Leveraging XBRL Extensibility Effectively

By Charles Hoffman, CPA (Charles.Hoffman@me.com)

Last Revised – October 16, 2018, 2018 (DRAFT)

"He who loves practice without theory is like the sailor who boards a ship without a rudder and compass and never knows where he may cast." Leonardo da Vinci

Executive summary:

- XBRL builds on top of XML.
- Native XML is not extensible; XBRL can be safely extensible.
- If you choose to leverage XBRL's extensibility you have to manage that extensibility appropriately.
- A complete set of clear rules is mandatory for information conveyed within an XBRL instance to be understood as intended.

Copyright (full and complete release of copyright)

One type of practical knowledge is **know-how**; how to accomplish something. This document explains how to accomplish something.

The document *Blueprint for Creating Zero-Defect XBRL-based Digital Financial Reports*[1] explains why an XBRL-based report must be validated an how to do that. The document even shows examples of common mistakes made when creating XBRL instances to convey information.

What that document does not do, however, is explain in detail the issues introduced by XBRL's extensibility, why those issues can be problematic, and how to overcome those issues to effectively convey information between knowledge bearers and knowledge users. Keep in mind that some knowledge users might be machine-based processes.

The purpose of this document is to explain how to control XBRL's extensibility so proper software can be created which helps business professionals use XBRL without needing to delve into the technical details of XBRL.

XBRL Builds on Top of XML

XBRL is an approach to using XML[2]. XBRL is actually more similar to RDF than it is to XML. XBRL is XML; XBRL uses the XML syntax. Therefore, XBRL can leverage the entire family of XML specifications. XML allows you to construct "trees". XBRL and RDF allow you to construct "graphs".

XBRL expresses semantics (meaning, business logic) in a global standard format. XML only articulates syntax. You can create mechanisms using XML to represent meaning. For others to do what XBRL does with XML, you would basically have to reinvent what XBRL has already created. Because this meaning, the business logic can be expressed in a global standard format that meaning and business logic can be effectively exchanged.

XBRL allows validation against the expressed meaning and business logic also. Because the meaning (semantics) are expressed in machine readable form; it is possible to validate XBRL instances against that expressed meaning or business logic. And XBRL has created global standard mechanisms for performing this validation, such as XBRL calculations, XBRL definition relations, and XBRL Formulas.

XBRL separates concept definitions from the content model. Typically with XML languages, the concept definitions and the content model are mixed together within an XML schema. This makes it difficult to reuse either the concepts or the concept definition. Further, XML provides

[1] *Blueprint for Creating Zero-Defect XBRL-based Digital Financial Reports*, http://xbrlsite.azurewebsites.net/2017/Library/BlueprintForZeroDefectDigitalFinancialReports.pdf
[2] XBRL Builds on Top of XML, http://xbrl.squarespace.com/journal/2009/4/25/xbrl-builds-on-top-of-xml.html

you with only one implicit set of relations (because it has only one content model) and the definition of those relations is mixed with the definition of elements and attributes. XBRL, on the other hand, uses an atomic approach (flat XML content model) similar to RDF in defining concepts and moves the expression of relations away from the XML schema. This separation of concept and relation definition leads to the next benefit of XBRL, you can express more than one set of relations and each of those sets of relations can be explicitly identified as being for a specific purpose.

XBRL can express multiple hierarchies of explicit relations. Because XBRL separates concept and relation definitions, you can define more than one hierarchy of such relations. Further, the hierarchies of relations defined are explicit rather than XML's implicit content model. But, you must keep those XBRL hierarchies consistent in terms of meaning. One representation of meaning or business logic should not be inconsistent with another representation.

XBRL provides prescriptive extensibility. XML's greatest strength is also its greatest weakness. XML is extensible everywhere, in every direction. XBRL is extensible in a specific, prescriptive, and therefore predictable manner. As such, the extensibility is usable without modifying software for the extension. You can think of this as XBRL always having the same "shape".

XBRL provides a multidimensional models. The multidimensional provides a consistent model with flexibility in the right areas which provides for flexible presentation of information and the ability to "slice and dice" information. Although XML can be made to fit into a multidimensional model in many cases, doing so can be a struggle.

XBRL enables "intelligent", metadata driven connections to information. With XBRL, connections to information can be created by business users adjusting metadata rather than by requiring technical people writing code. As such, rather than building multiple point solutions, XBRL enables the creation of effective and efficient solutions that allow extendibility and don't require programming modifications to connect to new information or new information models. Again, this is because of the prescriptive manner of XBRL's extensibility, the predictable "shape" of XBRL is always the same. With XML, every new connection pretty much has to be enabled by a programmer writing code because XML only communicates technical syntax and does so at the data level, not the meaning or business logic level, of information and because the shape of different implementations of each XML implementation can be so varied.

JSON is similar to XML in that it is a syntax. Some people prefer the JSON syntax over the XML syntax; but to use all the features offered by XBRL in JSON; just like if you wanted to use XML to do what XBRL does; to use JSON like XBRL can be used, you would basically need to reinvent

XBRL in the JSON syntax. JSON and XML have pros and cons[3]. JSON and XML are two very different tools[4]. XML is a language for building languages and is very powerful and is a family of technologies[5]. JSON is a data format. Use the right tool for the right job.

Flexibility Means Responsibility

XBRL has a predictable shape. What you put into that predictable shape is up to you. However, what you represent must be logical by some definition. But how do you define what is logical? How do you explain to creators of XBRL-based reports what is allowed and what is not allowed via an XBRL taxonomy so that reports based on that taxonomy are logical?

The consequences of not defining what is allowed and what is not allowed can be observed in XBRL-based financial reports of public companies to the SEC[6]. The logical errors of such reports are easy to find because the semantics and business rules are clearly defined. The meaning and business rules of financial reports is US GAAP. US GAAP has been used for almost 100 years and is for the most part well understood. However, US GAAP is not written in machine-readable form. But aspects of US GAAP can be represented in machine-readable form and then information about the consistency or inconsistency of financial reports to US GAAP can be measured using automated machine-based processes.

Reverse engineering the observations in these XBRL-based reports can be used to figure out the patterns of errors that exist in this rich set of XBRL-based information. Empowered by this information, one can then figure out the rules that must be created to eliminate such inconsistencies.

Rules prevent anarchy[7].

Judicious use of Rules to Guarantee Information is Correct

Information can only be guaranteed to be correct to the extent that machine-readable rules are provided to assure that information is correct. For every dimension of flexibility allowed, rules must be provided by the system to control that dimension of flexibility. Stepping through the

[3] W3C, *JSON vs XML*, https://www.w3schools.com/js/js_json_xml.asp

[4] Yegor Bugayenko, *Stop Comparing JSON and XML*, https://www.yegor256.com/2015/11/16/json-vs-xml.html

[5] Airi Salminen. *XML Family of Languages*, http://users.jyu.fi/~airi/xmlfamily.html

[6] Quarterly XBRL-based Public Company Financial Report Quality Measurement (September 2018), http://xbrl.squarespace.com/journal/2018/9/29/quarterly-xbrl-based-public-company-financial-report-quality.html

[7] Comprehensive Introduction to Business Rules, http://xbrlsite.azurewebsites.net/2017/IntelligentDigitalFinancialReporting/Part01_Chapter02.4_ComprehensiveIntroductionToBusinessRules.pdf

system of XBRL-based financial reports will show you how to control flexibility through the judicious use of machine-readable rules:

- **XBRL technical syntax**: While the XBRL-based financial reports of public companies have some latitude as to how to represent information using the XBRL technical syntax; all companies must conform to the XBRL technical syntax. XBRL International provides a conformance suite that indicates what is allowed and what is not allowed. Software vendors are expected to be consistent with that XBRL conformance suite. The conformance suite tests software's consistency with the rules specified in the XBRL technical specification. When XBRL-based reports of public companies are submitted to the SEC EDGAR system, inbound testing prior to accepting the reported information checks to make sure submitted information is consistent with expectation in terms of the XBRL technical syntax. This works well and 99.99% of all XBRL-based reports are consistent with the expected XBRL technical syntax.

- **Model structure**: While the XBRL-based financial reports of public companies have some latitude as to how to represent information related to XBRL presentation relations; there are clear guideline in most cases. Note that I am saying there are clear guidelines in "most cases". Why are guidelines not clear in "all cases"? Well, it can be clear…however, the SEC neglected to provide clear rules as to the relationships between the categories of report elements that make up an XBRL-based financial report: Networks, Tables, Axis, Members, Line Items, Concepts, and Abstracts. To resolve these issues all that needs to happen is the allowed and disallowed relationships must be made clear.

- **Reporting styles**: While all financial reports of public companies are required to be consistent with US GAAP; there are different ways, or styles of reporting, that are consistent with US GAAP. Reporting styles are basically patterns. Approximately 80% of public companies us one of 29 specific reporting styles[8]. The remaining 20% of companies use somewhere between 27 and possibly 172 additional reporting styles. The errors caused by not having clearly defined reporting styles is easy to solve; the solution is simply provide the complete set of allowed reporting styles.

- **Continuity cross-checks**: Information reported within financial reports must be internally consistent and logical. Information reported in on part of a report cannot contradict information reported in another part of a report. Related to the notion of reporting styles is the notion of continuity cross-checks. Continuity cross-checks are simply high-level relationships between financial concepts that are universally

[8] Making the Case for Reporting Styles, http://xbrlsite.azurewebsites.net/2017/library/MakingTheCaseForReportingStyles.pdf

consistent within a reporting style. For example, “Assets = Liabilities and Equity”, the accounting equation, is a continuity cross-check. The way to make sure that the continuity of the information within a report is logically consistent is to provide continuity cross-check rules.

- **Types**: Information reported within financial reports must be internally consistent and logical as we stated above. Another way information can be inconsistent is that detailed items can be used in an incorrect manner. For example, the concept “general and administrative expenses” is always part of “operating expenses”; “operating expenses would never be part of “general and administrative expenses”. To prevent information from being used in an incorrect manner in a report type or class relations are used to define the appropriate use of a concept. If this is done in machine-readable form then automated validation processes can be used to detect misuses of concepts.

- **Reporting checklist**: Financial reports most conform to statutory and regulatory rules. For example, an economic entity creating a financial report is required to explain the “basis of reporting” of the financial report. Certain disclosures are always required including: balance sheet, income statement, cash flow statement, basis of reporting, nature of organization, revenue recognition policy, and significant accounting policies. Other disclosures are required if specific line items are reported. For example, if the line item “Inventories” is reported on the balance sheet; then an inventories policy is expected and the disclosure of the components of inventory is required. The reporting checklist organizes all of these statutory and regulatory rules in machine-readable form to the extent possible.

- **Disclosure mechanics**: Each disclosure is represented in machine-readable form. These representations take the form of structural, mathematical, logical, and other such mechanical relationships between the report elements that make up a disclosure. For example, an inventory components disclosure is a roll up therefore it must follow the rules of a roll up representation the roll up is expected to actually “roll up” (i.e. foot) mathematically and that roll up disaggregates the concept “us-gaap:InventoryNet”.

What I am pointing out is that flexibility can be provided and effectively controlled. There are additional control mechanisms that can be used. It is to the extent that these control mechanisms are provided that (a) flexibility can be provided and (b) quality of information can be still be maintained. Not providing these control mechanisms means that quality issues will inevitably exist. Additional discussions of these mechanisms are beyond the scope of this document[9].

And so, imagine that you do provide all of the necessary mechanisms for controlling quality and you provide the flexibility you desire and you do realize the quality you anticipated as a result of these measures and all information is verified prior to the information being submitted and then made available within some financial information database. Those same rules provide rich functionality which can be leveraged for extracting and making us of this information.

The aspects summarized above have been proven to be issues within XBRL-based financial reports submitted by public companies to the Securities and Exchange Commission. I have created an entire framework for creating rules using the XBRL technical syntax[10]. Within that framework I have created rules for US GAAP, rules for IFRS, and rules for a reporting scheme that I created to test these ideas which I call the XASB reporting scheme. This helps understand the capabilities of rules to assure high-quality information within reports.

Next, I want to explain specific information related to what rules XBRL provides for detecting errors, what XBRL does not provide and therefore what is missing must be supplemented, and examples of specific, common mistakes that are made and how to avoid those mistakes.

Presentation Relations Rules Not Available from XBRL

An XBRL taxonomy is a collection or set of report elements. A **report element** is simply an element or part of a report. Report elements can be categorized into groups. Those groups are:

- Network
- Table (or hypercube)
- Axis (or dimension)
- Member
- Line Items (primary items)
- Abstract
- Concept

These are the only pieces you have to work with when creating a report. When you extend an XBRL taxonomy you can only create one of those types of report elements. You cannot define a

[9] If you desire more information please see, *Blueprint for Creating Zero-Defect XBRL-based Digital Financial Reports*, http://xbrlsite.azurewebsites.net/2017/Library/BlueprintForZeroDefectDigitalFinancialReports.pdf

[10] Here are the rules, http://xbrlsite.azurewebsites.net/2018/Pesseract/DynamicRules_AllFilesLocal.zip

new type of report element. I will show in a moment that 100% of the XBRL-based reports submitted to the SEC can be broken down into those seven categories.

Further, when you create relations within an XBRL taxonomy, those report elements must be related in very specific logical ways. When you add report elements within an XBRL definitions linkbase structure, the XBRL technical syntax of the XBRL Dimensions[11] specification controls those relations very specifically. Those relation rules are enforced by an XBRL processor which is required to give feedback when a relation is structured incorrectly. A discussion of all those rules is beyond the scope of this document, the point is that XBRL definition relations that are related to XBRL Dimensions are enforced by the XBRL Dimensions specification.

Likewise, XBRL Calculation relations rules are enforced by the XBRL specification[12]. For example, you cannot create an XBRL Calculation relation between two string concepts. Another rule is that you cannot create an XBRL Calculation relation between two numeric concepts where one concept has a period type of “instant” and the other has a period type of “duration”. Again, relation rules are enforced by an XBRL processor which is required to give feedback when an XBRL Calculation relation is structured incorrectly. A discussion of all those rules is beyond the scope of this document, the point is that XBRL Calculation relations are enforced by the XBRL technical specification.

On the other hand, XBRL Presentation relations are only partially enforced by the XBRL technical specification. Every XBRL Presentation relation has exactly one type of relation[13]. That relation is defined by the XBRL technical specification: http://www.xbrl.org/2003/arcrole/parent-child.

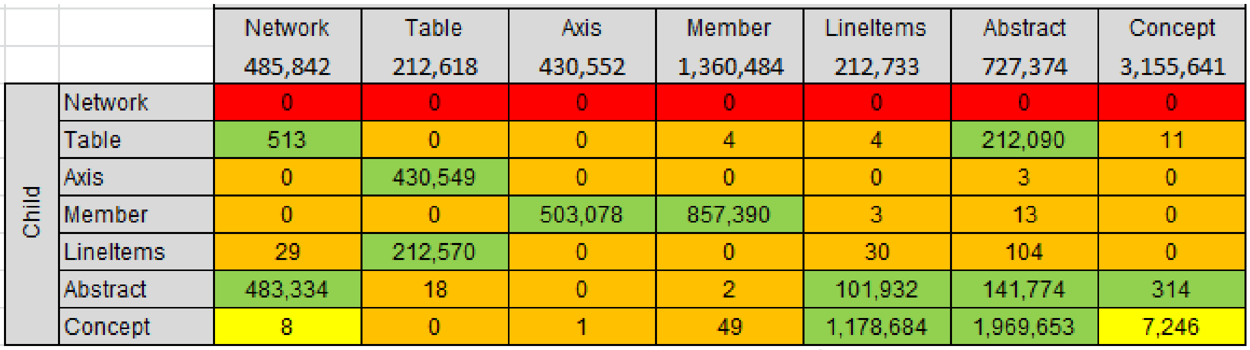
Further, the XBRL technical specification itself has no formally defined rules related to how different report elements can be related to other types of report elements. So, illogical or even pathological relations can be created and are not required to be reported by software that reads these relations per the XBRL technical specification. But illogical and pathological relations are logical or structural mistakes and can make it difficult or even impossible to interpret the meaning of reported information.

The following table summarizes the relations between the report elements that exist in the 10-K financial reports submitted to the U.S. Securities and Exchange Commission by approximately 6,000 public companies for fiscal year 2015:

[11] *XBRL Dimensions 1.0*, http://www.xbrl.org/specification/dimensions/rec-2012-01-25/dimensions-rec-2006-09-18+corrected-errata-2012-01-25-clean.html

[12] *Extensible Business Reporting Language (XBRL) 2.1*, http://www.xbrl.org/Specification/XBRL-2.1/REC-2003-12-31/XBRL-2.1-REC-2003-12-31+corrected-errata-2013-02-20.html

[13] *5.2.4.2 The <presentationArc> element*, http://www.xbrl.org/Specification/XBRL-2.1/REC-2003-12-31/XBRL-2.1-REC-2003-12-31+corrected-errata-2013-02-20.html#_5.2.4.2



		Network	Table	Axis	Member	LineItems	Abstract	Concept
		485,842	212,618	430,552	1,360,484	212,733	727,374	3,155,641
Child	Network	0	0	0	0	0	0	0
	Table	513	0	0	4	4	212,090	11
	Axis	0	430,549	0	0	0	3	0
	Member	0	0	503,078	857,390	3	13	0
	LineItems	29	212,570	0	0	30	104	0
	Abstract	483,334	18	0	2	101,932	141,774	314
	Concept	8	0	1	49	1,178,684	1,969,653	7,246

In the columns are the categories of report elements that are the PARENT in a relation. The rows are report elements that are the CHILD in a relation. RED cells show XBRL errors of which there are none because XBRL-based reports are 99.99% or better consistent with XBRL technical syntax rules. A Network can never be represented as a CHILD of another Network in an XBRL presentation relation. And, sure enough, no reports have that relation represented in the set of 6,000 reports analyzed.

GREEN cells show generally appropriate relationships or relations that are not illogical (i.e. the relations are logical). For example, if you look at the column titled "Network" you see that while most reports have an [Abstract] report element as the root element in a set of XBRL presentation relations indicated by the number 483,334; there were 513 relations where [Table] was used as a root report element in a Network, 29 where [Line Items] was used, and 8 where a Concept was used.

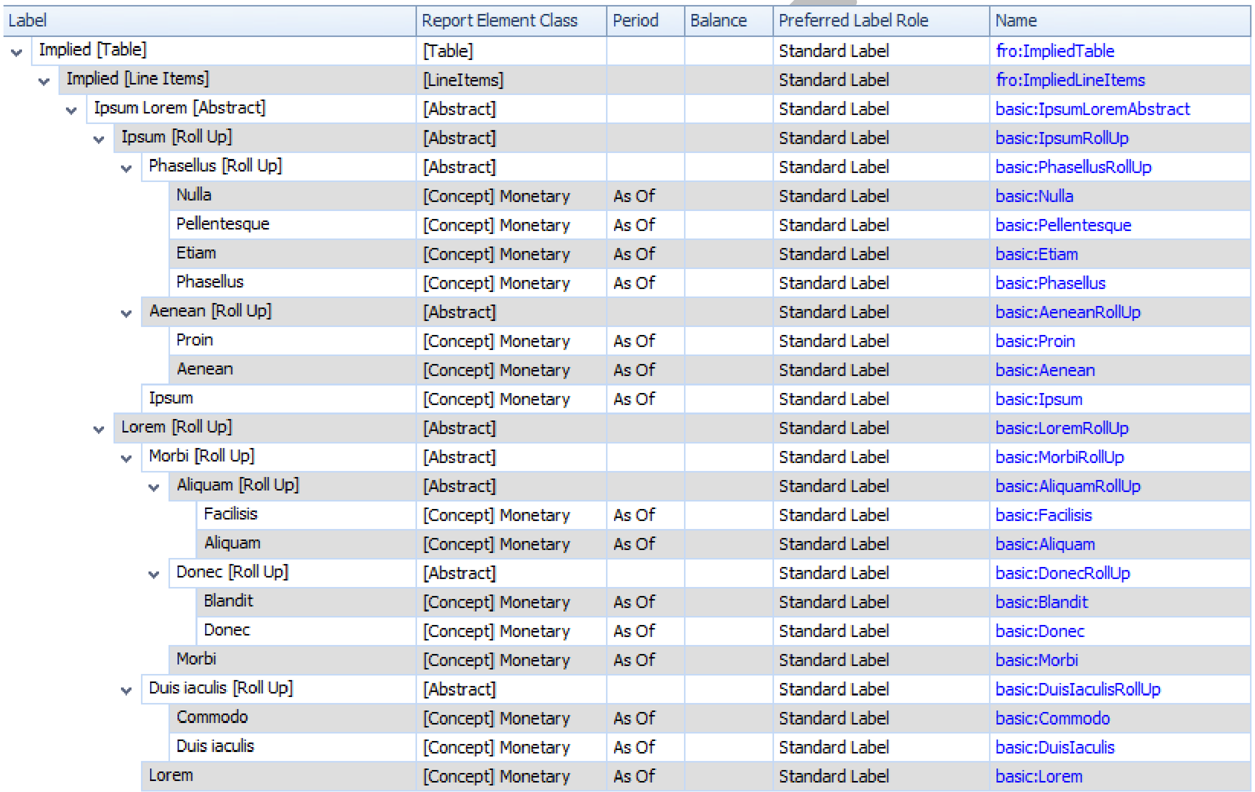
ORANGE cells in the table above show clear modeling errors. If you add all the values in the ORANGE cells you get only 307 modeling errors for a total of about 6,099,406 relationships. That is an error rate of .005%. So, there are very few of the 6,000 companies that are not following the patterns documented within the table.

YELLOW cells are not necessarily considered errors, but they are certainly not best practices either and should be avoided as the relations tend to be illogical or at least a little bit confusing and tend to be interpreted differently by those using that information. For example, what exactly is the logic of having one Concept represented as a child of another Concept? Where exactly is that logic documented? (It is not documented.) What exactly is the difference between two Concepts being represented as siblings or as children of an [Abstract] Concept (there are a lot of those) and one Concept being represented as a PARENT in a relation and another represented as a CHILD of that PARENT Concept?

The point is this. First, there are clearly patterns of logical relationships that exist in XBRL-based reports. Second, there are also clear illogical or pathological relations. Third, it could potentially be argued that some of the ORANGE relations are not technically illogical and one could possibly come up with some explanation that explains why that relation should be allowed. However, this is to miss the point. The point is (a) why are the allowed and un-allowed relations not clearly knowable and (b) there is a method to document allowed and un-allowed relations using XBRL[14].

Allowed and Disallowed Class and Subclass Relations

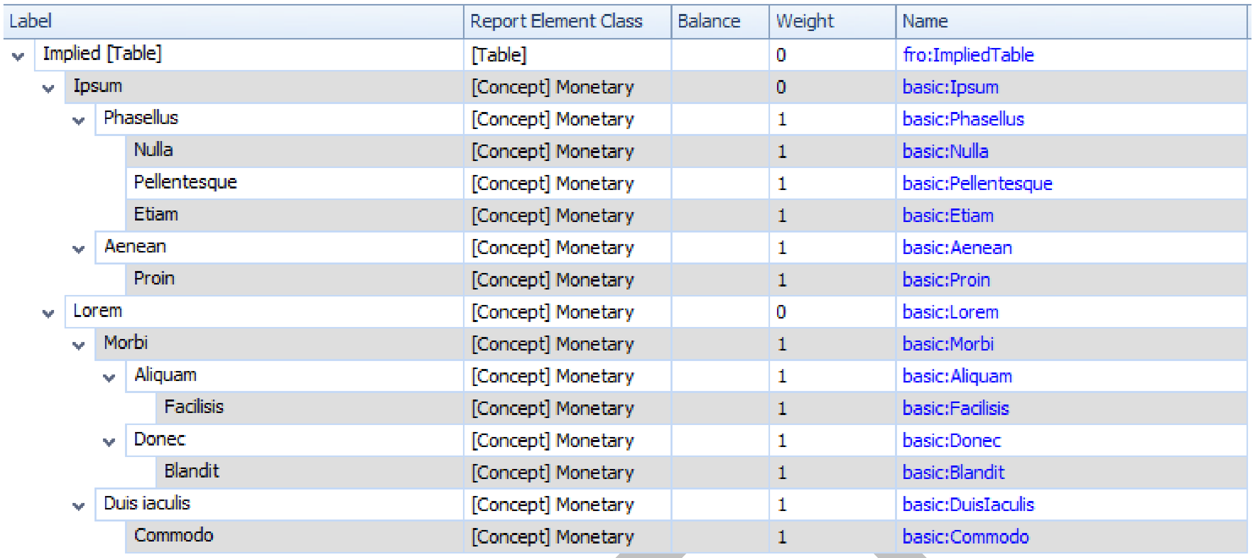
The following graphic shows the set parent-child relations within XBRL presentation relations:



Label	Report Element Class	Period	Balance	Preferred Label Role	Name
Implied [Table]	[Table]			Standard Label	fro:ImpliedTable
Implied [Line Items]	[LineItems]			Standard Label	fro:ImpliedLineItems
Ipsum Lorem [Abstract]	[Abstract]			Standard Label	basic:IpsumLoremAbstract
Ipsum [Roll Up]	[Abstract]			Standard Label	basic:IpsumRollUp
Phasellus [Roll Up]	[Abstract]			Standard Label	basic:PhasellusRollUp
Nulla	[Concept] Monetary	As Of		Standard Label	basic:Nulla
Pellentesque	[Concept] Monetary	As Of		Standard Label	basic:Pellentesque
Etiam	[Concept] Monetary	As Of		Standard Label	basic:Etiam
Phasellus	[Concept] Monetary	As Of		Standard Label	basic:Phasellus
Aenean [Roll Up]	[Abstract]			Standard Label	basic:AeneanRollUp
Proin	[Concept] Monetary	As Of		Standard Label	basic:Proin
Aenean	[Concept] Monetary	As Of		Standard Label	basic:Aenean
Ipsum	[Concept] Monetary	As Of		Standard Label	basic:Ipsum
Lorem [Roll Up]	[Abstract]			Standard Label	basic:LoremRollUp
Morbi [Roll Up]	[Abstract]			Standard Label	basic:MorbiRollUp
Aliquam [Roll Up]	[Abstract]			Standard Label	basic:AliquamRollUp
Facilisis	[Concept] Monetary	As Of		Standard Label	basic:Facilisis
Aliquam	[Concept] Monetary	As Of		Standard Label	basic:Aliquam
Donec [Roll Up]	[Abstract]			Standard Label	basic:DonecRollUp
Blandit	[Concept] Monetary	As Of		Standard Label	basic:Blandit
Donec	[Concept] Monetary	As Of		Standard Label	basic:Donec
Morbi	[Concept] Monetary	As Of		Standard Label	basic:Morbi
Duis iaculis [Roll Up]	[Abstract]			Standard Label	basic:DuisIaculisRollUp
Commodo	[Concept] Monetary	As Of		Standard Label	basic:Commodo
Duis iaculis	[Concept] Monetary	As Of		Standard Label	basic:DuisIaculis
Lorem	[Concept] Monetary	As Of		Standard Label	basic:Lorem

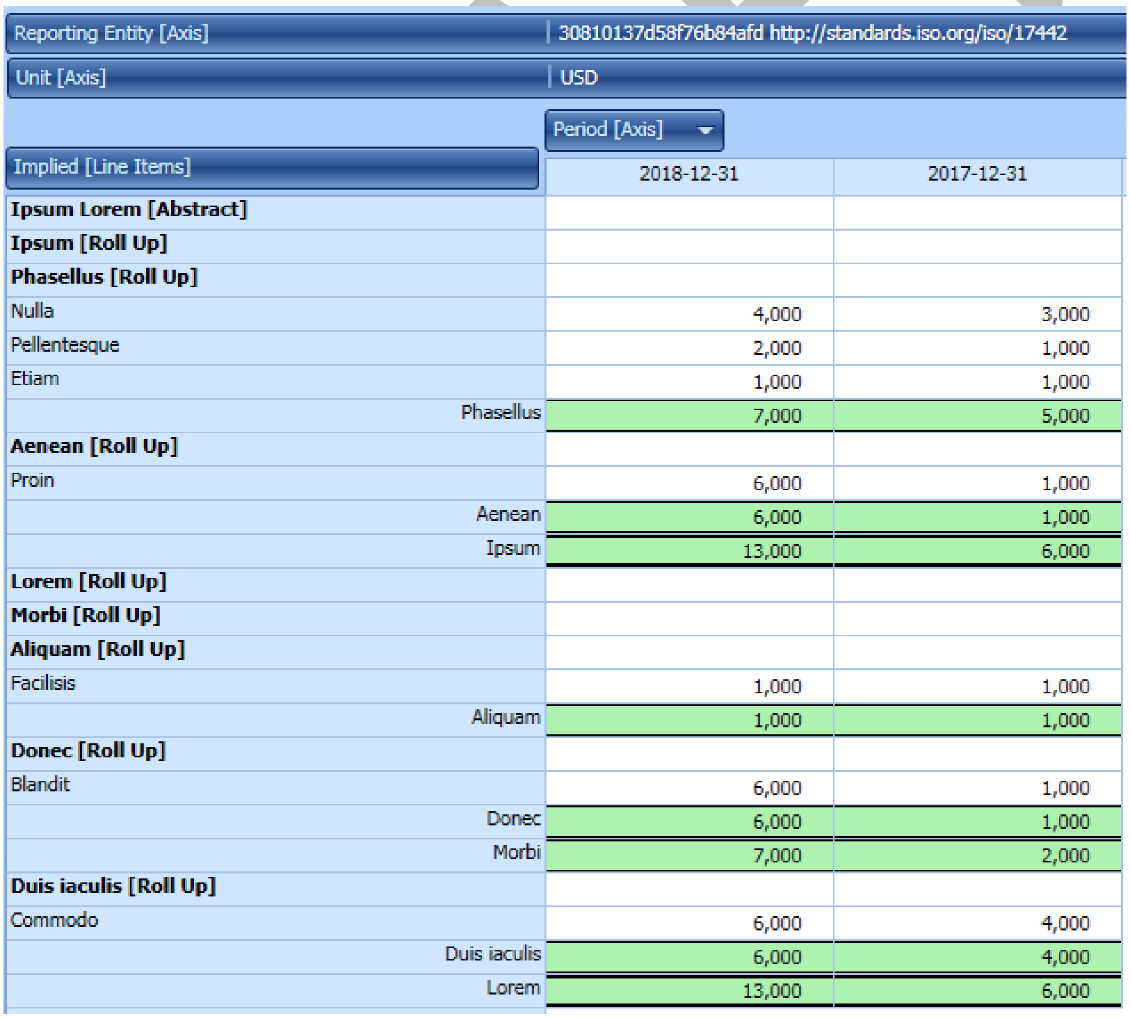
The following graphic shows the set of summation-item relations within XBRL calculation relations:

[14] Here is an XBRL definition linkbase that documents allowed and un-allowed model structure relations within an XBRL presentation relations linkbase, http://xbrlsite.azurewebsites.net/2016/conceptual-model/reporting-scheme/us-gaap/model-structure/ModelStructure-rules-us-gaap-def.xml



Label	Report Element Class	Balance	Weight	Name
Implied [Table]	[Table]		0	fro:ImpliedTable
Ipsum	[Concept] Monetary		0	basic:Ipsum
Phasellus	[Concept] Monetary		1	basic:Phasellus
Nulla	[Concept] Monetary		1	basic:Nulla
Pellentesque	[Concept] Monetary		1	basic:Pellentesque
Etiam	[Concept] Monetary		1	basic:Etiam
Aenean	[Concept] Monetary		1	basic:Aenean
Proin	[Concept] Monetary		1	basic:Proin
Lorem	[Concept] Monetary		0	basic:Lorem
Morbi	[Concept] Monetary		1	basic:Morbi
Aliquam	[Concept] Monetary		1	basic:Aliquam
Facilisis	[Concept] Monetary		1	basic:Facilisis
Donec	[Concept] Monetary		1	basic:Donec
Blandit	[Concept] Monetary		1	basic:Blandit
Duis iaculis	[Concept] Monetary		1	basic:DuisIaculis
Commodo	[Concept] Monetary		1	basic:Commodo

Finally, the following a rendering of the facts which was created by software using the XBRL presentation relations, XBRL calculation relations, and the facts within an XBRL instance:



Reporting Entity [Axis]	30810137d58f76b84afd http://standards.iso.org/iso/17442	
Unit [Axis]	USD	
	Period [Axis]	
Implied [Line Items]	2018-12-31	2017-12-31
Ipsum Lorem [Abstract]		
Ipsum [Roll Up]		
Phasellus [Roll Up]		
Nulla	4,000	3,000
Pellentesque	2,000	1,000
Etiam	1,000	1,000
Phasellus	7,000	5,000
Aenean [Roll Up]		
Proin	6,000	1,000
Aenean	6,000	1,000
Ipsum	13,000	6,000
Lorem [Roll Up]		
Morbi [Roll Up]		
Aliquam [Roll Up]		
Facilisis	1,000	1,000
Aliquam	1,000	1,000
Donec [Roll Up]		
Blandit	6,000	1,000
Donec	6,000	1,000
Morbi	7,000	2,000
Duis iaculis [Roll Up]		
Commodo	6,000	4,000
Duis iaculis	6,000	4,000
Lorem	13,000	6,000

The terms used are Lorem Ipsum[15] dummy text and means nothing. Lorem Ipsum dummy text was used so that biases cannot be projected into the information by using actual terms. This discussion is about the logic of the information.

In this representations above, the XBRL presentation relations, the XBRL calculation relations, and the rendering of the information are consistently saying that "Nulla + Pellentesque + Etiam = Phasellus".

So, what would it mean if someone created a similar representation but placed the line item "Nulla" under the line item "Proin" and therefore made it part of the total "Aenean"?

Is moving the relation that "Nulla" has allowed or disallowed? Is that relation logical or illogical?

The answer to the question does not matter as much as the question itself. How does one articulate that it is OK or NOT OK to move such a relation within an XBRL taxonomy?

Only one of the following must be true:

- It is NOT OK to move the concept "Nulla" to be part of any other concept.
- It is OK to move "Nulla" to be part of "Proin"; both alternatives are allowed.

It cannot be OK and NOT OK at the same time. And so the question is, how do you articulate this information within an XBRL taxonomy? Further, what exactly are the XBRL presentation relations saying when they articulate a "parent-child" relation and how that that relate to the "summation-item" relations articulated within the XBRL calculation relations?

There are many other questions which can be raised here. For instance, the concept "Lorem" is a total. Is it OK to create an extension concept that might be used as an alternative to that total or is it never OK to create an extension for "Lorem"?

One of two things has to be true about concepts:

- It is OK to extend a concept.
- It is NOT OK to extend a concept.

Anchoring and "Wider-Narrower"

The ESMA is introducing interesting functionality within the European Single Electronic Format (ESEF) documents. This functionality was likely created to mitigate issues which were observed

[15] Lorem Ipsum, https://www.lipsum.com/

in XBRL-based reports submitted using US GAAP and IFRS to the U.S. Securities and Exchange Commission.

The issue relates to the perceived or real issue of not being able to identify the nature of extension concepts that are created and used with XBRL-based financial reports. The idea is that an arcrole can be defined “wider-narrower” and an extension concept can be related or “anchored” to a concept that exists within an XBRL taxonomy.

The functionality is somewhat like the W3C's SKOS[16] defines the terms "broader" and "narrower". The SKOS definitions are;

- broader in meaning (i.e. more general)
- narrower in meaning (i.e. more specific)

One question that I have is are "wider" and "narrower" the best terms? Seems like "wider" and "thinner" go together better than "wider" and "broader". The terms "broader" and "narrower" work together. What are most important is the description of what is supposed to occur and the definition of the terms.

Another issue that I have is the following: What is the difference between "wider" and "narrower" as contract to "general" and "special" that already exists in the XBRL “general-special arcrole[17]?

Another term that SKOS uses is "related[18]" which would be very helpful in organizing concepts with lots of concepts. But again, XBRL has the “essence-alias[19]” relationship. It seems like “related” is more of an informal relationship than “essence-alias” which is pretty specific in meaning, it seems, that two things are equivalent. “Related” and “equivalent” do not mean the same thing.

Many people tend to confuse or read meaning (and many times different people read in different meaning) to the notions of “hierarchy” and “association” or “associated”.

What exactly is the benefit of the W3C in SKOS and OWL; and XBRL International to have different semantics for what might be exactly the same functionality? What precisely is the

[16] W3C, *SKOS Simple Knowledge Organization System Primer*, 2.3.1 Broader/Narrower Relationships, http://www.w3.org/TR/skos-primer/#sechierarchy

[17] XBRL International, *Extensible Business Reporting Language (XBRL) 2.1,* 5.2.6.2.1 "general-special" arcs, http://www.xbrl.org/Specification/XBRL-2.1/REC-2003-12-31/XBRL-2.1-REC-2003-12-31+corrected-errata-2013-02-20.html#_5.2.6.2.1

[18] W3C, SKOS *Simple Knowledge Organization System Primer*, 2.3.2 Associative Relationships, http://www.w3.org/TR/skos-primer/#secassociative

[19] XBRL International, *Extensible Business Reporting Language (XBRL) 2.1*, 5.2.6.2.2 "essence-alias" arcs, http://www.xbrl.org/Specification/XBRL-2.1/REC-2003-12-31/XBRL-2.1-REC-2003-12-31+corrected-errata-2013-02-20.html#_5.2.6.2.2

difference between "general-special" in XBRL and class-subclass in OWL[20]? Same for "essence-alias" as contract to "equivalent concept" in OWL? If there is a NEED for a difference, fine. But if the difference is caused by being unconscious of the moving parts here, that is a really bad idea.

Another issue is the XBRL calculation relation "summation-item". What exactly is the difference between "summation-item" which indicates a roll up relationship in XBRL calculations and "wider-narrower". Do they mean the same thing? Should they always be consistent?

The definitions of all these should be coordinated, not created in a vacuum. Further, these "wider-narrower" and anchoring ideas are also very applicable to XBRL-based reports submitted to the SEC. The SEC's definition of "use the narrowest concept" leaves a lot to be desired and there is certainly no specific explanation of how metadata such as "parent-child" and "summation-item" is used to define a "wider" or "narrower" concept in the IFRS or US GAAP XBRL Taxonomies.

Either (a) the ESMA and SEC should have EXACTLY the same functionality or (b) the functionality should be different and the specific differences in functionality should be identified, conscious, and clearly documented. At this point it does not seem that people have not thought through all of these details. There should be no ambiguity here.

One of the issues exists in XBRL-based reports to the SEC and the idea of using "wider" and "narrower" is that there tended to be endless debates. For example, is "Net Income Before Tax" more narrow than "Net Income After Tax"?

Lack of good, sound, clear definitions and examples could clear up the endless debates.

Which of the following applies to wider-narrower?

- The child concept is a component or PART OF the parent concept (we have the XBRL "summation-item" calculation relation that already represents this)
- The child concept is a TYPE OF the parent concept (we have the XBRL "general-special" definition relation that already represents this)
- The child concept is the SAME AS the parent concept (we have the XBRL "essence-alias" definition relation that already represents this)
- The child concept is a PROPERTY OF the parent concept (XBRL currently has no way to represent this relationship.)

[20] W3C, *OWL 2 Web Ontology Language Primer (Second Edition)*, 4.2 Class Hierarchies, http://www.w3.org/TR/owl2-primer/#Class_Hierarchies; OWL 2 Web Ontology Language, 5.1 Classes, http://www.w3.org/TR/owl2-syntax/#Classes

- The child concept represents the CHANGE in an instance concept, for example the change or roll forward in cash and cash equivalents) (XBRL currently has no way to represent this relationship.)

Is the wider-narrower the same as one of these, a combination of these, or something new all together?

So take a step back for a moment. Consider the US GAAP and IFRS XBRL Taxonomies as they exist today. Do you feel they contribute to the ability to understand “wider” and “narrower” concepts or is the representation approach used contributing to the problem? What if the US GAAP and IFRS taxonomy were represented differently; would that perhaps help this issue and make the issue clearer? What would that better representation look like? Has the FASB or IFRS Foundation created any prototypes to help understand this problem and figure out how to fix those problems?

If you are going to “experiment”, then I feel that XBRL International should not be adding this “wider-narrower” arcrole to the LRR. Let the ESMA experiment; no need for XBRL International to get involved at this point. When you want to do this “perfectly” then that is when XBRL International should get involved.

If software vendors don’t implement “wider-narrower” appropriately you really will not get the benefits that are desired. For example, “general-special” and “essence-alias” already exist. Why are they not used?

Text search is NOT the proper way to find the appropriate concepts in an XBRL taxonomy. Relations matter. The type of relationship matters. People are projecting into the meaning of the term “parent-child” relations in XBRL presentation relations.

Why are XBRL definition relations not used in the IFRS or US GAAP XBRL Taxonomies more?

Unraveling the dynamics of what is really going on here is critical to solving the problem correctly.

An extension can be "wider" than elements of the taxonomy but it can, at the same time, be "narrower" than other elements. Which side should you choose? (Can you please provide specific examples of this?)

To decide where to place your extension you look at the calculation or the presentation relationships and see whether it is part of a total or it is a subtotal above (concepts or member). This is nice when all elements are placed in a hierarchy in the taxonomy, but the IFRS taxonomy has Common practice elements which are just a list of concepts outside of the statements hierarchies.

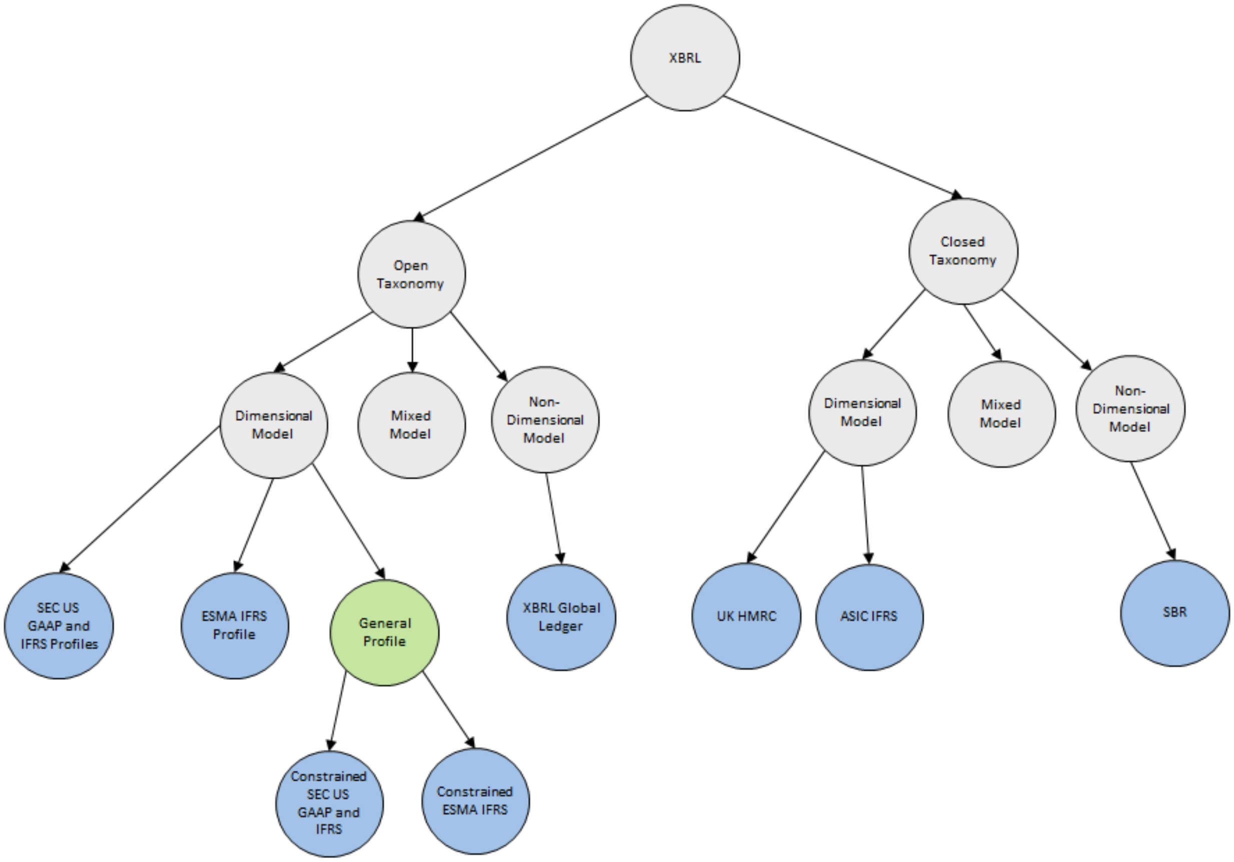
Profiles

One technique to manage extensibility and flexibility is to use the notion of profiles[21]. A profile or application profile is a pattern of XBRL.

The diagram below provides a graph of the broad implementation alternatives related to XBRL. First, a taxonomy can be "open" and allow extensibility or "closed" and not allow extensibility. A taxonomy can (a) use a dimensional model and not allow the use of tuples, (b) use a non-dimensional model and employ tuples, or (c) provide a mixed model which really should be avoided but that is what you end up if the taxonomy creators did not think through what they are doing.

And so the general profile is a well-thought-through implementation of XBRL that follows safe, consistent, best practices. The general profile further constrains the SEC and ESMA implementation profiles (i.e. the general profile disallows some of the things the SEC and ESMA profiles allows, but is still perfectly legal per the SEC or ESMA profiles). The benefit of the additional restrictions is that users are steered away of dangerous practices, the don't have to make decisions that tend to be technical and that they can get wrong, and gaps are filled that the SEC and ESMA implementations did not think through thoroughly.

[21] *XBRL-based Digital Financial Reporting Profiles and General Business Reporting Profile*, http://xbrlsite.azurewebsites.net/2018/Library/Profiles-2018-10-22.pdf



Conclusion

Not providing clear rules related to the allowed relations between report elements, rules related to class-subclass relations, rules related to which concepts can be extended and which cannot be extended are critical to making use of information provided within an XBRL instance. Providing clear rules where extension is and is not allowed is critical to leveraging XBRL's extensibility effectively.

Acknowledgements

Most of the ideas in this document come from discussions and feedback that I received over the past 15 or so years from many, many colleagues who are too numerous to list here. That input was critical to shaping the thoughts expressed in this document. Thank you to the entire XBRL community!