

4. Overview, History, State, Trends of XBRL

XBRL is quickly becoming a global standard for the exchange of business data. Why? The primary reason is that something like XBRL needs to exist in the age of the computer and the Internet, coupled with a continuing shift away from paper-based to e-based business models.

XBRL is an XML language. It is important to understand why XML exists in order to understand why XBRL exists and why XBRL is needed. XML will be briefly explained.

In a nutshell, XML is all about two things:

1. Structured information versus unstructured information.
2. One format or syntax for structuring information as opposed to hundreds of different formats for structuring information.

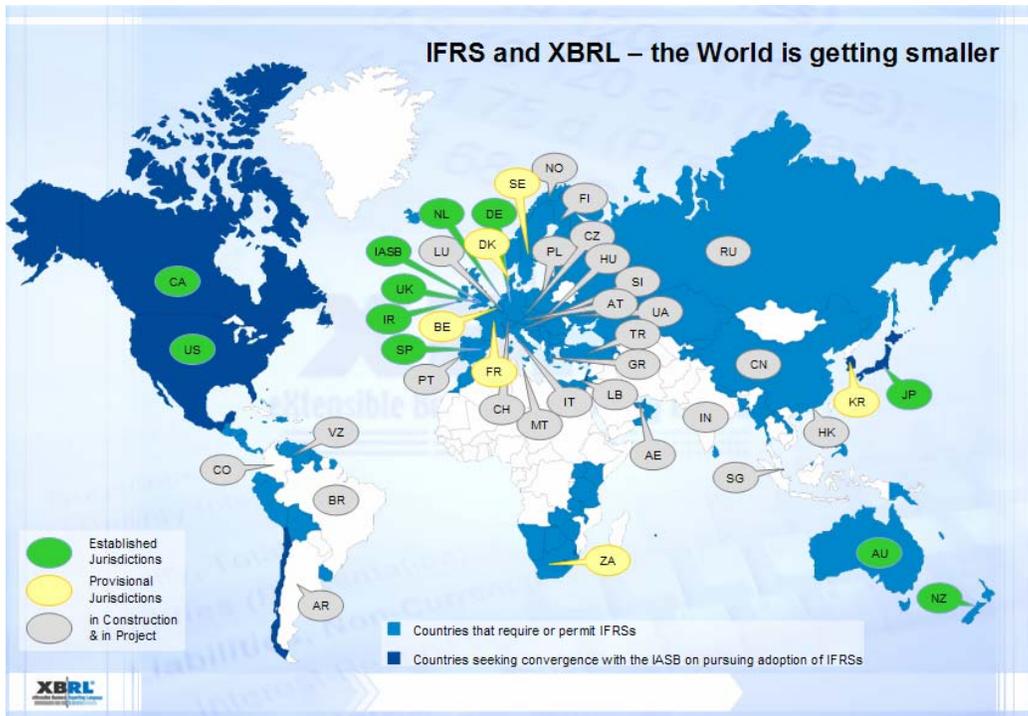
Many people who understand XML don't understand why XBRL is not more like other native XML languages. In order to show why XBRL is what it is and what XBRL does above and beyond a native XML language, it will be necessary to explore the limitations of XML, a topic which is covered later.

In order to make data exchange work within a domain it is important to agree on semantics, or terms to use within a domain. XBRL International and its members have created these terms. One of the first manifestations of the use of these terms can be found in the XBRL taxonomies for US GAAP and IFRS based financial reporting.

It is the author's view that the creation of the taxonomies for US GAAP and IFRS are far more valuable than the XBRL specification itself. However, it would be hard to argue that point, as without XBRL, creation of these taxonomies in a standard way would have been impossible. But making the point is important: it is difficult to agree on these semantics, yet that has happened, the taxonomies for US GAAP and IFRS exist. In fact, the effort to create the taxonomies is likely 10 times as great as the effort required to create the XBRL specification itself.

The need for XBRL is pretty much a no-brainer. That is why there are already so many countries around the world expressing interest in XBRL. When a country is interested in XBRL, they typically establish an XBRL jurisdiction within their country and then become a member of XBRL International. The following is a summary of existing jurisdictions. Additionally, the graphic below shows the existing and provisional jurisdictions as well as countries expressing interest in becoming members of XBRL. The graphic is from XBRL International.

- **Existing Jurisdictions:** USA, Canada, UK, Australia, New Zealand, Japan, Korea, Netherlands, Germany, Spain, Sweden, Ireland, IASCF
- **XBRL Jurisdictions in development:** Argentina, Austria, Belgium, Brazil, China, Columbia, Denmark, Hong Kong, India, Italy, Luxembourg, Mexico, Singapore, South Africa, Taiwan, Venezuela



Each of these organizations is a member of XBRL International, the umbrella organization which maintains the XBRL Specification.

This book covers XBRL version 2.1 only, not XBRL 1.0 or 2.0, which are basically made obsolete by XBRL 2.1. We assume a desire for FRTA 1.0 and FRIS 1.0 compliance. We assume users of this book have the availability of the appropriate software, which will be described in a later section, which provides the appropriate level of XBRL functionality; this is not about using "notepad" to create XBRL taxonomies and instance documents to be used in financial reporting.

4.1. What Does XBRL Provide?

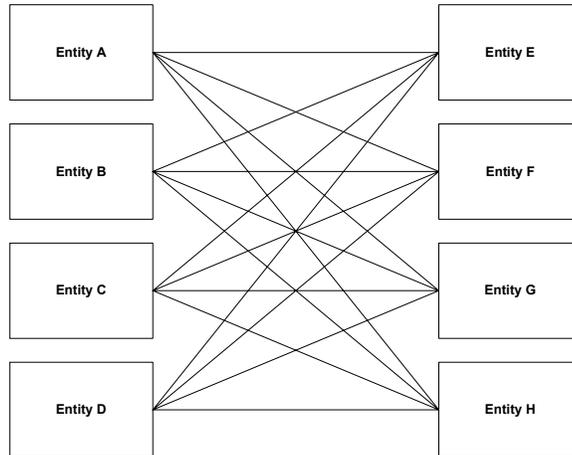
Again, XBRL provides the following fundamental things:

1. XBRL is a **global standard** method for the electronic exchange of business information (replacing 100s of proprietary methods). XBRL is also a method of expressing meta-data and semantics, that is how the business information can be exchanged. Basically, this is what the XBRL Specification provides.
2. XBRL represents a **global agreement of the semantics** of financial reporting concepts and business rules. These concepts and rules, the semantics, have already been created for IFRS and US GAAP. These two taxonomies provide agreed upon semantics against the respective set of accounting standards. So, rather than each company defining its own financial reporting terms and business rules, standard taxonomies of concepts and rules have been created which enhances comparability across companies.
3. XBRL is also an **organization**, comprised of 400+ members from around the world which stands behind and maintains XBRL. The non profit organization XBRL International provides this.

XBRL will cause a fundamental shift in infrastructure relating to the creation, storage, transmission and consumption of business information as the cost/benefit model for creating and using such information has substantially changed.

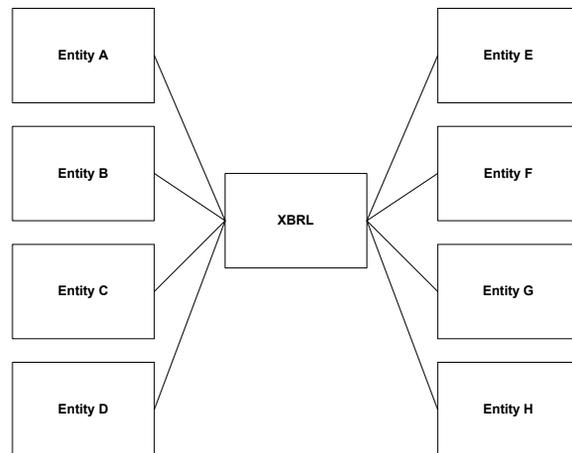
4.2. What Problem XBRL is Trying to Solve

The following graphic explains why XBRL is needed in terms of point 1, the "replacing 100s of proprietary methods":



Above, every entity which wishes to exchange business and financial information with another entity needs to create a "one to one" solution. This graphic does not even paint the entire mess: what if "Entity A" desires to exchange information with "Entity B"? That line does not exist. If you drew every possible line, it would be even messier. Consider only one aspect of exchanging information, validating the data. A "one-to-one" validation needs to be created, probably programmatically, and it is probably limited because it is so expensive.

Now consider the graphic below:



With XBRL as the "standard" that everyone uses, when you desire to exchange information with any other entity, you never have to write a specification, you only communicate, "Lets exchange our information using XBRL". Looking at the validation again, because XBRL has validation built in, you are basically getting

"many-to-many" validation, which is cheaper because many entities by the software, so the software is cheaper for everyone. And because the software is cheaper to build, it has more features.

Keep in mind that the graphic communicates only one of the major advantages of XBRL. And as a by-product of fundamentally being able to exchange information, the real advantage of XBRL will eventually be realized – software which can make use of the meta-data, others developing solutions because of this. Financial reporting will look substantially different in 25 years!

4.3. Brief History of XBRL

The history of XBRL starts with the beginning of XML. XML will be discussed briefly later, but for now, it is only important to know that XML 1.0 was released as a specification in February 1998.

XBRL had its beginnings in April 1998 when the author of this book conceived of the idea of using XML to express financial statements and audit schedules after reading a book on XML in April 1998.

From that point, the AICPA (American Institute of Certified Public Accountants) was approached to spearhead the introduction of what was to become XBRL to the world. The AICPA and Knight, Vale & Gregory, a CPA firm (now part of RSM McGladrey), funded an initial prototype of XBRL. The AICPA funded a business plan, several subsequent prototypes called XFRML (Extensible Financial Reporting Markup Language) and starting the XBRL International consortium. Even before the first XBRL consortium meeting, the AICPA leadership decided to pursue business reporting as a broader scope which was to include financial reporting, rather than limiting the scope to only financial reporting.

In September 1999, the first XFRML consortium meeting took place. The consortium had 13 members at this point.

Soon thereafter, the initial name XFRML was changed to XBRL once the name was determined by the consortium members.

XBRL 1.0 was released in July 2000. The first taxonomy was also released at that time; Financial Reporting under US GAAP for Commercial and Industrial companies.

Feedback was received which started the consortium down the path of creating something more extensible than was possible with XBRL 1.0. XML technologies were maturing and XML Schema was becoming available, as was XLink 1.0.

XBRL 2.0 was released in December 2001.

Extensive testing, including the creation of additional robust financial reporting taxonomies using XBRL 2.0 exposed some weaknesses in that specification. These weaknesses were addressed with the creation of XBRL 2.1 specification which was released in December 2003.

XBRL 2.1 was far superior to XBRL 2.0 for the following reasons:

- XBRL 2.1 had a conformance suite (which XBRL 2.0 did not) which helped software be truly interoperable; there were significant interoperability issues with XBRL 2.0.
- Unnecessary flexibility was removed from XBRL 2.0.
- The quality of the specification was significantly improved do to input from the conformance suite, creation of FRTA and the creation of FRIS.

- Fairly significant real world experience, gained through trying to make use of XBRL 1.0 and XBRL 2.0 improved upon those versions.

Today, XBRL 2.1 works, it is stable, and it can be used effectively in a closed, but more importantly also in an open system, effectively particularly when used with FRTA and FRIS are also used. Also, XBRL International desires to keep XBRL 2.1 stable and without change to at least 2007. This stability is important for both software vendors and business users.

4.4. XBRL Modules

The XBRL Specification is only one part of XBRL. XBRL 2.1 needs to be stable, therefore, to achieve the desired stability but to make additional desired features available, XBRL International is releasing additional functionality in the form of XBRL Modules. These XBRL Modules are basically extensions of the functionality of XBRL 2.1, but without changing the core specification.

These modules include:

- XBRL Dimensions (currently a candidate recommendation)
- XBRL Link Role Registry (LRR, currently a candidate recommendation)
- XBRL Formulas (currently a public working draft)
- XBRL Functions (currently a public working draft)
- XBRL Versioning (or Taxonomy life cycle management, currently in the requirements gathering stage)
- XBRL Formatting (or rendering, currently gathering marketing information relating to the need for this functionality)

4.5. Who is Using XBRL

The following is a brief list of some of those using or preparing to use XBRL. This listing provides a sense as to what XBRL can be used for:

- **APRA (Australian Prudential Regulatory Authority)** – 12,000 regulated organizations, went live with XBRL in 2000
- **Inland Revenue** – similar to the IRS and state offices which collect sales tax; went live with XBRL to collect taxes in 2001
- **FDIC (Federal Deposit Insurance Corporation)** – 10,000 national banks, growing to 65,000 financial institutions regulated by the FFIEC; will go live in 2005
- **Dutch Government** – financial reporting by the Dutch Water Board; will expand to 12 Dutch providences for reporting by municipalities. Currently the Dutch Government is building a taxonomy for all information collected by the government. This XBRL taxonomy will be used to collect information by all agencies of the Dutch Government.
- **FSA (Financial Services Authority)** – regulates 70,000 financial institutions in the UK, will go live with XBRL in 2005
- **ATO (Australian Tax Office)** – similar to Inland Revenue and the IRS; decided to use XBRL in 2003; will go live by 2005
- **Korean Stock Exchange** – regulates approximately 850 listed entities; prototyped XBRL in 2003; will go live in 2004/2005 (53 stock exchanges around the world)

- **EU** - \$1.2 million grant to jumpstart XBRL-based financial reporting in the European Union.
- **CRAS** – Credit Risk Assessment Services currently uses EDI to exchange credit risk information; they are building a solution to replace the EDI exchange mechanisms with XBRL.
- **NBB** – The National Bank of Belgium and Finance and Insurance Commission have developed an extension to the IFRS-GP taxonomy for collecting financial information from banks in Belgium. The 30 largest banks will begin submitting their regulator filings in XBRL beginning January 2006.
- **CEBS** – The Committee of European Banking Supervisors (CEBS) is building Common Reporting (COREP) taxonomy for solvency ratio reporting by 9000 credit institutions and investment firms to 25 supervisors/regulators in Europe under future EU capital requirements regime. This is part of Basel II. They are also creating FINREP, a financial reporting taxonomy for these same financial institutions, this will be modelled using the Belgium Banking Taxonomy as a starting point.
- **US SEC** – The United States Security and Exchange Commission (SEC) announced in February 2005 that public companies could begin supplying XBRL data for specific EDGAR filings. This voluntary filing program is to help the SEC evaluate XBRL.

A pattern exists in these users. The first wave of use of XBRL is being made by regulators who already collect data from those they regulate. Each of these regulators already had some proprietary format for gathering information or used paper of unstructured formats such as PDF.

While the list of regulators may not seem long, the list of those they regulate, who clearly also must make use of XBRL, is significantly longer.

4.6. Limitations of XBRL

XBRL is not suitable for everything. Understanding where XBRL is not useful helps understand what it is useful for.

XBRL is currently not ideally suited for:

- documents which require a content model,
- low-level transactions such as sales invoices and purchase orders; however, XBRL-GL provides some transaction level exchange features,
- presentation of information,
- It is a global solution, better proprietary solutions can be created which have better characteristics, but no better global solution can be agreed upon; XBRL is WHAT was agreed upon.

4.7. Short-Term Future Changes to XBRL

As mentioned, XBRL 2.1 is stable, meaning there are not new versions of XBRL appearing every three or six months requiring modifications to software applications. The market, or users of XBRL are demanding this stability.

At this time, there are no projects in place at XBRL International to create XBRL 2.2 or XBRL 3.0. The focus of the XBRL Specifications Working Group is addressing erratum for XBRL 2.1, making the conformance suite increasingly comprehensive meaning even better cross-vendor XBRL software support. Also,

the XBRL Specification Working Group is working to complete the XBRL Formulas specification.

Even if the XBRL consortium were to start a project to create XBRL 2.2 or 3.0, that project would not start prior to the completion of the formulas specification, and if started would take a year to 18 months to complete at a minimum, based on past history. Therefore, it is likely that there will be no significant changes to XBRL prior to December 2007 or even later.

There are a number of issues relating to calculations which will likely be addressed within the current version of XBRL to the degree that they need to be achieved, then later versions of XBRL will likely build on these interim adjustments which will likely not impact the XBRL core schema files.

4.8. Potential Changes to XBRL

The following is a summary of the known potential changes to XBRL. This information is based on discussions within the XBRL Specifications Working Group and other such discussions.

- As discussed, specifications for formulas (including functions), dimensions, versioning, and formatting are likely.
- It is likely that the CWA (closed world assumption) attribute, which existed in XBRL 1.0, would be added back to XBRL. The reason for this is that there is no way to perform value creation within XBRL, which is a desired feature relating to calculations.
- It is likely that the three components of contexts; entity information, period information, and scenario information; will be separated. For example, units used to be contained in contexts also. That was segregated as it was deemed not to be a true dimension. XBRL 1.0 had entity, period and scenario separate.
- Features required to make XBRL GL work better will likely be added.
- Backwards compatibility to XBRL 2.1 from any subsequent version of XBRL version is pretty much guaranteed and demanded by the market.

4.9. XML

XBRL is an XML-based language. Therefore, it is important to understand a bit about XML in order to understand XBRL.

4.9.1. Brief History of Markup Languages

Fundamentally, what XML is, is a big "time out". There are 100's of ways to achieve what XML is trying to achieve. XML is fundamentally about everyone picking one of those ways, XML, to exchange structured information.

The eXtensible Markup Language, or XML, is a method to add structure and context to unstructured information. XML is a derivative of Standard Generalized Mark-up Language (SGML), a widely supported standard for over 15 years. XML is the direct result of the knowledge gained by experiences with both SGML and HTML. XML is a mark-up language.

Mark-up languages have been around for years. They had their origins in the early 1960s when IBM realized that it had many different types of systems that didn't talk to each other. The solution software engineers, led by Charles Goldfarb, developed was Generalized Mark-up Language, which was a series of codes all systems could share and use to communicate.

XML is the direct result of the knowledge gained by experiences with both SGML and HTML. It really offers nothing new except a standard way of doing hundreds of tasks that are done in different ways today. The result of standardization will be a significant reduction in complexity and costs. Many different things have already been expressed in XML, for example XHTML is an XML version of HTML; XMLEDI is an effort to express EDI transaction sets using XML.

XML is a collection of elements and attributes that describe information and data in a form that both a human and a computer can understand and designed for use over the Internet. But, the internet is not necessary for XML to be useful.

The following is a sample of XML. Note that the sample contains elements, such as <Header>; attributes such as date="1999-10-31"; and values such as "John Doe":

```
<?xml version="1.0"?>
<PurchaseOrder date="1999-10-31" number="PO-10011">
  <Header>
    <Customer id="AB-12345">
      <Name>John Doe</Name>
      <Address>PO Box 99</Address>
      <City>New York</City>
      <State>NY</State>
      <ZipCode>12345</ZipCode>
    </Customer>
  </Header>
  <Line>
    <Quantity>1</Quantity>
    <Item>Blue Sled</Item>
    <Price>22.00</Price>
  </Line>
  <Line>
    <Quantity>1</Quantity>
    <Item>Red Wagon</Item>
    <Price>43.00</Price>
  </Line>
</PurchaseOrder>
```

As we mentioned earlier, XML is not something new, but it offers a standard way of doing work that is now done in hundreds of different ways. This standardization causes reduced costs, reduction of complexity, and fundamentally makes this just that much easier. XML is a global standard.

4.9.2. XML is a Family

XML is really a family of functionality. The basis for the family is XML itself. But, the following are some of the members make contributions to the XML family and to making web services work:

- **XML Schema** (or XML Schema Definition Language) is a W3C standard which allows for the definition of an XML structure, therefore allows validation of XML to ensure it is structured correctly.
- **XSLT** allows for the transformation of an XML document to another form of XML document, or transformation to other formats.
- **XLink** provides a mechanism to link one XML document to another XML document, or other type of document.
- **XPath** provides a mechanism to point to a node or set of nodes within an XML document.
- **XSL-FO** allows for the rendering of XML as PDF documents or other formats.
- **XML Query** provides flexible query facilities to extract data XML documents, somewhat like what SQL is to the database world.

- **SOAP** (Simple Object Access Protocol) is a messaging format for remote procedure call type messages to work over HTTP. SOAP provides a way to communicate between applications running on different operating systems, with different technologies and different programming languages.
- **WSDL** (Web Services Definition Language) is an XML format for describing network services.
- **UDDI** (Universal Description, Discovery, and Integration specification) is somewhat of a "phone book" of for web service offerings.
- **XML Encryption** is a W3C standard that allows for parts of an XML document to be encrypted.
- **XML Signature** is a standard which allows for the expression of digital signatures in an XML format.

There are other members of the XML family, but these give you a sense. See the W3C web site (<http://www.w3.org>) for more information about the XML family. All this contributes to creating what is being hailed as "the semantic web" which will be discussed later.

4.9.3. Other XML Languages, Specifications, Frameworks

XBRL is only one standard for exchanging business reporting information. Also note that XBRL is for exchanging business reporting information, not for reporting transactions such as invoice or purchase order information.

The following are some of the other XML-based standards for reporting, transactions, or frameworks. These XML languages will not exist in isolation:

- **GESMES/CB** is an implementation of the UN/CEFACT's GESMES EDIFACT message.
- **MDDL** (Market Data Definition Language) enables the interchange of information necessary to account, to analyze, and to trade financial instruments of the world's markets.
- **RIXML** (Research Information Exchange Markup Language) is a consortium of buy-side firms, sell-side firms and vendors that have joined together to define an open standard for categorizing, tagging and distributing global investment research.
- **SDMX** (Statistical Data and Metadata Exchange) is an initiative to exchange banking related statistical data.
- **UBL** (Universal Business Language) is an international open standard for the conduct of XML-based electronic business. UBL is more for transactions than business reporting. It defines over 600 different business documents.
- **ebXML** (Electronic Business XML Initiative) allows businesses to conduct business over the internet. UN/CEFACT and OASIS established ebXML to develop a technical framework that will enable XML to be utilized in a consistent manner for the exchange of all electronic business data.

Each of these languages has its purpose. They work together to facilitate doing business in a standard way, rather than hundreds of different ways which only causes inefficiency.

There are two primary points which are being made here. First, pick the right tool for the right job. One can hammer in a nail using a crescent wrench, but a

hammer does work better. Second, XML is definitely the way business will be conducted in the future.

4.9.4. But Why Not Just Use XML; Why XBRL?

Comparing XBRL and XML is not really a fair comparison. XBRL is XML. XBRL is an XML language. XML is a specification for creating languages, by itself XML is not really anything until it is used to create something from it.

XBRL has an XML Schema (or rather a set of XML Schemas), just like other XML languages. But XBRL goes further. XBRL goes above and beyond the rules which XML Schema can enforce. Specifics of this will be discussed later as part of a detailed comparison of native XML and XBRL.

An entire section is devoted to answering the question "Why XBRL and not just XML" later in this material. In that section, we will compare XML and XBRL in detail, but first we want to explain a bit about XBRL so that the comparison will have more meaning.

4.10. XBRL Overview

XBRL International, developers of the XBRL Specifications and other specifications etc. relating to XBRL, is the best source of knowledge relating to XBRL today. XBRL International is actually a consortium of its members. And more accurately, of the 400+ member organizations and perhaps several hundred to a thousand or so employees of those organizations; this material comes from the "handful" of individuals who make most of the contributions to creating of XBRL.

4.10.1. XBRL Specification

This book covers only XBRL 2.1. Basically, earlier versions of XBRL such as XBRL 1.0 and XBRL 2.0 are OK for closed systems, but only XBRL 2.1 is ready for prime time with respect to "open systems".

An open system is defined as some unknown creator exchanging XBRL information with some unknown consumer of that information. If both parties of this exchange transaction are known (a closed system), XBRL 1.0 and 2.0 work fine.

The XBRL Specification is available from the XBRL International web site (<http://www.xbrl.org>). However, as this is a non-technical discussion there is no real need to consult it at this time.

But the XBRL specification tends to be a bit too general, offering too much flexibility which can get into the way of using XBRL for financial reporting. For example, XBRL instance documents are allowed to create duplicate fact values (duplicate items). But what does this mean in terms of financial reporting?

This leads to the FRTA and FRIS which are specifications (stronger than conventions, best practices or guidelines) where agreement could be reached on how XBRL should be used for financial reporting.

4.10.2. FRTA

The Financial Reporting Taxonomy Architecture, or FRTA, is a specification which constrains financial reporting XBRL taxonomies more than the XBRL Specification in order to make these XBRL taxonomies work better for financial reporting. XBRL itself tends to be very flexible as it needs to fill many needs. FRTA removes much of this undesirable flexibility making it easier to exchange financial information within an open system.

This book assumes there is a desire to comply with FRTA.

FRTA has approximately 111 rules. Most of these rules can be automatically tested by software. Other rules must be implemented by humans. This document uses the FRTA recommendation released 2005-04-25 by XBRL International. Software implemented FRTA rules will not be discussed, we will simply assume you have the appropriate FRTA compliant validators.

4.10.3. FRIS

The Financial Reporting Instance Standards, or FRIS, constrains XBRL instance documents more than the XBRL Specification in order to make these XBRL instance documents work better for financial reporting in open systems. Again, XBRL alone tends to be very flexible as it needs to fill many, many needs. FRIS removes undesired flexibility making it easier to exchange financial information within an open system.

This book assumes there is a desire to comply with FRIS.

The FRIS is not a recommendation as of this writing. However, there is a candidate recommendation version and this material assumes that version at this point in time. This version is FRIS candidate recommendation 6 dated 2004-11-05.

4.10.4. Future XBRL Functionality

XBRL is not done. It will likely never be done. The following are major components of XBRL which are not yet complete. While XBRL works without these features, it is important to understand these pieces as they may cause difficulties now for users, difficulties which may be resolved when these features are fully incorporated into XBRL. All of these can be achieved now, but using proprietary solutions. XBRL will likely create or is creating public open standard solutions for these features.

4.10.4.1. XBRL Formulas

One of the highest priorities of XBRL International Specifications Working Group is to complete the "Formulas Specification", or sometimes called business rules for XBRL. This specification will provide robust functionality beyond what the calculation linkbase, which is somewhat impotent, provides for XBRL. In terms of financial reporting probably the most desired feature is cross context calculations which are used extensively in financial reporting.

It is likely that the formula specification will be released as a recommendation perhaps sometime between March 2006 and November 2007. Software will also likely be available in that same timeframe as the specification cannot be released until two vendors support the formulas specification.

Several proprietary formulas solutions exist.

4.10.4.2. XBRL Dimensions

XBRL 2.1 currently has no mechanism for expressing relationships between such things as the entities which the data is about. For example a "Group" can be broken down into "Geographic Segments" or into "Business Segments", both of which are considered dimensions.

XBRL International has released an XBRL Dimensions specification public working draft, and this has moved to candidate release status and will likely become a recommendation before June 2006.

4.10.4.3. XBRL Versioning (or Taxonomy Life Cycle Management)

Of increasing importance to the creators of taxonomies and users of taxonomies is the ability to "version" between two different taxonomies. The same will also be true for instance documents, but the "pain" is harsher for versioning taxonomies currently. There is a project to define versioning requirements, however no project to create a versioning specification currently underway by XBRL International. It is unlikely a versioning specification will appear prior to June 2007, and it actually may be later; perhaps much later.

No proprietary versioning solutions exist for the general public, however, if the versioning specification is not provided, it is likely that some vendor will create a proprietary solution.

Requirements have been gathered for a versioning specification. This includes only requirements for taxonomy versioning, not instance document versioning at this time.

4.10.4.4. XBRL Formatting

Another limitation of XBRL is the ability to provide formatting information which allows a human readable rendering for input and output of XBRL instance document data so that the data can be "consumed" (meaning read) by humans. Like versioning, there is not a current project to creating a formatting solution.

It is highly likely that a vendor or vendors will create a proprietary formatting solution for XBRL instance document data.

Market requirements are currently being collected for creating a public specification for formatting information.

4.10.4.5. XBRL Generic Linkbase

If you look at the linkbases which work with XBRL, you will see a lot of similarities. Currently, each type of linkbase is individually specified. An alternative to that is to create one linkbase specification which would then serve the needs of all the XBRL linkbases. In addition to that, then creating new linkbases would simply be providing additional metadata, not writing an additional specification.

That is what the XBRL Generic Linkbase is all about. A project is underway to create the generic linkbase specification. The specification could possibly be done by the end of 2006. It appears that this linkbase will be used by the XBRL Formulas Specification, that may drive the completing of the generic linkbase specification.

See the "Advanced" section of this document for more information about the generic linkbase.

4.10.4.6. Other

Other functionality discussed by XBRL International includes:

- A standard API (application programming interface) specification for XBRL.
- A composite document format for XBRL instance documents and taxonomies.

4.10.5. XBRL Shortcomings and Unanswered Questions

XBRL is still fairly new. While XBRL 2.1 does work quite well, there are still shortcomings and known features which are required or desirable, but

unsupported by XBRL. For example, the ability to do cross-context calculations is not possible with the XBRL 2.1 specification. However, this feature will be available through the formulas linkbase specification.

This section points out known limitations of XBRL which are good to understand so that your expectations for XBRL are appropriately set. Remember, XBRL is still relatively immature.

Also, realize that although XBRL cannot handle these, users are free to build proprietary software which will handle each of these situations. What is missing is a "global standards" way of addressing these situations.

4.10.5.1. Expressing Dimensions

One missing feature of XBRL is the ability to express cross-context relationships. For example, entities can be related; say the subsidiaries for a group. An XBRL instance document can have, say, information about a group and for all the entities of the group, but there is no way in XBRL to express the relationship between the entities so standard off-the-shelf XBRL processors can do calculations or check calculations expressed in the instance document.

4.10.5.2. Calculations

There are a number of issues relating to calculations in XBRL. While calculations do work quite well, certain edge cases, use cases, taxonomy creation and maintenance issues, etc. exist and still need to be addressed. Many of these are already being addressed with work arounds, additional modules like formulas, and clarifications in the XBRL specification. Other calculation related issues will take longer to resolve. The following is a summary of calculation related issues:

- **Does changing calculations change a concepts definition** – There are two views as to whether changing a calculation link changes the definition of an XBRL concept. Some say yes, others say no. Eventually this will need to be decided so users of instance documents can interpret them the same, if users change calculation links.
- **Duplicate Calculations** – This is a taxonomy maintenance issue. It relates to the fact that it is possible to create "duplicate calculation" links which represent the same calculation in different extended links. Keep in mind that there are no extended links in instance documents. For example, under IAS 7 there are two different methods allowed to create a cash flow statement, namely the direct and indirect method. Under both methods the same three headings are used to segregate the different line items. The three headings are "Cash flows from operating activities" "Cash flows from investing activities" and "Cash flows from financing activities". However while the line items included in the latter two headings under both methods are the same, the line items included in the first heading are different under both methods, but the total figure is the same under both methods. The calculations for "Cash flows from investing activities" can be created in one of the extended links or in both the extended links. These duplicate calculations were going to be made illegal in FRTA, however this rule was dropped as it was complicated to write this rule and more time was needed to think about defining what a "duplicate calculation" actually is. So, there is a possibility of inconsistency in the way taxonomies can be built (either approach could be used) and increased maintenance and chance for error exists, as the two different calculations could accidentally be created differently.

- **Initiating of unwanted calculations** – This is an instance document issue. Currently, there is no way to “turn off” certain extended links. The problem can be seen by loading the IFRS-GP “Sample Company” instance document which has a classified balance sheet. When the instance document is validated, the “Portfolio Basis” (Financial Institutions) show calculation inconsistencies because there are elements common to both the classified balance sheet and the portfolio basis balance sheet. This issue can be resolved by creating calculation links in physically separate files and only referencing the files with the extended links that are required.
- **Reporting Inconsistencies/Errors** – Many people are not comfortable with calculation “inconsistencies”. There is no such thing as a calculation “error” in the XBRL 2.1 specification which only refers to calculations which don’t add up as inconsistencies. The fact some software vendors still report these inconsistencies as errors adds another level of confusion. Most accountants will realize if a calculation is not applicable. More casual users may be confused by these inconsistencies.
- **No Calculated values** – This issue relates to the fact that the CWA (Close world assumption) attribute was dropped primarily because so many people were throwing so much meaning into CWA which never existed and it was causing all sorts of problems. Later, it was realized that one could not make calculations of new information work without it. It was decided to make it explicit in the XBRL 2.1 specification that you CANNOT calculate, or “impute” values. Value creation is another term commonly used for this. The calculation linkbase ONLY documents relationships; it does not create new information. The CWA attribute will likely be added back to XBRL (or something like it) because it is desirable to calculate values. But for now, XBRL cannot compute new fact values.
- **Reword some of calculations in the specification** – The specification group is “rewording” some of the wording in the spec relating to calculations. Mostly this is about “edge cases” which few will encounter, not the main stream areas of calculations most users will encounter. The wording is not as precise as it needs to be to make things crystal clear to the developers of applications. This has been started but deferred as more time is needed to get this wording totally correct. This is a very, very difficult area of the spec. This is under control, but does still need to be done.
- **Concepts with no balance attribute** – This has been an issue for quite some time, and it may be solved, but it is not proven to be solved. If a concept has a balance attribute, there is not a problem in this area. However, if a concept does not, it can be difficult to use the correct “polarity” of the value in the instance document. For example, an instance document creator could enter the value of “1” for a fact, or the value of “-1” for a fact. It can be difficult to know which to enter, and if the wrong polarity is entered, it could cause problems for those using the instance data.

In order for instance documents to be exchanged effectively and automatically, these issues will need to be resolved.

4.10.5.3. Prohibiting Concepts/Elements

Currently, there is no way of prohibiting the use of a concept which exists in an XBRL taxonomy.

4.10.5.4. Formatting

As discussed, presenting XBRL instance document information in a human readable input and/or output format is important for the creation and consumption and use of XBRL instance document data.

4.10.5.5. Versioning

As discussed, a method of versioning XBRL instance documents and taxonomies is needed.

4.10.5.6. Taxonomy for Audit Reports

Currently, there is no taxonomy for expressing information contained in an audit report issued by an independent accountant.

4.10.5.7. Class Properties

Everyone agrees that the current approach for expressing certain types of taxonomy components, referred to as "class properties" currently, is inappropriate. However, there are currently four different options on the table for addressing this issue. See the section on "Modelling Financial Reporting Concepts in Taxonomies" later in this book for a detailed discussion.

4.10.5.8. Extensible Enumerations

A use case exists for expressing certain types of "extensible enumerations" using XBRL. XML schema type enumerations will not work as they are not extensible. A solution exists and has been used by the CRAS working group, however, this is not an officially sanctioned approach at the moment therefore it is risky to use.

4.10.5.9. Hosting of Company Instance Documents and Taxonomies

It is not clear as to the best approach of hosting a companies instance documents and taxonomies. XBRL documents will be very long-lived. There are many issues surrounding this. Actual use of XBRL will likely provide the best information.

4.11. Comparing XBRL and Native XML⁵

Why do we need XBRL? Why can't we just use XML?

Many people ask the question, "What is the difference between XBRL and XML?" First, XBRL is XML; but XBRL adds to XML and uses XML in a somewhat unique way to meet the business requirements it must fulfil. In this next section, we endeavour to make it clear where XML falls short of business needs in some areas, and where XBRL fills that need.

4.11.1. Requirements of XBRL

The best way to start to understand the differences of XBRL and native XML is do take a look at the requirements of XBRL. The following is a summary of these requirements:

- Automated, efficient and reliable extraction of information from an XBRL document
- Automated comparison of information expressed in XBRL documents such as financial and other business information such as, accounting policies, notes to the financial statements between companies and supply chain information

- Drill down from information to more detailed information such as, authoritative literature, audit working papers, etc.
- Support for multiple languages
- Extensibility in terms of adding concepts and modifying relationships
- Semantic and syntactic validation of information within XBRL instances, particularly numeric information, and textual-type information
- Presentation, such as the use of bold, italics, and other stylistic techniques are NOT A REQUIREMENT

The highest priority on the list is extensibility. Business reports can go well beyond forms. Companies which report business information may not find what they need in existing XBRL taxonomies in order to properly report their business information. As such, XBRL must be flexible to meet the needs of its user community. In addition, the data in XBRL documents is commonly numeric and has relationships to other numbers. These relationships need to be accurate.

4.11.2. Summary of Differences

The following is a summary of the features of XBRL. Each of these features is further discussed and elaborated on in the details section:

- **XML:** XBRL is XML.
- **Prescriptive Extensibility:** XBRL's powerful flexibility is achieved through prescriptive extensibility, rather than using the both overly constraining and overly flexible extensibility methods of native XML. It still allows XML's flexibility which can be leveraged in your applications; it is just that an XBRL processor will not use those pieces as it won't understand what to do with them.
- **Express Semantic Meaning:** XBRL provides a robust method of expressing semantic meaning (for example, agreed upon business relationships), such as "Assets = Liabilities + Equity". This feature does not exist in native XML and probably never will.
- **Content Validation of Semantic Meaning:** XBRL provides instance document content validation of semantic meaning, or business rules, using the expressed semantic meaning (again, something not achievable with native XML because there is no method of expressing semantic meaning; you would have to build it yourself).
- **Normalized:** XBRL is a normalized version of XML (that is, the lack of normalization in native XML makes it difficult to extend in a consistent way).
- **Express Multiple Relations:** Because XBRL is normalized, XBRL has the ability to express multiple explicit relations for use with semantic validation, formatting, etc. as opposed to being limited by the one implicit relation provided by native XML's content model. Try and express semantic meaning within the one content model XML provides, you quickly realize the limitation of native XML.
- **Fits into Relational Databases:** XBRL was built to provide an ability to work well with a relational database, because it is normalized and because it is rather "flat". Lots of people use relational databases.

Several other points are worth mentioning about XBRL. First of all, XBRL is not really intended to be used without an XBRL processor. XBRL processors understand all these additional features provided by XBRL. XML parsers don't.

While you could do it, it is not that practical. One would have to dumb-down what you are trying to achieve so much that you would miss many of the useful features of XBRL. Or, you would basically have to build the functionality which is already in an XBRL processor to use the features; therefore you literally end up creating your own XBRL processor.

Open global standards are rather expensive and time consuming to create, especially highly functional languages, such as XBRL. Native XML cannot solve the needs of business reporting, to do so one would have to create an XBRL like XML language to do that. To achieve what XBRL has achieved, it would literally have to be reinvented which took years to agree on, create, let alone implement. But what would you have when you are done? It probably would be something like XBRL, perhaps with somewhat different syntax.

Lastly, one of the values of XBRL is standardized taxonomies; particularly taxonomies like the IFRS-GP taxonomy and US-GAAP taxonomy framework, both of which express semantics for financial reporting. It took thousands of hours to create these two taxonomies, probably more man hours than creating the XBRL specification itself. The point is that this value is above and beyond having XML to create the semantics in.

XBRL brings the software closer to the end user. Software which works with all XML software is typically developer tools, such as XML Spy. Off the shelf software is beginning to be created to use XBRL. Lots of this software will be end user software which will work with any XBRL taxonomy or instance document.

4.11.3. Details of Differences

Now we elaborate on the statements made above, explaining the statements in order to allow the reader to understand why the statement is true. We also explain the business case behind each statement.

4.11.3.1. XML

XBRL is XML.

XBRL uses XML, XML Schema, XLink, XPath, and other W3C XML standards heavily. XBRL is very committed to XML. But, native XML (an XML instance validated by an XML Schema) is only a partial solution to what XBRL needs to achieve. Also, some of XML's powerful features – its strengths – in one circumstance are also its greatest weaknesses given other circumstances.

XML is a meta language used to build other languages. Typically the syntax of these XML languages are expressed using a DTD or more typically today, an XML Schema in order to validate XBRL documents prepared to comply with a specific XML Schema or set of schemas. For example, XHTML is an XML language. As is HR-XML, ebXML, and others. These languages are referred to as "native XML" languages in this document.

A very powerful feature of XML is validation of a content model to ensure it is correctly expressed: the elements and attributes are properly arranged and you are not missing an angle bracket; that the document is syntactically valid. This is achieved using XML and XML Schema. XBRL leverages these powerful features. Data type validation is also achieved via XML Schema, for example a specific element has to be a number, not a string. This is validation of the syntax of the XML. XBRL does give up part of the content model of XML, XBRL is rather "flat". The reason for this will be discussed later.

One thing about XML which is very powerful is its ability to express a rich content model. But at the same time, that content model expresses only implicit

relationships, rather than explicit ones. With XML you can express relationships between concepts in your document, but you cannot express how they are related, or if two different types of relationships exist, the fact that they are different cannot be expressed. Also, you can only express one relationship: the content model.¹

Another powerful feature of XML is its extensibility. Again, this extensibility becomes a problem if the extensibility is used by the producer of information, and the consumer of that information does not expect those extensions.

Since accuracy of the data is so important and XML only provides syntactic validation of the data, XML can only be a partial solution to improve business reporting.

From a business perspective, the point is this: XML can only be a partial solution while XBRL can provide a complete solution to fully automate business reporting. If XML alone is used, each user of XML will have to add additional functionality to any XML solution already provided by XBRL. To meet your proprietary needs, a user of XBRL still may need to add functionality not offered by XBRL, but that list of functionality which would have to be added would be longer had XML alone been used.

4.11.3.2. Prescriptive Extensibility

XBRL's powerful flexibility is achieved through prescriptive extensibility, rather than using the both overly constraining and overly flexible extensibility methods of native XML. It still allows XML's flexibility which can be leveraged in your applications; it is just that an XBRL processor will not use those pieces as it won't understand what to do with them.

One of the most powerful features of XBRL is a robust prescriptive extensibility mechanism, yet flexible in the areas XBRL needs to be flexible. Using this mechanism, users of XBRL can add financial concepts and/or add or remove relations from an XBRL taxonomy. Two things are achieved. First, producers of information can take an existing taxonomy and modify it for their specific reporting needs and second, the changes made by the creator of the information are clearly documented in their extension taxonomy.

An XBRL processor is required to understand extensions, and extensions using XBRL standards will not break the software.

Additionally, the flexible XML extension mechanisms can still be used, but they will likely be used for proprietary purposes, where the producer knows who will be consuming the information and the consumer's software can take advantage of the new elements or attributes added to XBRL.

From a business perspective, the point is this: XBRL provides flexibility where you need it, whereas XML provides too much flexibility where you don't. XBRL allows an end user to take a standard XBRL taxonomy like US GAAP Insurance and make changes to it, extend it, or use a subset of an XBRLized regulatory report like Basel II for bank capital reporting and still use standard software tools. This would not be possible with an in-house developed XML language for business reporting. In addition, you can still add the limitless flexibility offered by XML to XBRL for your internal proprietary needs. XBRL can be both flexible and not break existing applications; it was architected to do just that.

¹ XML Schema provides *identity constraints* which provide limited semantic support that mimics "relational integrity" in a database.

4.11.3.3. Express Semantic Meaning

XBRL provides a robust method of expressing semantic meaning (for example, agreed upon business relationships), such as "Assets = Liabilities + Equity". This feature does not exist in native XML and probably never will.

Another very powerful and robust feature of XBRL is the ability to model semantic meaning, which can be thought of as a way of saying that one can model business rules, for example. This is a very important requirement desired and even demanded by users of XBRL. For example, the Federal Deposit Insurance Corporation (FDIC), which is an early adopter of XBRL has 1800 business rules which needed to be run against data being captured. As they say, "Garbage in, garbage out." XBRL users exchange a great deal of numeric information they require to be accurate and complete.

The semantics which need to be expressed can take different forms. Calculations to handle summations are one form, a formula is another, and the definition linkbase can contain others. The content model of XML allows for the creation of only one content model, and that implies relations rather than making them specific. The important thing is that the content model actually gets in the way of expressing semantic meaning in XML.

From a business perspective, the point is this: The data used in business reporting has a lot of semantic meaning. With XBRL you get a way to express this meaning "out of the box", so each user does not have to create them. Because each user, for example, can reuse the 1800 XBRL business rules to validate bank regulatory reports it is less expensive for everyone. Also, users don't have to reinvent a scheme for expressing the semantic meaning; XBRL provides that scheme which is used by all XBRL software.

4.11.3.4. Content Validation of Semantic Meaning

XBRL provides instance document content validation of semantic meaning, or business rules, using the expressed semantic meaning (again, something not achievable with native XML because there is no method of expressing semantic meaning; you would have to build it yourself).

Because the semantic meaning can be expressed, content validation against this semantic meaning is possible. XBRL provides this also. The example above, "Assets = Liabilities + Equity" is impossible to validate using XML or XML Schema. XML parsers might validate that "Assets" is numeric, or that it must exist, but XML cannot validate the complex semantic meaning which exists in business reporting. XML does not even have a way of expressing the meaning; users have to individually build these solutions above and beyond what you get from simply implementing a solution using just an XML Schema.

From a business perspective, the point is this: Because XBRL has expressed the semantic meaning (see the above section), users can use this semantic meaning to test the data. Quality of the data is improved. What you get with XBRL is a "rules engine" based solution which is a "one-to-many" solution. Many users can reuse the one solution. XML semantic validation can only be built "one-to-one"; basically each implementation needs to build its own. This is part of the reason the FDIC projects it will save \$27 million using XBRL to capture data from US banks.

4.11.3.5. Normalized

XBRL is a normalized version of XML (meaning that XML is not normalized, which causes extensibility issues).

XBRL is normalized, similar to what it means to normalize a relational database, a process makes processing more efficient. What this means is that there is a separation between XBRL concepts (which are defined in Taxonomy—XML Schema—files) and relationships (which are defined in Linkbase—XML—files). Trying to put calculation information, presentation information, definition information, label information, reference information into one (or one set) of XML Schema files would be challenging at best. Trying to then extend this information once it has been expressed would be even more challenging, for example, "Assets = Liabilities + *Commitments* + Equity".

However, because XBRL is normalized, it is actually very straight forward when it comes to trying to rewire a calculation, move or prohibit a presentation, etc.

From a business perspective, the point is this: Because XBRL is normalized, somewhat like the data in a relational database is normalized (not exactly, but you can think of it in these terms), XBRL is more flexible. While XBRL is often considered too complex, business users understand that business reports can be complex and expressing them in a standardized can be technically complex. Ignoring the business requirement of extensibility is not an option for XBRL. However, tools have been created to reduce the complexity to the end user, and easier and more powerful tools will be coming. Normalization leads to other benefits and features that offer advantages compared to XML as noted below.

4.11.3.6. Express Multiple Relations

Because XBRL is normalized, XBRL has the ability to express multiple explicit relations for use with semantic validation, formatting, etc. as opposed to being limited by the one implicit relation provided by native XML's content model. Try and express semantic meaning within the one content model XML provides, you quickly realize the limitation of native XML.

This is the result of normalizing XBRL. Literally an infinite number of different relations or resources can be expressed in XBRL. The key core relations are provided: presentation, calculation, definition. Others can be easily created using the same mechanisms which XBRL uses. This architecture will really make it easy for XBRL to evolve.

With native XML, again, you get one hierarchy, the content model of XML.

While "multiple relations" is not really the totally correct term, it is used because most people understand the term relation or hierarchy or "tree". More precisely, XBRL allows a user to model an "n-dimensional" (infinite number of dimensions) graph, rather than just a tree. Graphs can contain cycles, tree hierarchies cannot. XBRL information should be viewed more like a "web" than a "tree".

From a business perspective, the point is this: Because of XBRL's flexibility it can handle multiple relationships and better express complex business reports. Trying to do this with XML would be problematic. In addition, most users of XBRL will have proprietary needs in addition to the base which XBRL offers. Fulfilling these proprietary needs is easier because literally an infinite number of different types of relations or resources can be expressed.

4.11.3.7. Fits into Relational Databases

XBRL was built to provide an ability to work well with a relational database, because it is normalized and because it is rather "flat". Lots of people use relational databases.

XBRL provides an infrastructure for modeling relational data. Lots and lots of people have relational databases which contain business information which will be

used to generate XBRL instance documents and/or to capture business information and store it after it has been received from someone else.

Because XBRL is normalized, it is quite easy to import information from XBRL taxonomies and instance documents into relational database tables.

From a business perspective, the point is this: It is important for XBRL to work well with relational databases because (a) there are a lot of relational databases in existence, (b) that is where much of the data which is exchanged is stored, (c) that is where data received from others will likely be stored. Exchanging financial data between relational databases and XBRL is efficient, reducing costs of automating the process.

4.11.4. Some Example

What follows are a few examples which expand on what was discussed above, providing more detail in several specific areas. These examples are intended to provide specific examples which help readers see the points which are being made.

4.11.4.1. XML syntax validation

Syntax is the structure of the "stuff" inside an XML document, for example the following is the syntax of XML:

```
<foo:bar john="doe">345</foo:bar>
```

This is a different syntax:

```
[foo:bar john/doe/]345[/foo:bar]
```

XML parsers enforce syntax, not semantics. The structure is validated in the first example; if the beginning tag `<foo:bar john="doe">` and the ending tag, say `</foo:barbar>`, did not match, the XML parser would report that the syntax was not valid.

4.11.4.2. Value Correctness

One big problem which exists with native XML alone is determining the accuracy and therefore the reliability of the information. To have fully automated information exchanges, error detection is required in the information provided, that is an ability to assert as to the information correctness. The regular XML Schema specification falls short on many accounts if the exchange of reported business information is the application domain. XML only provides what is called syntax validation, or asserts that the right words are being used and that they are in the right order.

Consider the following example:

```
<foo:TotalAssets>100000</foo:TotalAssets>
<foo:TotalLiabilitiesAndEquity>200000</foo:TotalLiabilitiesAndEquity>
```

So, what is wrong with the example above? Well, any accountant will tell you that the balance sheet has to "balance", that "Total Assets" must equal "Total Liabilities and Equity". With XML, you can specify the syntax for the XML, and using XML Schema, you can specify a data type for the elements. However, XML has no method of specifying the value of one element relative to the value of another element.

Now, users can overcome this by building these business rules in their applications. But there are two problems with that. First, building this validation

takes time, so it costs money. Second, each user may build different validation leading to different results.

This type of validation is built into XBRL.

4.11.4.3. Extensibility

The Computers breakdown items are not part of the Universe of Discourse and can not be reported about. Consider the following:

	2003	2002
	€ 000	€ 000
Land	5,347	1,147
Buildings	244,508	366,375
Computers and Equipment	4,169	5,313
Furnitures and Fixtures	34,457	34,457
Other	6,702	6,149
Total	295,183	413,441

Say a schema contained "Land" and "Building", but the you needed to also report "Computer Equipment" and "Furniture and Fixtures". The XML only knows about "Land", "Building" and "Other" from the schema. You added other stuff. How does an application know how to deal with the stuff you added or what the stuff is? XBRL specifies a way to extend so applications KNOW how things will be extended. XML's greatest strength, extensibility, is also its greatest weakness, you can extend it in so many ways.

Consider the example below. The format of the reported value is invalid for Furniture. An XML parser would catch that if the data type were "decimal".

Land	5,347	1,147
Buildings	244,508	366,375
Furnitures and Fixtures	34,457	many thousands
Computers	2,069	3,113
Servers	1,850	1,600
Desktops	216	1,513
Equipment	2,100	2,200
Other	6,702	6,149
Total	295,183	#VALUE!

Now you added detailed information to "Computers", breaking out that one line item into two different line items. XML cannot do this. XBRL can.

4.11.4.4. Semantic validation

Consider the example below. An attentive reader will notice that the first column, in the example below does not add up to the reported total.

	2003	2002
	€ 000	€ 000
Land	5,347	1,147
Buildings	244,508	366,375
Computers and Equipment	4,169	5,313
Furnitures and Fixtures	34,457	34,457
Other	6,702	6,149
Total	295,183	413,441

The syntax is correct, the structure is correct, the data types are correct, but the reported information is not valid. The reader expects the rows to add up to the reported total value and it doesn't. What are semantics in the reporting context? The users of spreadsheets are conditioned to expect that these rows would add up to a total. How does a receiving application understand context if all it receives are a list of numbers?

XML does not have the expression power to explain the following to an application:

- this set of numbers should add up to this number.
- this number on the balance sheet equals that number on the income statement.
- this number on the cash flow statement plus that number on the income statement equals this number on the balance sheet.

XBRL can express these types of relations.

4.11.4.5. Business rules

A more complex application of semantics is when they are expressed as business rules. These are situation dependent rules as opposed to domain dependent rules. To automate an information exchange, it would be very beneficial to be able to set thresholds outside which human intervention is needed.

To do this, rules defined in explicit statements that the information must adhere to are required and these rules must be attached as properties of the document structures. The assertion of information correctness is a crucial but missing aspect of the XML specification.

The previous examples have shown that XML:

- is the perfect foundation for reporting business information.
- provides some rudimentary validation.
- does not cover all the needs for reporting of business information.
- does not provide for the fully automated exchange of business information.

4.11.4.6. How does XBRL Solve These Problems?

The fundamental difference between XBRL and XML is that XBRL makes the associations between concepts explicit.

- This reification allows properties to be attached to the association.
- The reification allows meaning (semantics) to be attached to a relation between concepts.
- Its explicit relations over-ride the rigor of the implied structure that XML Schema structure definition forces upon its users.
- The reification allows the defining of computational and business rule associations between elements.

4.11.4.7. XBRL: The 'missing' extension to XML

The previous explanations made clear the need for an Open Standard Markup Language such as XBRL that provides a generic platform for deploying the XML technologies in business information reporting and exchange applications.

True automated interoperability and information exchange will only occur when correctness of information can be verified by applications.

4.11.4.8. COTS Software specific for XBRL

XML Spy is commonly used consumer off the shelf (COTS) tool for building XML schemas or XML languages. It is impossible to make use of all of XBRL's features using XML Spy, or other such XML tools. Likewise, it is impossible to make use of XBRL's rich set of features using just an XML parser.

XBRL processors will, most likely, exist in the future that are of reasonable cost for client-side use. These XBRL processors may range in price from free to \$99 to maybe \$499 and more for server-side functionality. Several open source versions of XBRL processors already exist.

XBRL taxonomy creation software is available today for between \$2,000 and \$3,000 for sophisticated XBRL taxonomy creation tools. Most taxonomies will likely be created by a few and used by many. In addition, more simple extensions will likely need to be created by users and these tools will typically be available with XBRL document creation software.

Basic XBRL document creation software will likely be free or low priced. Microsoft has created an XBRL Toolkit which it is making available for free.

Many XBRL documents will be created by existing applications which add XBRL functionality. For example, several software vendors are using Excel to build instance documents – XBRL processors are called from within Excel.

In addition, robust XBRL document creation tools will be created and will likely be priced at \$99, \$500, \$1,000, and more.

Robust integrated business reporting, content management, etc. solutions will likely be available for \$50,000 to \$500,000 which will save companies millions of dollars, improve their data quality, and be well worth the prices for this software.

4.12. State/Maturity Level of XBRL

XBRL is still in its infancy as is software which supports XBRL. There are an increasing number of vendors of the basic software which is required to make use of XBRL – an XBRL processor – but user oriented tools and tools with sophisticated and useful functionality do not currently exist at present.

Think of Visicalc in 1981. Compare that with Microsoft Excel 2003. XBRL is where spreadsheets and spreadsheet software was in 1981. XBRL will be very usable when it reaches the point in its maturity when it is comparable to the level of maturity of Excel 2003. The rate at which XBRL matures into a fully functional business tool depends on the speed of adoption on a world wide basis. All the indications are that the adoption process is gathering momentum which, in turn, will drive the development of the much-needed software required to meet the diverse needs that XBRL can, and will, address across the business reporting supply chain.

4.12.1. Core XBRL Infrastructure Ready

The vast majority of the infrastructure in terms of XBRL are ready for use. The XBRL 2.1 specification is a solid, tested specification. It has a conformance suite which exercises software and implementation of XBRL by vendors has made significant contributions to the conformance suite.

While XBRL can effectively be used, and is being used by regulators such as the FDIC, there are a few pieces to the XBRL puzzle which still need to be built out.

As mentioned in a previous section on the future of XBRL, several areas such as dimensions, formulas, and versioning still need to be built out. These additional features will be built as modules to XBRL, not changing the XBRL 2.1 specification itself.

In the interim, proprietary workarounds can be created to temporarily fill the gaps of functionality. For example, the FDIC implemented a proprietary business rules specification as it needed to have business rules as part of its implementation. This implementation is very similar to the probable form of an XBRL International formulas implementation. They have commitments from vendors to convert to the XBRL International standard at some point when that specification becomes available.

4.12.2. Taxonomies Ready

Taxonomies are also ready for use, although be careful. Neither the IFRS or US GAAP taxonomies have gone through the stress test of having many, many instance documents created to see how good they work. While the taxonomies are quite good as a result of testing, they may not necessarily be ready for prime time. Only actual use will ferret out problems, but the problems should be minor.

One very good thing about XBRL is that if there are problem areas in the taxonomies, extension taxonomies can be created to work around these problems.

The FRTA has contributed to high quality taxonomies such as the IFRS-GP and US GAAP taxonomy framework. It is highly likely that these two taxonomies will be the benchmark by which other taxonomies will be judged.

Lastly, there is quite a spectrum of XBRL taxonomies which have been built:

- IFRS-GP financial reporting taxonomy,
- Belgium Banking taxonomy, an extension of the IFRS-GP taxonomy for regulatory reporting of financial institutions,
- US GAAP taxonomy framework for financial reporting,
- Edgar Online extension to the US GAAP taxonomy framework which is used by thousands of instance documents,
- CRAS for credit risk,
- COREP for Basel II reporting by financial institutions,
- XBRL GL for lower-level general ledger transactions.

This variety has stressed XBRL around the edges. Creating these taxonomies have contributed to the quality of XBRL in general and quality and interoperability in XBRL software.