Not Only SQL Analytical Processing (NOLAP) XBRL Application Profile

A safe and reliable implementation of an OLAP-type cube and digital spreadsheet leveraging XBRL

The Not Only SQL Analytical Processing (NOLAP) XBRL application profile is an application profile of XBRL which is 100% compliant with the XBRL 2.1, XBRL Dimensions, XBRL Formula, and Generic Linkbase specifications. The profile follows the spirit of the XBRL Abstract Model 2.0 Public Working Draft. It can be used to create OLAP-type hypercubes or digital spreadsheets of either low or high information fidelity. This document is a non-normative explanation of NOLAP, see the formal specification for normative guidance.

(This started off as XBRLS, then evolved to "the general profile" and not to a NOLAP application profile which is usable generally)

Co-editors:

Charles Hoffman, CPA (charleshoffman@olywa.net)

DRAFT of 2014-07-01

Table of Contents

Introduction

This application profile endeavors to specify an XBRL application profile which will allow for the creation of business reports which are readable by both humans and machines. While presentation is a requirement, pixel perfect presentation are not a requirement.

This document is a non-normative explanation of NOLAP, see the formal specification for normative guidance. This document provides an information description of the Not Only SQL (NoSQL) analytical processing (NOLAP) XBRL application profile (the profile). The profile is compliant with the XBRL 2.1, XBRL Dimensions, XBRL Formula, and Generic Linkbase specifications. The profile follows the spirit of the XBRL Abstract Model 2.0 Public Working Draft. It can be used to create OLAP-type hypercubes or digital spreadsheets of either low or high information fidelity.

The profile makes further restrictions upon the syntax and semantics of the XBRL technical specifications, basically eliminating certain specific aspects of XBRL from being used and more rigorously defines semantics of OLAP-type cubes or digital spreadsheets.

The goal of this profile is to provide a safe, reliable, predictable, robust, consistent, and easy to use approach to making use of XBRL to enable OLAP-type cubes to be exchanged between business systems. The way this goal is achieved is to eliminate unsafe or unnecessary parts of XBRL and to clearly and rigidly define business report semantics.

Another objective of this application profile is to create SEC XBRL financial filings using this application profile. This profile follows the spirit of the US GAAP XBRL Taxonomy Architecture. It is possible to create SEC XBRL financial filings using this more disciplined application profile because this profile is a subset of what is allowed by the SEC Edgar Filer Manual (EFM). However, not all SEC XBRL financial filings would be compliant with this profile. Not all XBRL instances and taxonomies created using this profile would be valid SEC XBRL financial filings.

Other documentation related to this application profile can be found here:

http://nolap.wikispaces.com/

1.1. *Intended audience of this document*

The intended audience of this document is business users and software developers implementing software intended to be used by business users.

The average business user should not need to read or understand this document. Software vendors should embed the information specified within this document within software applications such that the average business user may only comply with these rules.

1.2. Organization of this document

This document is organized to be read linearly, start to finish.

1.3. Terminology

Throughout this document, several words are used to signify the requirements of this specification. These words are capitalized when they should be interpreted as having a strict meaning. The following definitions are taken from RFC2119 and modified so that they are more appropriately worded for use within this standard.

Term	Meaning
MUST	This word means that the definition is an absolute requirement of this specification.
MUST NOT	This phrase, or the phrase "MUST NEVER," means that the definition is an absolute prohibition of this specification.
SHOULD	This word means that valid reasons may exist in particular circumstances to ignore a particular item, but the full implications must be understood and be carefully considered before choosing a different course.
SHOULD NOT	This phrase means that valid reasons may exist in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully considered before implementing any behavior described with this phrase.

Term	Meaning
MAY	This word means that an item is truly optional. One business unit may choose to include the item because a particular marketplace requires it or because the business unit feels that it enhances the product while another business may omit the same item. An implementation which does not include a particular option MUST be prepared to interoperate with another implementation which does include the option, though perhaps with reduced functionality. Conversely, an implementation which does include a particular option MUST be prepared to interoperate with another implementation which does not include the option (except, of course, for the feature the option provides).

1.4. System narrative

Entities create business reports which may take the form of an OLAP-type cube, a digital spreadsheet, or other business report which contains report facts.

A **business report** is a report provided by some entity. A financial report is one type of business report. Expense reports, earnings release, sales analysis, are other types of business reports. Business reports always include at least one component and may include many components.

A **component** is a set of facts which go together (tend to be cohesive and share a certain common nature) for some specific purpose within a business report. For example, a "balance sheet" is a component of a financial business report. "Maturities of long-term debt" is a component of a financial business report.

A **fact** describes a single, observable, reportable piece of information contained within a business report which is contextualized for unambiguous interpretation or analysis by distinguishing characteristics of the fact. Every fact has exactly one value. Every fact must have one characteristic but may have many characteristics.

A **characteristic** provides information necessary to describe a fact or unambiguously distinguish one fact from another fact. A fact has a set of one or many characteristics, the set being a property of the fact, which describes the fact.

A **report building block** is a piece of a business report component. A report building block represents a sensible, logical set of related facts, which follows exactly one **concept arrangement pattern**. One or more report document building blocks comprise a business report component.

A **report document** is an organized set of business report components. Business report components are sequenced or organized in an appropriate flow. A well-constructed digital report document can be presented sensibly and logically as an electronic document (such as a Word or PDF document), a web document (such as an HTML file or Wiki page), an OLAP-type cube, a spreadsheet, or any other visual form including something provided by a dynamic viewing application (such as a pivot table, or drill-down information viewer).

The presentation or view of a digital business report is created by one or more digital report viewing tools (commonly known as rendering engines), which are specifically capable of reading the structured digital format (in this case XBRL) and then creating a structured presentation. It is important to know that different rendering engines may present the same digital business report in different ways. This does not mean that the underlying representation is different, only that the translation of the semantic representation to a visual presentation is different.

Rendering engines are expected to understand concept arrangement patterns and member arrangement patterns which are helpful in creating understandable human readable renderings.

1.5. General high-level requirements

The following is a summary of general high-level requirements:

- **Straightforwardly usable over the Internet**: The business reports are intended to be used over the Internet.
- **Shall support a wide variety of common business user cases**: A wide variety of business use cases should be handled, considering the 80/20 rule is appropriate. It is not a requirement to meet all business use cases.
- **Minimal options**: The number of optional features is to be kept to the absolute minimum, ideally zero.
- **Formal and concise**: The design shall be formal and concise.
- **Readable by both humans and machines**: A spreadsheet should be readable by both humans and machines. Information provide within a spreadsheet should be more a representation of information than presentation oriented. The representation can be presented in sheets, rows, columns, and cells but this is done leveraging information metadata and

commonly understood patterns. 100% pixel perfect renderings are specifically not a requirement.

- **Global standard format with high level of semantics:** The format of the spreadsheet should be a global standard which can provide a high level of semantic clarity.
- **The "model" and the "view" should not be intermingled**: The information and the model should be separate.
- **Business rules separate**: Business rules should be separated from the information. Business rules which are external to the spreadsheet should be allowed for.
- **Managed global standard**: The better spreadsheet should be a global standard under the control of someone like OMG, XBRL International, ISO, Apache OpenOffice, or some other such organization.
- **Provide technical syntax, structural interoperability, but be domain neutral**: A technical syntax should be chosen (XML, JSON, RDF, etc.), spreadsheet or "report" level semantics should be specified to provide a formal "shape" of information, but the semantics of the information within a spreadsheet should be determined by the creators of the spreadsheet. Business domain users information would always fit into the required spreadsheet "shape".
- **Format should allow for versioning, collaboration, etc**: The syntax format should allow for ease of versioning, constructing systems which are collaborative in nature (multi-user).

1.6. *System high-level requirements*

The following is a summary of the high-level system requirements for digital business reports:

- Minimize the probability of ambiguity between what a reporting entity may say and what a user of the report may interpret.
- Maximize safe reuse of information contained within a business report
- Minimize the possibility of errors within the business report
- Maximize the probability of detecting errors using automated processes assisted by software applications

- Maximize the probability that any software which supports XBRL will be able to make use of a business report with no need for adjusting the software

2. Restrictions on XBRL Technical Syntax

The following section summarizes parts which exist within the XBRL 2.1 Specification, XBRL Dimensions 1.0 specification, and XBRL Formula specification which are must not exist within XBRL taxonomies, XBRL linkbases and XBRL instances which comply with this application profile.

All other aspects of XBRL 2.1, XBRL Dimensions, XBRL Formula, and Generic Linkbase are allowed other than those items specifically mentioned in this section.

2.1. Tuples MUST NOT exist.

Tuples can always be detected because elements which define tuples have a substitutionGroup attribute value of “xbrli:tuple”. No such elements are allowed under this profile.

Reasoning: Tuples are not allowed by the US GAAP Taxonomy Architecture and are therefore not allowed within SEC XBRL financial filings. Tuples and XBRL Dimensions tend to provide the same functionality so both are not necessary. XBRL Dimensions provides better functionality.

2.2. Typed members MUST NOT exist.

Typed members can always be detected as they contain the xbrldt:typedDomainRef attribute which defines the typed member. No such attribute should ever be detected within a discoverable taxonomy set (DTS) which makes use of this profile.

Reasoning: Typed members are not allowed by the US GAAP Taxonomy Architecture and are therefore not allowed within SEC XBRL financial filings. Explicit members can be created which serve the same general purpose as typed members.

[CSH: We could allow simple typed members only and only disallow complex typed members as it is complex typed members which cause problems. It may be a good idea because while it is true that explicit members can always be used to articulate what can be articulated using typed members, typed members do has a bit more flexibility which can be useful.]

2.3. Context scenario elements MUST NOT exist.

Context elements which contain a <scenario> element can always be detected. No such element should ever be detected within a discoverable taxonomy set (DTS) which makes use of this profile.

Reasoning: Scenario elements within contexts are not allowed by the US GAAP Taxonomy Architecture and are therefore not allowed within SEC XBRL financial filings. XBRL makes no distinction between dimensions and members which are contained within a <scenario> element and those contained within a <segment> element. Therefore, use of both elements is unnecessary.

2.4. *Precision attributes MUST NOT be provided on any fact within an XBRL instance.*

Precision attributes can always be detected on facts. No such attribute should ever be detected within an XBRL instance which makes use of this profile.

Reasoning: Precision attributes are not allowed by the US GAAP Taxonomy Architecture and are therefore not allowed within SEC XBRL financial filings. The decimals attribute, which is allowed, serves exactly the same purpose as the precision attribute. If both attributes are allowed then it make it necessary to convert from decimals to precision and precision to decimals which could cause interoperability issues.

THESE ARE FROM A DOCUMENT ROLAND CREATED:

Proposed architectural guidelines

A reporter who wants to create a custom extension MUST obey the following guidelines:

1. A maximum of one schema and/or one linkbase can be created;
2. A custom (empty) schema has already been prepared by SBR, and needs to be used;
3. A custom (empty) linkbase has already been prepared by SBR, and needs to be used;
4. In the schema only concepts in the substitutionGroups xbrli:item, sbr:specificationTuple, sbr:presentationTuple, sbr:presentationItem or sbr:member may be created;
5. All concepts created MUST be XBRL valid;
6. All tuples created MUST be on root level;
7. All tuples created MUST NOT contain xs:choice or xs:all children nodes;
8. All concepts created MUST have a standard label in the preferred language of the report;
9. All concepts created MUST occur in the instance and in a relationship specifying the place in the presentation (either through a presentation or table relationship);

10. All concepts created MUST have a definition of their meaning. Plain text is to be provided through the definitionLabel role, references to a definition document (PDF etc.) through a standard reference link;
11. No addition of custom XML schema attributes or elements is allowed, only elements in the listed substitutionGroups;
12. No addition of ELR's, only extension of existing ELR's is allowed;
13. No addition of root parents inside existing ELR's is allowed;
14. No prohibiting of existing relationships is allowed;
15. Numeric custom items MUST be placed in a definitional relationship to existing numeric items:

    a. Custom item is a summation of existing items: custom item is the parent in a D-relationship with the existing items as the children and an arcrole of 'total-breakdown'.

    b. Custom item is a detailing of an existing item: custom item is the parent in a D-relationship with the existing item as its child and an arcrole of 'general-special'.

    c. Custom item is both a summation of existing items but also detailing other existing items: custom item is the parent in a D-relationship with the existing items as the children in the arcroles as stipulated above.

3. Restrictions on Semantics

The following is a summary of additional restrictions explicitly placed on the semantics of business reports articulated using the XBRL technical syntax. This section basically makes things which are legal in XBRL illegal. The reason for imposing these restrictions is they cause irrational, illogical or nonsensical representations when expressed in XBRL.

3.1. Report elements contained within or defined by an XBRL taxonomy MUST clearly be defined such that they can be categorized into one of the following groups of report elements: Hypercube, Dimension, Member, Concept, Abstract.

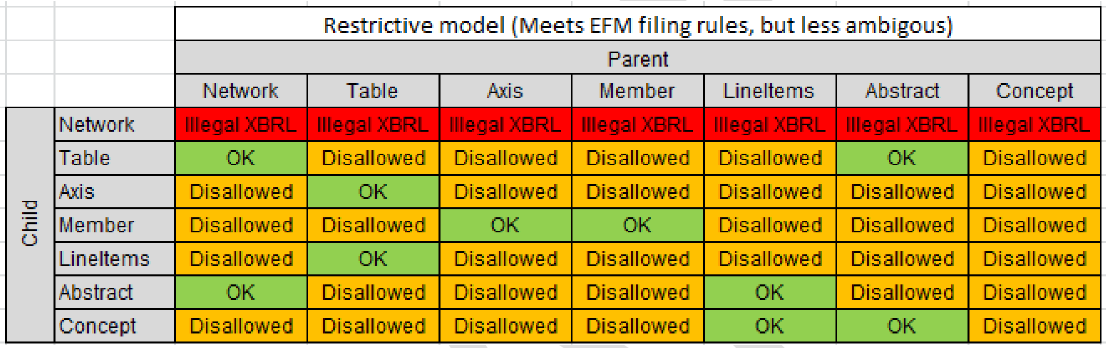
- **Hypercube** – a hypercube can always be detected by the value of the substitutionGroup attribute value of xbrldt:hypercube. Other terms used for hypercube include Table, Cube, Matrix, Array.
- **Dimension** – a dimension can always be detected by the value of the substitutionGroup attribute value of xbrldt:dimension. Other terms used for dimension include Axis or Aspect.
- **Member** – a member can always be detected by the value of the dataType attribute value of nonnum:domainItemType from the namespace identifier http://www.xbrl.org/dtr/type/non-numeric.
- **Primary Items** – a primary items report element can always be detected by the fact that it is the last child of a hypercube within the presentation relations and that it has an abstract attribute value of "true". Other terms used for Primary Items includes Line Items.
- **Abstract** – an abstract can always be detected by the fact that it is not identifiable as a hypercube, dimension, or member and that does have an abstract attribute with the value of "true".
- **Concept** – a concept can always be detected by the fact that it is not a hypercube, dimension, member, primary item, or abstract.

This rule implies that every XML Schema element defined in an XBRL taxonomy schema can be categorized into one of these groups and that the term "report element", or "XML Schema element" or "element" or "XBRL element" should never be used. Rather, the terms hypercube, dimension, member, abstract, or concept should be used to discuss such report elements.

3.2. Report element categories MUST be related in specific ways.

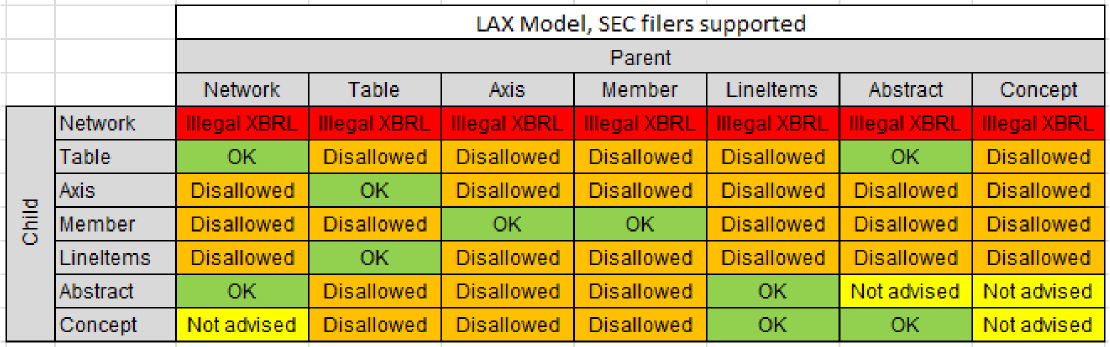
One report element category can only be related to another report element category in very specific ways when represented in XBRL presentation relations. Note that XBRL definition relations are more restrictive than XBRL presentation relations. The intent of this rule is to minimize ambiguity and maximize consistency with XBRL definition relations, particularly XBRL Dimensions relations expressed using XBRL definition relations.

This is a more restrictive relations model, this model is encouraged. This model complies to all SEC EFM rules, but not all SEC filers follow this restrictive model:



		Restrictive model (Meets EFM filing rules, but less ambigous)						
		Parent						
		Network	Table	Axis	Member	LineItems	Abstract	Concept
Child	Network	Illegal XBRL	Illegal XBRL	Illegal XBRL	Illegal XBRL	Illegal XBRL	Illegal XBRL	Illegal XBRL
	Table	OK	Disallowed	Disallowed	Disallowed	Disallowed	OK	Disallowed
	Axis	Disallowed	OK	Disallowed	Disallowed	Disallowed	Disallowed	Disallowed
	Member	Disallowed	Disallowed	OK	OK	Disallowed	Disallowed	Disallowed
	LineItems	Disallowed	OK	Disallowed	Disallowed	Disallowed	Disallowed	Disallowed
	Abstract	OK	Disallowed	Disallowed	Disallowed	OK	Disallowed	Disallowed
	Concept	Disallowed	Disallowed	Disallowed	Disallowed	OK	OK	Disallowed

This is a more relaxed model. Nonsensical relations are disallowed because the relations introduce ambiguity. Other less ambiguous relations are not advised.



		LAX Model, SEC filers supported						
		Parent						
		Network	Table	Axis	Member	LineItems	Abstract	Concept
Child	Network	Illegal XBRL	Illegal XBRL	Illegal XBRL	Illegal XBRL	Illegal XBRL	Illegal XBRL	Illegal XBRL
	Table	OK	Disallowed	Disallowed	Disallowed	Disallowed	OK	Disallowed
	Axis	Disallowed	OK	Disallowed	Disallowed	Disallowed	Disallowed	Disallowed
	Member	Disallowed	Disallowed	OK	OK	Disallowed	Disallowed	Disallowed
	LineItems	Disallowed	OK	Disallowed	Disallowed	Disallowed	Disallowed	Disallowed
	Abstract	OK	Disallowed	Disallowed	Disallowed	OK	Not advised	Not advised
	Concept	Not advised	Disallowed	Disallowed	Disallowed	OK	OK	Not advised

3.3. Hypercubes, dimensions, members, primary items, and abstracts MUST have a periodType attribute value of XBRL equal to “duration”.

PeriodType has no semantics for these types of report elements.

3.4. Hypercubes, dimensions, members, primary items, and abstracts MUST have an abstract attribute value of XBRL equal to "true".

Abstract has no semantics for these types of report elements. XBRL requires hypercubes and dimensions to be abstract. It seems reasonable to therefore require members and primary items to likewise be abstract.

3.5. Hypercubes, dimensions, members, primary items, and abstracts MUST NOT have a balance attribute.

Balance is not an appropriate property for these report elements.

3.6. Relations expressed using XBRL definition relations related to XBRL Dimensions and relations expressed using XBRL presentation relations MUST be consistent.

This rule implies that XBRL definition relations and XBRL presentation relations must never be inconsistent and therefore ambiguous.

4. Explicit Semantics

While the previous section restricts certain specific uses of the XBRL technical format to for the purpose of minimizing the chances of ambiguity and otherwise eliminating irrational, illogical, nonsensical representations of information; this section articulates specific business report meaning or semantics.

4.1. Networks or hypercubes MUST articulate clear business meaning. As such all hypercubes MUST be isomorphic (carry one meaning) or no meaning at all (one standard hypercube is used and networks carry report component semantics.

[CSH: This needs work.]

4.2. Report components MUST be represented using the same network URI across all XBRL presentation, XBRL calculation, XBRL definition, and XBRL Formula related relation networks.

In essence this means that if a report component is expressed, then the network identifier of that report component must be the same for each set of presentation, calculation, definition, and XBRL Formula networks which express information for that report component.

[CSH: This needs work.]

4.3. Report element names and IDs MUST NOT carry semantics.

The meaning of report elements is provided by the report element documentation, labels, references, relations, and business rules expressed.

4.4. The XBRL context element entity identifier and scheme MUST identify the entity issuing the report.

Use dimensions to provide any other information deemed necessary to characterize a reported fact.

4.5. The XBRL context period MUST indicate the calendar period of a reported fact.

Use dimensions to provide any other information deemed necessary to characterize an period related characteristic of a reported fact.

4.6. *Dimensions MAY be categorized into one the following patterns: semantic categorization, content association, temporal or time-based.*

Axis can be group into the following patterns:

- **Semantic categorization**: Reporting entity [Axis], Legal entity [Axis], Concept [Axis], Business segment [Axis], Geographic area [Axis], Operating activities [Axis], Instrument [Axis], Range [Axis]
- **Content association**: Reporting scenario [Axis],
- **Temporal or time-based**: Calendar period [Axis], Report date [Axis], Fiscal period [Axis]

Characteristic (all of these are candidate categories for sub-characteristics or classes of characteristics)

4.7. *Members of a dimension can be arranged within one of the following member arrangement patterns:*

The relations between the members of a dimension can be organized into member arrangement patterns: composition, aggregation, wholeness.

- **Whole-part:** [Put the whole-part stuff here]
- **Composition:** Some single thing or finite set of things. (Infinite sets would never be reported)
- **Aggregation**: pieces of some whole and the complete list of parts of that whole. [CSH: Need to deal with subtotals or hierarchies within an aggregation. Disallow them?]

[CSH: This needs work. Create the arcrole definitions.]

Basically, the arcrole "http://www.xbrl.org/2003/arcrole/parent-child" used to communicate that there is in fact some sort of relationship leaves open to interpretation exactly what that relation is and what it means. While what is expressed might be clear to those who use the "parent-child" relationship to express something; the intent tends to not come through, be misinterpreted, be inconsistent because of different people working on different areas of a taxonomy, and in general leads to confusion.

Providing some general examples helps one understand why this is a problem. The document A Taxonomy of Whole-Part Relations goes into detail. This presentation (on slide 9 of that document) explains the difference between a taxonomy and what the author calls a “partonomy” and provides these general categories of relations:

- **component-integral object**: for example (pedal – bike)

- **member-collection**: for example (ship – fleet)
- **portion-mass**: for example (slice – pie)
- **stuff-object**: for example (steel – car)
- **feature-activity**: for example (paying – shopping)
- **place-area**: for example (Everglades – Florida)

These arcroles are defined here:

http://www.xbrlsite.com/2013/BusinessReportOntology/GeneralBusinessReportingProfile-arcroles.xsd

4.8. Primary items MUST be arranged within one of the following concept arrangement patterns: roll up, roll forward, adjustment, variance, complex computation. All other concept arrangements of primary items will be considered a hierarchy.

The following is a summary of and further explain these concept arrangement patterns:

- **Roll up**: Fact A + Fact B + Fact C = Fact D (a total)
- **Roll forward**: Beginning balance + changes = Ending balance (this is sometimes called a "movement analysis" or BASE pattern; beginning balance + additions – subtractions = ending
- **Adjustment**: An adjustment reconciles an originally stated balance to a restated balance between two report dates; Originally stated balance + adjustments = restated balance
- **Variance**: A variance is a change across a reporting scenario. Actual amount - Budgeted amount = variance
- **Complex computation**: A complex computation is a type of concept arrangement where facts are related by some computation other than a roll up, roll forward, adjustment, or variance. For example, Net income / Weighted average shares = earnings per share. These can always be detected by the existence of XBRL Formulas.
- **Hierarchy**: A hierarchy is a type of concept arrangement where facts are related in some way, but not mathematically.

Each report component SHOULD be organized into identifiable and discrete concept arrangement patterns.

NOTE: As new concept arrangement patterns are identified the list of supported patterns will be expanded.

4.9. All computations or business rules which the creator of the business report desires a user of the report to understand and MUST be expressed using XBRL calculations (roll up) or XBRL Formula (all other computations).

All computations, which are part of the accounting concept arrangement patterns, can be automatically generated by software as XBRL Formula based business rules which enforce the accounting concept arrangement patterns. One exception to this is the complex computation pattern which could literally be any computation and therefore this is impossible to automate.

XBRL Formulas are preferred to XBRL calculations in most situations.

5. Expressing or Extending Domain Semantics

[CSH: This entire section needs work.]

This section provides a mechanism for controlling extensibility of a taxonomy. This section describes rules which must be followed when creating high-fidelity digital business reports.

5.1. Each domain which desires to allow extensibility and to explicitly control that extensibility MUST create a set of core report elements into which each base taxonomy concept and any extension taxonomy report element MUST fit.

Extensions must extend some existing core domain concept.

For example, for the domain of financial reporting the core concepts exist: Assets, Liabilities, Equity, Revenue, Expenses, Gains, Losses, Investments by owners, Distributions to Owners, Comprehensive Income. To those core elements, the following core elements are also added: Policy, Disclosure.

A base taxonomy and any taxonomies which extend this base MUST assign a core report element to each extension concept to indicate which report element the extension report element is extending.

This is achieved by using XBRL definition links which the arcrole “class-subclass”.

[CSH: This needs work.]

5.2. Each extension taxonomy MUST assign any extension report element to a core report element or a report element of the base taxonomy.

As with base taxonomies, extension taxonomies must assign each extension concept to an existing core concept.

This is achieved by using XBRL definition links which the arcrole “class-subclass”.

[CSH: This needs work.]

5.3. Report elements MUST be grouped into one of four categories: concept required, concept optional, subclasses allowed, extension allowed.

- **Concept required**: it would make no sense to extend concept. For example, there are obvious examples of such concepts like dei:EntityRegistrantName and dei:DocumentFiscalYearFocus. It absolutely, 100% makes zero sense to allow extension of such concepts.
- **Concept optional**: it would make no sense to extend concept. This is similar to the category above, but the concept is NOT required to be reported, such as dei:TradingSymbol, or the concept would only be reported if the filer actually had the concept, such as us-gaap:InventoryNet.
- **Extension NOT allowed, subclass allowed**: For some concepts, it makes a lot of sense to allow a filer to add a subclass for that concept or as some people think about it, to add a "child". But, it makes no sense to extend the concept. So again take the concept us-gaap:InventoryNet. It is hard to imagine the need to provide for some other concept "my:InventoryNet".
- **Extension allowed**: Suppose that some SEC filer wanted to disclose carbon credit information in their financial report but the US GAAP XBRL Taxonomy contained no such concepts. Pretty clear than an extension concept could be created. Likewise pretty clear that if a filer needs subclasses, children, or other stuff those should be allowed for also. Basically, if something clearly does not exist and a filer needs it, creating an extension should be allowed in such cases.

For more information see:

http://xbrl.squarespace.com/journal/2014/9/19/phenomenon-points-to-need-for-global-standard-way-to-define.html

6. Report Lists

When using more than one business reports, some sort of list of reports contained within a report repository is necessary. The following section specifies how to create such a report list using RSS and RDF syntax.

[CSH: This is similar to the SEC XBRL document RSS feed. To do]

TO DO: Units Registry:

http://xbrl.org/units-registry