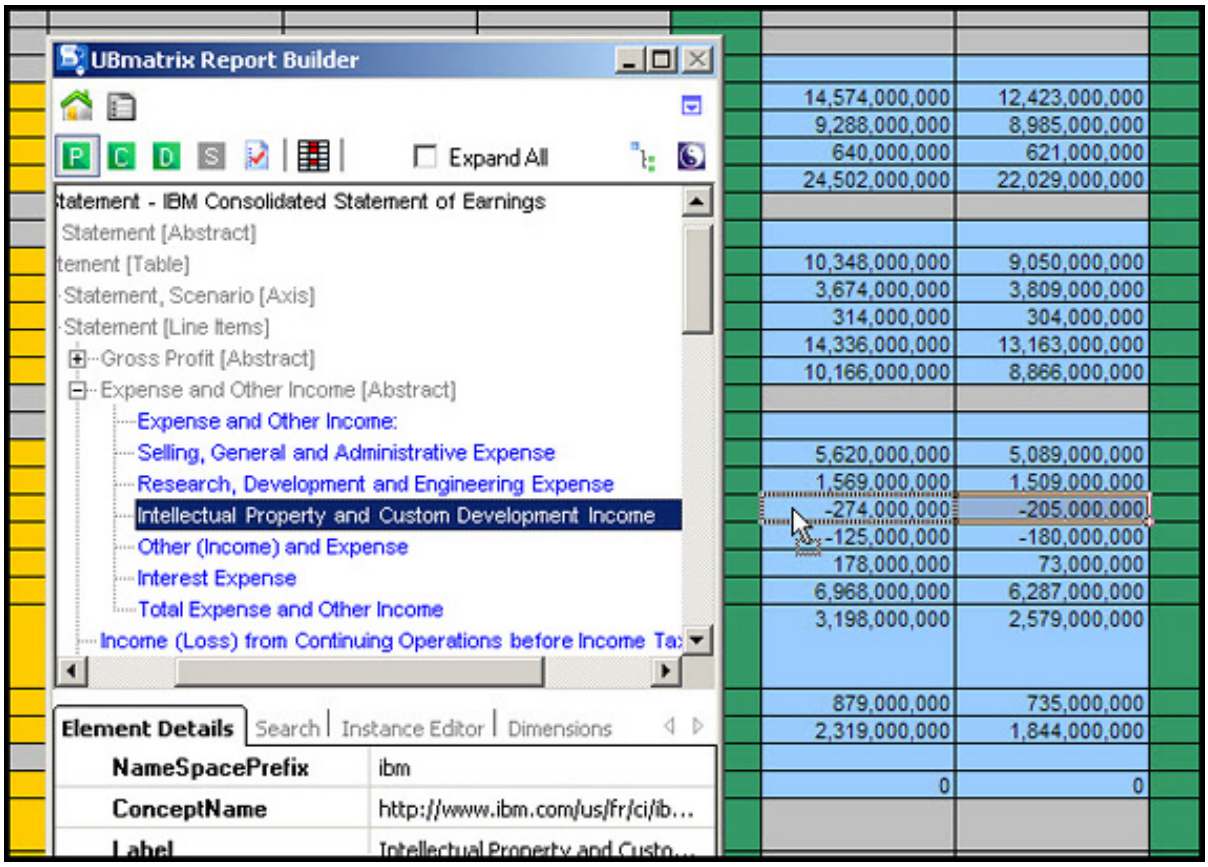
The XDT is modeled in a spreadsheet. It captures the information needed both for the taxonomy and also for the instance document. Figure 6 shows an example.

(See a larger version of Figure 6.)

Figure 6. XBRL descriptive table

The XDT is captured in an Excel spreadsheet. For the Q1 FY2008 filing, UBmatrix Report Builder was used to tag the data in the XDT spreadsheet to the CFT. Other tools to tag data include Dragon Tag by Rivet Software and Snappy Reports. A tagged cell is shown in Figure 7.

Figure 7. Tagging a cell with an XBRL concept

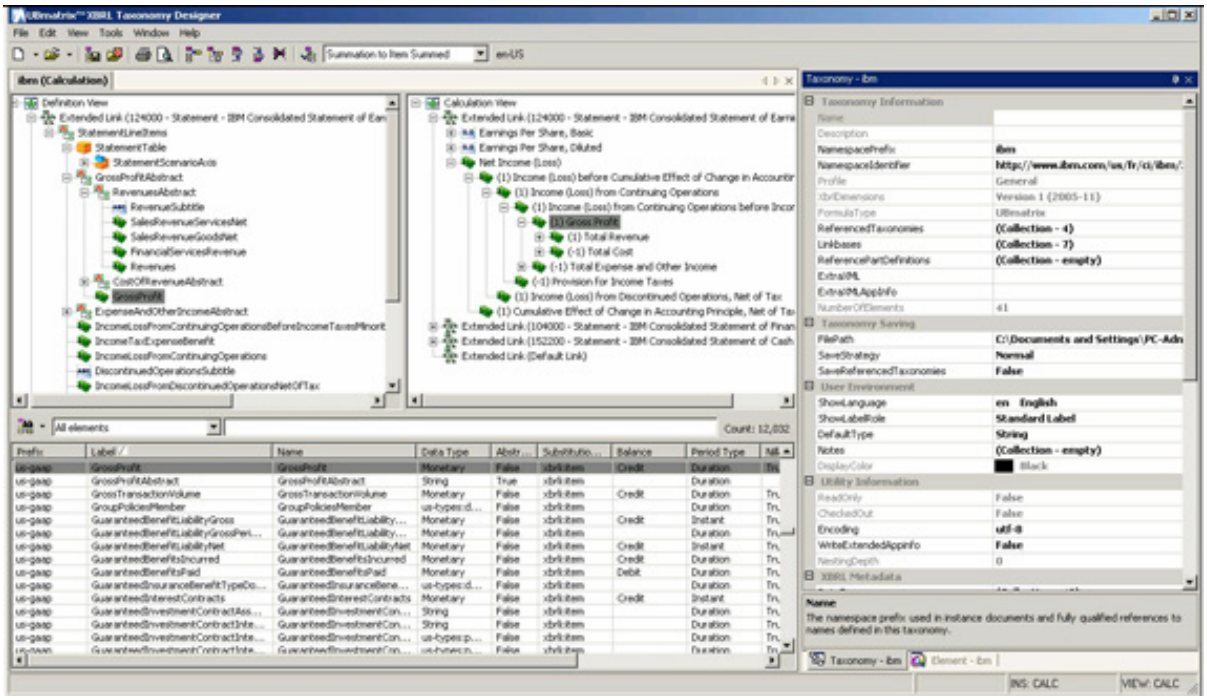


The XDT is organized by statements (for example, balance sheet, income statement). For each statement the XDT records the values to be reported, their labels, the name of each XBRL element, the reporting period, the decimal position for numeric values, and other information. As noted previously, this information documents the requirements for the CFT, which you subsequently implement in the modeling step.

Creating the CFT can be simple or complex, depending on the complexity of the information reported. It is relatively easy to model in XBRL the three standard financial statements associated with a 10Q (quarterly statement of financial position, statement of income, and statement of cash flow). A 10K (Annual Report) is much more complex because it includes the statement of shareholders' equity, which requires dimensional modeling. Modeling is performed in a taxonomy authoring tool, and there are now several of them on the market. For the IBM filings, UBmatrix Taxonomy Designer was used. Fujitsu XWand is another option.

(See a larger version of Figure 8.)

Figure 8. IBM Q1FY08 Taxonomy in UBMatrix Taxonomy Designer



Lessons learned from creating XBRL format filings for the SEC

For compliant and correct XBRL filings at the SEC, you should focus on acceptance criteria, lessons learned from experience, and best practices. The acceptance criteria, as defined by the SEC, are in the deceptively small document "Public Validation Criteria" (see Resources). XBRL.US provides a "Preparers Guide" (see Resources) that offers ideas for a workflow in creating a filing and some best practices.

However, experience demonstrates that experience is the best teacher! The authors have collective experience creating multiple filings for IBM and other large companies using three different versions of the UGT (2005, 2008 beta2, and release 1.0). This section outlines some of their lessons and best practices (all directed to creating SEC filings).

Table 1. General lessons learned and best practices

Lesson	Best practice
If you've ever tried to learn the folk guitar, you know you can play pretty quickly, even if badly. Creating an XBRL taxonomy is quite similar. It's easy to create a taxonomy, particularly a bad one. "Bad" means it fails to apply the architecture rules of the UGT, selects the wrong UGT concept for a financial figure, fails to accurately represent your company's financial reporting model, renders statements that do not match your other data	Have at least two people on the team, one an expert in XBRL and another in the financial reporting practices of the specific company. Ideally, each expert should know something about the other's area of expertise.

formats, and so on.	
It takes longer than you think. Even with the help of someone experienced in this process, and with a good process to follow, the preparation of a 10Q will take from 32 to 40 hours (not including training time).	Establish a reliable internal process. A reliable process is especially important because the SEC will soon require concurrent delivery in all data formats. Imagine the fallout for your company if it delays the release of its quarterly figures by a week or two because the XBRL filing is taking longer than anticipated. You should never have to start from scratch except on your first attempt. Make that first attempt while you still have time to make and learn from mistakes (before the end of 2008).
This is not a linear task. Each step involves iterations of testing and revision. The iterative process should be done for each financial statement. You might think you're near the end of the project only to discover (surprise!) that you've picked an incorrect UGT concept for a financial figure.	Plan for iterative development and lots of reviews.
Your filing is not perfect. Currently, there are many opportunities for human error, including the human-dependent steps in examining the quality of your taxonomy and instance documents. The service industry arising from the SEC mandate is slowly recognizing the challenges in detailed tagging of every piece of data in a financial report. There are technical issues and process issues. But, most critically from the perspective of a CFO, who carries personal liability for the accuracy of the data, there are accountability challenges. For example, how do you know you've made the right mapping of figures to XBRL concepts? How do you know that you have complied with UGT architecture guidelines? How do you know that cross-reporting-period relationships are correctly reported? At this time, nearly every company filing XBRL reports is using eyeballs to ensure accuracy. As the number of reported facts climbs into the hundreds, and the complexity of the data model grows, human verification of data accuracy will clearly be an unacceptable method.	Plan for perfection by understanding your requirements. Then push on the XBRL services industry, tools vendors, and UGT steering committee to create the tools and XBRL formulas that you need to support the universal requirement for demonstrably correct filings.

Table 2. Taxonomy lessons learned and best practices

Lesson	Best practice
The first attempts to create our CFT involved modifying the UGT. XBRL provides a clean mechanism for extending other taxonomies, so the technology is well-suited for SEC	Create company-specific extended links for each statement unless the UGT provides one that's a near-perfect match. Think of the company-specific extended link as a definitive

filings. Every filing company begins with the UGT, then customizes it (customization is nondestructive to the UGT). Yet some customization was so extreme, it was more efficient to create a new network of relations rather than customize an existing taxonomy.	statement. Rather than seeing how the company deviates from the UGT (GAAP is *Generally* Accepted Accounting Principles), simply ensure your company-specific extended links make a clear statement (pun intended). Once you've completed your company-specific extended links, the next best practice is to remove the references to the UGT equivalents. They are no longer needed.
It was a challenge to create presentation relations that represented the correct data model relationships between concepts, and that would also support a desired rendering (sequence and indentation) based on the presentation's parent-to-child relations. This is an overloading of the presentation relations. They cannot serve two masters.	Use the presentation relations to secure the best data model, while sacrificing the data display. In the near term, particularly with the 10Q, making this sacrifice is not likely to present problems.
The UGT is big. Comprising than 150 taxonomies, more than 550 files, and over 12,000 concepts, it is over 55MB in size. In the early work, a UGT entry point was used that pulled in nearly everything. The team thought that was smart, but turned out to be wrong. Performance became a problem (especially as the files are loaded over the Internet), and they had to sift through networks of relations that they did not even need.	Identify the exact statements and disclosures you need for reporting, *then* reference those taxonomies (a subset of the 100) in your CFT. 1. Start with your actual financial reports (perhaps with old numbers, but the line items likely will stay the same). Organize them into statements, disclosures, and non-GAAP reports. 2. Load into a taxonomy authoring tool (Taxonomy Designer, XWand) or the SEC Viewer. Wade through the extended links that are likely a good match for your statements and disclosures. It can be helpful to search by a label or part of a label to find the relevant network. Consider looking for other concepts from your same financial statement, as these should be found in the same network. 3. Once you've found the network, note its URL. Unfortunately, the URL is not to be found with the network. You must do a little digging using the Web browser; point to http://xbrl.us/us-gaap/1.0. Statements are in the stm directory, disclosures in the dis directory, and industry-specific entry points are in the ind directory.

4. The specific files are named by a pseudo-acronym. Thus, if you are seeking the "statement of financial position" for the commercial and industrial industry, the appropriate taxonomy is named us-gaap-stm-ci-sfp-cls-2008-03-31.xsd (cls stands for classified). The valid URL is what you need to reference: http://www.xbrl.us/us-gaap/1.0/stm/us-gaap-stm-ci-sfp

Tools will get better. But for now, this is our recommendation for how to identify the exact statements and disclosures you will need for reporting.

Table 3. Instance document lessons learned and best practices

Lesson	Best practice
Financial reports often have figures in millions (for example, from the cash flow statement), others as whole numbers (such as shares outstanding), and some with two decimals places (earnings per share).	For handling the millions, report these values as whole numbers and indicate that they've been rounded to the nearest million. If your financial report provides, for example, a net revenue of $26,311 in millions for Q4 of 2007 (IBM is a big company), the value in the instance document would be reported as 26311000000 with a decimal attribute of -6 (which is substantially different from reporting it with decimal attribute of 0).
Instance documents also contain other details, including definitions for the numeric unit, context, and footnotes. Complex filings may have a hundred or more of these details.	Establish a naming convention so you'll know how to quickly distinguish one context from another. For units, we recommend U-[unit]_[decimal]. For negative decimal values, a minus sign (-) or dash (--) replaces the underscore (_).

Unit attributes	Unit name
Monetary in US dollars, rounded to the nearest million	U_USD-6
Monetary in US dollars, rounded to the nearest penny	U_USD_2
Shares of stock, in whole number increments	U_Shares_0

For nondimensional contexts, [period type]-[relative fiscal year]-[quarter+number, or annual] is recommended. Period type is either I for instant or D for duration. Relative fiscal year is C for current year, P for prior year, PP for prior-prior year, N is for next fiscal year (needed for some disclosures), and year-to-date reporting period is YTD.

Assuming the reporting period is a third-quarter report as of 9-30-2008:

Reporting period attributes	Context name
Instant period type, current reporting year	I-CY-Q3
Duration period type, current reporting period, Q2	D-CY-Q2
Instant period type, prior year, Q3	I-PY-Q3
Duration period type, current year-to-date, Q1 through Q3. Note: Because it is year-to-date, it is assumed to be Q1 through the current quarter.	D-CYTD

Assuming the reporting period is for a fiscal year ending 12-31-2008:

Reporting period attributes	Context name
Instant period type, current reporting year	I-CY-Y
Duration period type, current reporting period, Q4	D-CY-Q4
Instant period type, prior year, year	I-PY-Y

Dimensional contexts are more complex to name because of their length. Include the domain name in full, then an acronym representing the domain member. For example, an annual report disclosure about business combinations has the name "Business Segment [Domain]" and a member "Global Systems Technology [Member]." If the duration reporting period is current fiscal year, year, combined with the domain and member information noted above, the dimensional context would appear as: D-CY-Y_BusinessSegment-GST.

Acknowledgments

Thanks to Bill Hoffay at IBM, who was and remains an integral part of producing these filings. And, thanks to Charles Hoffman and Timothy Randle for sharing their design experience and insights on preparing SEC quarterly filings.

Summary

Use of XBRL is growing worldwide, largely because of its powerful capabilities. An equally compelling reason for XBRL's growth is its adoption by regulatory agencies

and the SEC mandate that all filings be in XBRL format within the next two years. XBRL will certainly disrupt many current methods of financial reporting. There is already significant XBRL activity in the service companies and organizations that support filings to the SEC, including the accounting industry, financial printers, and financial reporting departments of the largest companies. In this article, you learned from the authors' experiences about the opportunities in this challenging area.

Resources

Learn

- XBRL standards are maintained by XBRL International, a not-for-profit consortium of approximately 550 companies and agencies worldwide working together to build the XBRL language and promote and support its adoption.
- Nonprofit Microfinance Exchange, the world's leading business information provider for the microfinance industry, uses XBRL validation and formulae to analyze financial reports.
- Charles Hoffman, the widely acknowledged father of XBRL, provides information on the neutral format table and other XBRL information.
- The SEC publishes XBRL Public Validation Criteria that reflect SEC staff's current views on appropriate validation criteria for XBRL tagging to improve the consistency and quality of XBRL documents submitted to the SEC. The list will help filers and data-tagging software companies improve existing products or create new products to automate testing for these criteria.
- The *XBRL US GAAP Taxonomy Preparers Guide* is a good starting point for preparing XBRL filings. It explains XBRL terminology and concepts so preparers understand how to format financial statements in XBRL.
- Learn more about IBM DB2, which offers industry-leading performance, scalability, and reliability on your choice of platform from Linux® to z/OS®.
- Read about Fujitsu XWand, an application development and runtime environment for building and deploying full-feature, industrial-strength XBRL applications.
- Rivet Software develops financial communications and analysis software, including XBRL creation and viewing applications, and Crossfire Analyst, a Web-based financial aggregation, analysis, and publishing solution designed for accounting, finance, and other business users.
- UBMatrix, Inc XBRL solutions increase operational efficiency and financial transparency, and ensure reporting accuracy and regulatory compliance.
- The United States Securities and Exchange Commission XBRL section provides information about interactive data, news items, and more.
- The SEC Viewer lets you view different taxonomies.
- Browse the technology bookstore for books on these and other technical topics.
- Browse the technology bookstore for books on these and other technical topics.
- In the Architecture area on developerWorks, get the resources you need to advance your skills in the architecture arena.

- Find resources to help you architect enterprise and software systems in free IT architecture kits from IBM.
- Stay current with developerWorks technical events and webcasts.

Get products and technologies

- Download IBM product evaluation versions and get your hands on application development tools and middleware products from IBM, DB2, Lotus®, Rational®, Tivoli®, and WebSphere.

Discuss

- Participate in the IT architecture forum to exchange tips and techniques and to share other related information about the broad topic of IT architecture.

About the authors

David Newman
David Newman is an advisory applications developer. He was the technical team lead for the IBM XBRL filings in the SEC's Voluntary Filing Program in 2007 and 2008. He successfully prepared the 2007 10K XBRL submission per the more detailed filing requirements from the SEC's XBRL Mandate.

Dean Ritz
Dean Ritz is an independent XBRL consultant. He is currently helping IBM and other companies in their preparation of XBRL format filings to meet the mandatory filing requirements coming into effect January 2009. He formerly was a senior taxonomy developer in the professional services group at the XBRL technology vendor UBmatrix, Inc.

Murali Vridhachalam
Murali Vridhachalam, an Open Group certified IT architect, has been involved with XBRL since early 2005. He was the lead architect for IBM's first ever submission of financial reports using XBRL, as part of the SEC's XBRL voluntary filing program. His current interests include SOA and Software as a Service offerings built using the IBM enterprise software portfolio.

Trademarks

IBM, the IBM logo, ibm.com, DB2, developerWorks, Lotus, Rational, Tivoli, WebSphere, and z/OS are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at http://www.ibm.com/legal/copytrade.shtml.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft is a trademark of Microsoft Corporation in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.