XBRLS Architecture Specification

An open public standard. Note that XBRLS is not associated with XBRL International

http://openstandards.org

Draft, dated 2008-04-18

This version: XBRLS-Specification-2008-04-01.doc

Conformance Suite: XBRLS-Conformance-2008-04-18.zip

Meta Pattern Examples: XBRLS-MetaPatterns-2008-04-18.zip

Comprehensive Example: XBRLS-ComprehensiveExample-2008-04-18.zip

Business Use Cases: XBRLS-BusinessUseCases-2008-04-18.zip

Repository Example: (Under construction)

Relational Data Model: (Under construction)

Editors/Authors

Name	Contact	Affiliation
Cliff Binstock	cliff.binstock@CoyoteReporting.com	Coyote Reporting
Charles Hoffman	charles.hoffman@UBmatrix.com	UBmatrix
Mark Montoya	MMontoya@FDIC.gov	FDIC
Rene van Egmond	rene@c3ismc.com	XBRL Consulting Partners LLC.

Abstract

The purpose of this document is to specify the architecture of XBRLS. This document is intended for technical users implementing software for business users. All rules defined in this architecture specification should be embedded within software applications.

Status

Circulation of this Draft is restricted to the authors of this document and others invited by the authors to comment on this document. Recipients of this draft are invited to submit comments to its editors and to submit notification of any relevant patent rights of which they are aware and to provide supporting documentation.

GNU General Public License (GNU GPL)

Table of Contents

1 Introduction

The purpose of this document is to specify the architecture of XBRLS. This document will not justify the need for, explain, or otherwise evangelize XBRLS but rather will be limited to specifying the XBRLS architecture so that it can be implemented within software applications.

XBRLS is 100% compliant to the XBRL Specification. In order to achieve its objectives (enhancing information comparability, transparency and consistency of metadata modeling), XBRLS restricts the use of certain parts of XBRL.

For background information relating to why XBRLS is needed, please see the white paper, "*XBRLS – How a simpler XBRL can make a better XBRL*". This document can be found at the following URL: http://xbrl.squarespace.com/xbrls

1.1 Use of XBRLS Compliance Rules

It is important to understand the intentions we have with these compliance rules. While it is certainly possible to use the rules after a taxonomy has been created and subsequently verufy if it is compliant with the XBRLS specification, the primary intent of the rules is to provide a formal specification of the content and relationships between the various aspects of the XBRL specification. The purpose of the rules is to remove unnecessary options from the process of defining XBRL taxonomies thus inhancing consitency of XBRL metadata modeling.

For example, rather than allow the user to set a value and then have the software explain what the user did, the rules will infer taxonomy constructs based on the provided information. Thus the intent is to prevent the user from doing the "wrong thing" in the first place.

A good example is one of the more complex meta-patterns, the [Movement]. A wizard can be created to "frame" an entire movement analysis, guiding the user in the creation of a correct movement analysis. This is as opposed to having an XBRL tool that allows the user to do anything and then validating what the user has done, telling them they do not have their movement correct.

[CSH: This needs work, but I think this is important to communicate.]

1.2 Turning XBRL into a Different Problem

The way that most XBRL tools work today is that they mirror the XBRL specification. Therefore, if you want to create, say, an XBRL taxonomy, you have to intimately understand the XBRL Specification. This is a problem for business users and all those who are not XBRL experts. The current XBRL taxonomy creation tools allow so much variability in what you can do that the chances are big that you will not create what you really want to create.

XBRLS changes the entire paradigm. XBRLS based tools do not focus on creating XBRL rather they will let you create only specific meta-patterns. The meta-patterns are very close to the concepts we are used to deal with in a business reporting context and consequently, business users need to know less about XBRL. The reasone is that now the software tools contain the required knowledge about the XBRL specification, relieving the user from acquiring that highly technical know-how. Providing a user interaction model closer to the business reporting process also provides a substantially better user experience. As an additional benefit the meta-patterns are vastly easier for software applications to constrain, thus true value-adding software creation is easier.

2 Requirements and Other Motivating Principles

The following is a summary of requirements and other motivating principles which guide the creation of the architecture of XBRLS.

2.1 Requirements

The following is a summary of the requirements of XBRLS.

Requirement	Reasoning
REQ-01: XBRLS MUST be 100% XBRL compliant.	It is not a goal of XBRLS Architecture Specification to replace XBRL. Rather, it is intended to be a methodology for design of XBRL in metadata architecture for the business reporting domain, similar to how the US GAAP Taxonomy Architecture defines an application profile.
REQ-02: Maximize flexibility, but within bounds.	It is better to be provide minimally required flexibility and then loosen the restrictions as needed, rather than be too flexibility and cause confusion that will be very hard to correct later on in the process.
REQ-03: Minimize the "moving parts".	When one thing will do, why have two? There is no reason for instance to have both a decimals attribute and a precision attribute in XBRL Instance Documents when they serve the same purpose.
REQ-04: If at all possible, do not have rules which would make the US GAAP Taxonomy Architecture NOT complant to the XBRLS.	The US GAAP Taxonomy architecture already eliminates many (if not all) the language features of XBRL which are not allowed in this architecture.
REQ-05: Decisions should be driven by needed semantics, and NOT by syntax.	Many confuse syntax with semantics. There is a need to express certain semantic meaning with XBRL. Arguing over a specific syntax for it representation is unnecessary. Having multiple syntaxes for expressing the same thing is not acceptable.
REQ-06: Minimize the cost to implement XBRL within a system.	Through simplification of metadata specification and systems development, XBRL deployment costs can be reduced.
REQ-07: Maximize the ability of XBRLS to work with existing relational databases and business intelligence applications.	Most of today's implemented solutions are based on relational database engines. These relational databases are moving to include XML functionality. XBRL/XBRLS must fit into existing systems.
REQ-08: Minimize deviations in terminology from set standards, but use terminology with which a business user will be comfortable.	Using XML Schema, X-Link, and XBRL terminology should be minimized. Rather terminology which is more familiar to a business user should be used when there is a choice.

Requirement	Reasoning
REQ-09: Maximize the ability of XBRLS to work with existing multidimensinoal analysis, data warehousing, and cube-based systems.	Make XBRLS work with current business infrastructure for data analysis and decision support.
REQ-10: XBRLS must be easily convertible to other dialects of XBRL where the XBRLS language semantics permit.	For example, imagine a financial institution with operations in the US and in Europe which has to file with one regulator using the COREP taxonomy, another using FINREP, with the FDIC and with the SEC. This will allow a business to maintain their own internal taxonomy their way and render it as other XBRL dialects as necessary.
REQ-11: The taxonomy must contain enough information for an acceptable human rendering of the data from information within the taxonomy and instance document prepared using the taxononmy without having to use any additional information. This rendering must be usable by a typical business user.	Humans need to have a basic ability to input and view instance document information in a usable form. It is not acceptable to require each user to create their own rendering for input/output. While any taxonomy or instance document data can be rendered in literally any form using information or meta data external to the taxonomy or instance, a "one to many" minimally acceptable rendering must be derivable for business users from taxonomy and instance document information by software applications.

Fundamentally, XBRLS is intended to add nothing to XBRL. Rather, it proposes a "core" dialect of XBRL that provides the necessary features as a best practice for creating XBRL Based metadata and business reports.

If a user can live with or "make due" within the constraints outlined above, their lives will be easier, their costs will be reduced, and their XBRL will better fit into the most likely future of their XBRL enabled systems.

2.2 Other Motivating Principles

The following is a summary of other principles which, while not necessarily a formal requirement, motivated the decisions made in devising XBRLS.

Principle	Reasoning and Explanation
PRI-01: Complexity cannot be removed, but it can be moved.	Complexity can never be removed from a system. However, complexity can be moved. The complexity of XBRL should be hidden from the business user. This can be achieved by removing unnecessary flexibility, which decreases the "responsibility" of the business user to make choices or to understand certain things.
PRI-02: XBRL is a method of expressing information semantics. The purpose of the XBRL format is to support information exchange. XBRL is not intended to be an archival data storage format.	XBRL is not an archival data storage format, it is a global standard format for expressing information semantics to facilitate the (fully) automated exchange of information.

Principle	Reasoning and Explanation
PRI-03: Trying to solve a data modeling problem should be left to data modelers. Including presentation type information within a data model tends to cause poor data modeling choices.	Data modelers should do data modeling. Application users do not specify database schemas.

3 Summary of XBRL Components NOT used

The following is a summary of the components of XBRL which are disallowed from use in XBRLS and the reasoning for not allowing the component. This information is summarized by XBRL Specification and as best as possible cross referenced to these specifications.

Specification	Topic	Explanation and Reasoning
XBRL Specification , Instance (sections 4.7.4 and 4.7.3.2)	Context: segments, scenarios	Only use XBRL Dimensions to articulate the content of segments and scenarios, excluding the use of XML Schema-based contextual information allowed by sections. Furthermore, mixing XML Schema based-contextual information and XBRL Dimensions is technically dangerous. **REASONING**: XML Schema-based contextual information is too flexible, as there is no way to articulate hierarchy or constrain XML schema based contextual information. XBRL Dimensions achieve all these. Precedents for this approach are the COREP taxonomy and US GAAP Taxonomy that use this approach.
XBRL Specification, Instance	Fact Value: precision	Use ONLY the decimals attribute, precision MUST NOT be used. **REASONING**: Precision and decimals provides exactly the same functionality. There is no reason for both, particularly since that during analysis one approach will have to be converted to the other approach when data is analyzed. Precedent for this is FRIS section 2.8.11.
XBRL Specification, Taxonomy	Elements: tuples	Tuples are not allowed. **REASONING**: There are a number of negative characteristics of tuples. (a) Tuples reduce extensibility. (b) Tuples express meta-data within an instance document rather than in a taxonomy. (c) Tuples provide no way to articulate key values reducing comparability. (d) Tuples cause many issues relating to using items intended to be used within tuples outside those tuples, see FRIS section 2.8.3. (e) XBRL Formulas do not support tuple creation. Tuples and XBRL Dimensions could both be used in many cases to express meta-data. However, the XBRL Dimensions approach offers additional desirable characteristics such as the ability to express key values. Having two approaches is considered a negative. Additionally, precedent for this approach is the approaches of COREP, FINREP and the US GAAP Taxonomy to make no use of tuples.

Specification	Topic	Explanation and Reasoning
XBRL Dimensions	Typed Members	Typed members (simple or complex) are not allowed. **REASONING**: Typed members are not allowed because: (a) typed members create significant implementation issues for software developers; (b) the needed functionality provided by typed members can be provided via explicit members; (c) hierarchical typed members cannot be created; (d) typed members provide metadata within instance documents rather than within a taxonomy. Precedent for this decision is also that the US GAAP Taxonomy contains no typed members.
XBRL Specification, Taxonomies	Weight	The weight attribute value of calculations MUST be either "1" or "-1", no decimal value between the two is allowed. **REASONING**: No taxonomy has ever used anything other than 1 or -1 for the value of weights. Apportioning using calculations is not a use case that will be supported. XBRLS does not compute values; its "calculations" articulate constraint relations.
XBRL Specification, Taxonomies	Annotation, Documentation	Each schema and each linkbase MUST provide documentation that describes the contents of the file that is readable by a computer application. **REASONING**: Users should not have to rely on the file names for knowing what is inside a file. Applications should have access to this documentation that is helpful to business users.
XBRL Dimensions	Open Hypercubes	Open hypercubes are NOT allowed, only closed hypercubes are allowed.
XBRL Dimensions	notAll	Only "all" has-hypercube arcroles are allowed, "notAll" is not allowed. **REASONING**: In multi-dimensional analysis, this feature does not generally exist. This feature is difficult to implement and is not proven to be working correctly in existing XBRL processors (i.e., it is known to have issues). This may be allowed at a later time. Also, this can always be added via an extension taxonomy to enable this type of constraint.

Specification	Topic	Explanation and Reasoning
XBRL Specification, Instance	Context: entity identifier, entity scheme	Although not required when using XBRLS, it is highly encouraged that the entity scheme and identifier be "held static", or synchronized with an explicit member and rather have XBRL Dimensions be used to articulate entity information, perhaps with an XBRLS "Entity [Axis]" dimension. The "entity identifier" and "entity scheme" portion of a context SHOULD NOT be used. Rather, the "entity identifier" and "entity schema" are STATIC (i.e., dummy values in order to pass XBRL validation), using constant values. The information articulates relating to the entity identifier and entity scheme are moved to an XBRLS specific taxonomy that makes use of XBRL Dimensions to communicate this information. **REASONING**: The reasons that the entity identifier and entity scheme are not used is because (a) there is no way to articulate a hierarchy of entity identifiers/schemes within XBRL; however, such a hierarchy CAN be articulated if this information is defined in XBRL Dimensions; (b) there is no way to attach one or more labels to an entity identifier/scheme, whereas this can be done using XBRL Dimensions; (c) this approach moves the articulation of metadata from the instance document to the taxonomy where other metadata is articulated.
XBRL Specification, Instance	Context: period	Although not required when using XBRLS, it is highly encouraged that the period context be "held static", or synchronized with an explicit member and that XBRL Dimensions be used to articulate this information, perhaps with an XBRLS "Period [Axis]" dimension. Use XBRL Dimensions to articulate this XBRL quasi dimension. **REASONING**: There is no way to express a hierarchy of periods. Whereas it is possible to create some hierarchy as the hierarchy of period information is commonly known, there are other hierarchies that are not able to be articulated. The best example of this is the "fiscal period" which is commonly used within financial reporting.

4 Overview of Architecture

The following is a summary of the XBRLS architecture and reasoning behind this architecture.

- **Use no tuples.** Tuples are an unnecessary syntax within XBRL as XBRL Dimensions provides 100% of the functionality provided by tuples, provides additional functionality which tuples does not provide, and XBRL Dimensions-based approaches to articulating the complex concepts which tuples provide is vastly more flexible. It is the consensus of the editors of this document that tuples will, or should, be removed from XBRL 3.0. The move from XBRL 2.0 to XBRL 2.1 which changed the approach for articulating the complex content using the definition linkbase (in XBRL 2.0) to the use of XML Schema content model (in XBRL 2.1) has proved to be a dead end. XBRL Dimensions operates in a similar manner to tuple definition in XBRL 2.0. Note also that XBRL Formulas do not support tuple creation.
- **ONLY XBRL Dimensions can be used as segment context content**. In order to have comparability, there must be some specification driving the content of the and <scenario> context information. Creating such a specification is difficult. XBRL Dimensions is such a specification. It allows for: (a) constraint of contextual information, (b) articulation of hierarchical relations within that information, (c) XBRL Formulas makes use of XBRL Dimensions well. Mixing XBRL Dimensions type contextual information and XML Schema based contextual information is only asking for trouble, there are too many unknowns. As such, XBRL Dimensions is the only approach to articulating this contextual information.
- **The Scenario context element is not used.** To simplify the processing of instance documents the contextual information about information segments and scenarios are all place in the element in the form of XBRL dimensions.
- **All XBRL Dimensions information is placed within the segment context**. There is no reason for a user to have to decide if XBRL Dimensions information should go into the or <scenario> context component. In XBRLS, all contextual information relating to XBRL Dimensions will be placed into the context component
- **EVERY concept MUST participate in an XBRL Dimension hypercube and all hypercube are closed**. Inconsistent used of some concepts being reported within Dimensions and others outside dimensions in the same instance document makes automated handling of XBRL information unnecessary difficult. As such, EVERY reported concept within a "Standard Simplified XBRL Architecture" compliant instance MUST participate within an XBRL Dimension.
- **[RAVE: Re-word]**
  No use of typed members within XBRL Dimensions. XBRL Dimensions typed members can basically have any content. As such, this creates the same problems for complex typed members (typed dimensions). In addition, building a user interface for complex typed members is near impossible. Simple typed members are less of a problem except that the meta data (a) cannot be arranged in a hierarchy, (b) the meta data is stored in the instance document rather than the taxonomy and (c) there is no way of constraining key values (i.e. undesired duplicates could be created).
- **NEVER use the precision attribute, rather use the decimals attribute**. The two attributes within an XBRL instance document serve exactly the same purpose. It is possible to convert from decimals to precision; but impossible to convert from precision to decimals. Removing the option while retaining the semantic meaning is served with allowing only the decimals attribute
- **Do not use similar-tuple definition arcrole**. There is no need for this role as tuples are not used.
- **The notAll has-hypercube arcrole must not be used**. The notAll arcrole is complex to implement, it is hard for most business users to understand how to use,

and current multidimensional software does not implement such a feature. Other means can be used to achieve the validation offered by this feature such as formulas.

- **The calculation linkbase weight attribute MUST only have a value of 1 or -1**. Using other values is not allowed.

5 Domain Model

The domain model articulates the business needs from the system based on the process requirements, domain stakeholders and otherwise articulated business needs that have driven the design of XBRLS.

5.1 Overview of Domain Model

The focus for the domain model is the business system and not just the technical sub-system; it is the entire system. A holistic perspective is taken and the two main drivers for the solution design are to maximize benefit and minimize cost. The domain model considers the "80/20 rule" and does not attempt to address the needs of all possible users. This would make the XBRLS approach again so complicated that the barriers to entry are too overwhelming for the typical business user. Rather, the approach is taken to give the majority of business users what they need. If other users need more features from the XBRL Specification, the system can be supplemented on their dime.

The focus of the XBRLS dialect of XBRL is for financial reporting for which, we believe 80% or more of all business reporting use cases will be met. The spectrum of use cases ranges from a one page form which cannot be extended all the way to a complete 10-K filing (all parts including the MD&A) by a US public company to the Securities and Exchange Commission.

This section summarizes the domain model for financial reporting. This domain model is not only about filing information with a regulator, but rather considers the business from a holistic perspective. The following are a set of statements about business reporting and metadata management and IT solutions based on XBRL and XBRLS.

1. Businesses report to more than one regulator generally.
2. Businesses store data in relational databases, data warehouses, Excel spreadsheets, Word documents, and other locations.
3. The relationship between a subsidiary and a parent is much like the relationship between a reporting entity and a regulator.
4. It is preferable to not have to rip out legacy systems simply to make data exchange convenient or possible.
5. Pretty much every taxonomy will be extended.
6. Many features of business reports are driven by the historical paper based business report. It is not always a good idea to bring paper-based reporting practices into the new paradigm of electronic business reporting and interactive data. Sometimes it may have to be, however the choice should be conscious.

5.2 Domain Stakeholders

The following is a summary of domain stakeholders:

- Standard setters and regulators
- Business users (accountants, other preparers, auditors)
- Software vendors
- Analysts (investors, regulators)

5.3 Business Use Cases

The following is a summary of business use cases that must be supported by XBRLS.

#	Use Case Name	Explanation/Example
1	Simple Hierarchy	Financial Highlights
2	Hierarchy	Accounting Policies
3	Simple Calculation	PPE, Net/Gross Calculation
4	Nested Calculation	Balance Sheet, disclosures
5	Inverted Calculation	Income Statement, Cash Flow Statement
6	Multiple Calculations	Receivables breakdown
7	Simple Movement	Movement in Land
8	Complex Movement Using Items	Movement Property, Plant and Equipment
8a	Complex Movement Using Axis	Movement Property, Plant and Equipment
9	Simple Compound Concept	Director Compensation
10	Repeating Concept	Subsequent Events
11	Multiple Periods	Leaseholds
12	Movement in Compound Concept	Share Options
13	Nested Compound Concept	Related Party Transactions
14	Reconciliation of Balance	Reconciliation of Cash
15	Text Block	Director Compensation
16	Restatement	An accounting restatement
17	Reissue Report	Reissuing a previous report
18	Reclassification	Accounting Reclassification
19	Prose	Management discussion and analysis
20	Comment	XBRL Footnote
21	Sales Analysis	A pivot table
22	Segment Breakdown	Reporting segment information
23	Interim Reporting	Reporting interim information
24	Composite Keys	Rows of data with multiple key values such as a portfolio of investments

See the XBRLS Business Use Cases for more detail on these business use cases. Each of these business use cases must be handled by XBRLS.

5.4 Change/Life Cycle

XBRL Versioning will be used when this becomes available. There will be no problems as XBRLS is a subset of XBRL.

5.5 Data Modeling versus Data Presentation

Data modeling and data presentation should not be confused. Many times inexperienced data modelers fall back on how data is presented in order to model data which results in poor data modeling decisions.

5.6 Paper-based versus Electronic-based Reports

Paper-based business reports and reporting practices have certain characteristics which are a result of the fact that the report is immutably expressed on a two dimensional piece of paper.

"Electronic-paper" based reports likewise have constraints such as those imposed by the rows and columns of a two dimensional spreadsheet. A relational database or a data cube has other issues, mainly how to render three, four, or five dimensions into a form which is consumable by the human brain.

[CSH: Consider that string theory has 11 dimensions. I have heard that the human brain can comprehend up to about 5 dimensions.]

The characteristics of paper-based reporting should consciously be carried forward to electronic-based reporting or consciously left behind. Carrying paper-based characteristics into electronic-based reporting could hold back and unnecessarily constrain electronic-based reporting, not leveraging the "interactive" nature of electronic-based reporting.

6 Logical Model

The logical model details the system which is being addressed unconstrained by the XBRL syntax.

6.1 Overview of Logical Model

The following is a summary of the logical model of business reporting into which XBRLS must fit.

1. Some parties are creators of data, others are consumers of data. Parties can be both a consumer and a creator.
2. Business reports can be broken down into a collection of smaller components called "**schedules**" (sometimes referred to as tables). A report can be seen as a collection of schedules. A schedule is a collection of related information not from a presentation perspective, but from a logical use perspective. For example, a balance sheet is a schedule; its information is logically related. As it turns out, the logical relations can assist in rendering the information for human consumption and comprehension of the information. This is why, logically, a balance sheet and an income statement are two different schedules.
3. These schedules commonly consist of one or more patterns. For example a calculation or a movement-analysis, are patterns found in business reports. The sematics and structure of such constructs are expressed as **meta-patterns** within XBRLS.
4. The identified **meta-patterns** are: **hierarchy**, **calculation**, **movement**, **record**. A **schedule** is a container type meta-pattern which holds the other patterns.
5. A schedule may have one or more **axis** (aka XBRL Dimensions dimension). An axis has a domain (total, aka XBRL Dimensions domain) and members (aka XBRL Dimensions member). An entity or a portion of an entity (a segment) is an axis, even though this is articulated as a context within XBRL. A period is likewise an axis. These two axes articulated within XBRL as in the item context, are called "**quasi-dimensions**". **Units** are likewise an axis.
6. A schedule has **line items** which are contained within that.
7. **Line items** are organized within a taxonomy using **meta-patterns**.
8. **Fact values** are reported by business users in business reports. The fact values have a context. A context is expressed as a collection of axis member values. Together they define a unique "primary key" for a fact. A fact value which is numeric has a unit.
9. A meta-pattern has a **shape**. It can be though of as the layout of information expressed by the meta-pattern. A shape can for instance be expressed as the rows and columns of a table. For example, accounting policies are disclosed for one period, many calculations are shown for the current period and prior period. These are two different shapes.
10. Business reports have a **flow**. For example, flow can be the balance sheet, then the income statement, then the cash flow statement, etc. Or, the flow can be within a report such as the flow within the management discussion and analysis section of a financial report.
11. Meta-patterns have logical and pathological points at which they can be extended by a user. Logical places where a meta-pattern can be extended are called **extension points** of a meta-pattern.
12. It is a domain expert's choice as to whether a report creator can extend a taxonomy at one of the logical extension points. Whether or not a business user can use a logical

extension point within specific reporting system is articulated using **extensibility rules**.

13. Business users will never be able to agree on how data should be formatted or presented, this is why XBRL deemphasizes presentation, focusing more on articulating data. However, in order to understand provided information there must be enough organization in order to see relationships within the data. A minimal form of organization of data to allow a business user to visually see the data is called a **neutral format table**.
14. Axis can be of a **fixed** or **variable type depending on the schedule**. A fixed axis applies the same dimension member value to each fact value reported in the schedule. If a fact value set that has multiple dimension member values within the schedule, it is said to be variable.
15. A numeric fact value seen on or presented on a business report may have a **scale factor**. For example, a business report may say "in millions of dollars" and show the number "15,000,000" as simply "15" on the report. The scale factor can be seen as an axis, it must be set by the business user.
16. One business user's data set can be, and usually will likely be, different than another business user's data set. For example, different analysts care about different things. In addition to what is shown, how it is shown may be variable; for example consider an Excel pivot table, underlining, font, bold, etc.
17. A business report may have **comments** relating to fact values within that report.

The following sections describe and discuss these individual pieces which make up the logical model for business reporting into which XBRLS is trying to fit.

6.2 Schedules (aka tables, hypercubes)

Business reports can be broken down into components or individual schedules.

6.3 Flow

Business reports many times have a flow. A complicated example of flow is the management discussion and analysis section of a business report. Since at times this flow (order of schedules) is important, it can be preserved in XBRLS.

6.4 Shape

Schedules meta-patterns may have different shapes. For example a common balance sheet has two columns (current period and prior period), segment information has lots of columns. Shape is sometimes driven by data, other times driven by presentation.

6.5 Meta-patterns

A meta-pattern is a logical organization with certain implied semantics. A meta-pattern is a design pattern. The structure of each part of a business report is not unique, but rather follows a design pattern, which offers a standard solution. Once the pattern is recognized for a problem category, we know the approach to come up with a design solution. This concept is now a mainstay of the entire IT industry. The book Design Patterns by "The Gang of Four" was the book that introduced design patterns to the information technology community.

The specifics of meta-patterns will be discussed later, for now only realize that business reporting is not 100% arbitrary; business reporting patterns exist. Reporting patterns are identified and meta-patterns are created that cover all the business reporting requirements stated earlier in this document.

Seven years of work with creating XBRL-based business reporting solutions has uncovered a surprisingly small number of meta-patterns that cover all business reporting needs: hierarchy, calculation, movement, record and the "super" meta-pattern schedule.

Should the situation arise that a business reporting requirement is discovered that cannot be expressed in one of the existing meta-patterns it can simply organize it as a hierarchy until a new XBRLS standard meta-pattern is added. However, over the years many people have though that they have uncovered a new meta-pattern to finds out that they modeled their data incorrectly.

6.6 Line Items

Line items are for facts to be reported. In multidimensional analysis these are primary items.

6.7 Axis

An axis is a characteristic that a concept (a line item or primary item) possesses. For example, a fact value of "1000" is reported. That fact value is for the concept "Cash and Cash Equivalents". That fact value has the characteristic that it relates to the consolidated group which is part of the "Business Segment" axis.

6.8 Neutral Format Tables

A neutral format table is one way of presenting information. It is not the only way, but it is one way which can be logically consumed by a business user. It is the simplest acceptable (minimally acceptable) in which information can be consumed by a human.

Neutral format tables are two dimensional. They have columns and rows. Axes are either expressed as fixed or variable depending on how they relate to the fact values for a specific table. For example, "entity" relates to every fact value, therefore it is fixed. The period of concepts in a movement analysis are different for different fact values, therefore the period axis is variable.

There is a correlation between whether an axis is fixed or variable and user data input and data consumption. The user MUST be able to enter the value of an axis or read the value of an axis in order to explicitly understand the data. THAT is what drives whether an axis if fixed or variable for an axis for a specific schedule.

Neutral format tables are a language to express XBRL information without the angle brackets and with a minimum need for technical understanding of XBRL. It is important to understand neutral format tables in order to understand the meta-patterns shown below.

6.9 Scale Factor

The scale factor relates to both creation of data and consumption of data. It may be fixed or variable. It is seen as an axis as it has the characteristics of an axis. The best example of why a scale factor axis is necessary is the movement analysis. Some concepts make a change semantically "upwards"; others make the change "semantically downwards" while both are reported with a positive number. The user has to understand, visually, which is the case.

6.10 Extension Points

An extension point is a logical point at which a meta-pattern can be extended. It is not logical for a meta-pattern to be extended "anywhere". This is part of the problem business users have creating or extending XBRL taxonomies. Current tools are "free form" allowing literally anything, rather than supporting the user to execute semantically correct actions.

6.11 Extensibility Rules

Extensibility rules are different than extension points, but they make use of extension points. While it may be logical to extend a taxonomy at a give point given the meta pattern, the creators of the taxonomy may choose not to let the user extend the taxonomy at that point. For example, consider the following relation:

- Assets
    - Current Assets
    - Noncurrent Assets

Now, it may be logical from an XBRLS meta pattern perspective to add "Some other Assets" to the list of assets within a taxonomy, however within the domain of financial reporting there is no such category and therefore the business user should not be allowed to extend that specific taxonomy component.

So extensibility rules constrain extension points.

7 Physical Model

The physical model is an instantiation of the logical model within the XBRL syntax.

Where XBRL cannot provide functionality required meta data is expressed using a style which is consistent with the spirit of XBRL. For example, additional meta data required to express extension points and extensibility rules are expressed within a linkbase as other XBRL meta data is expressed.

7.1 Overview of Physical Model

XBRLS consists of a schema which is a valid XBRL taxonomy, a set of design rules, and tests which insure the design rules were implemented correctly.

The following components of XBRL are utilized:

- XBRL 2.1 specification
- XBRL Dimensions 1.0
- FRTA
- FRIS (public working draft)
- XBRL Formulas
- XBRL Generic Linkbase

7.2 Physical Components

The XBRLS Schema is located at the following URL:

http://xbrl.squarespace.com/storage/xbrls/2008/xbrls-2008.xsd

7.2.1 Substitution Groups

XBRLS defines the following substitutionGroups:

SubstitutionGroup	Alternate Marker	Explanation
xbrls:abstractGroup	[Abstract]	Used for all abstract concepts which do not fall into another category.
xbrls:DomainGroup	[Domain]	
xbrls:domainMemberGroup	[Domain Member]	
xbrls:lineItemGroup	[Line Items]	
xbrls:hierarchyGroup	[Hierarchy]	
xbrls:calculationGroup	[Calculation]	
xbrls:movementGroup	[Movement]	
xbrls:recordGroup	[Record]	
xbrls:commentGroup	[Comment]	

XBRLS re-uses the following substitution groups from the XBRL Specification:

xbrldt:hypercubeItem	[Schedule]	
xbrls:axisGroup	[Axis]	

A **marker** is an optional approach to indicating that a concept is in some concept group typically defined using the substitutionGroup.

For example, a [Hierarchy] may be defined by (a) defining the concept with a substitutionGroup of xbrls:hierarchyGroup or (b) using the marker [Hierarchy] on the standard label of the concept, or (c) both. This is useful when a taxonomy has been created by someone else but you desire to make the taxonomy XBRLS compliant. This can be achieved if other conditions are met you define a set of new label roles with these markers and then define new presentation linkbases which comply with the XBRLS meta patterns.

7.2.2 Types

XBRLS defines the following types:

Type	Explanation
xbrls:textItemType	The text item type defines a better type to use for strings than "xbrli:stringItemType". This string type may contain leading spaces, trailing spaces, double spaces, etc. The textItemType cannot. The textItemType is derived from the "xbrli:tokenItemType" which disallows leading, trailing, and double spaces.
xbrls:textBlockItemType	The textBlockItemType is also a better string, but it allows page breaks etc.
xbrls:percentItemType	
xbrls:perShareItemType	

The purpose for defining these types is to have consistent data types used, rather than each user create their own data type in each taxonomy.

7.2.3 Roles

XBRLS defines the following roles:

Type	Explanation
http://xbrls.org/2008/xbrls/role/originallyStatedLabel	To indicate that the originally stated balance is asked for.
http://xbrls.org/2008/xbrls/role/restatedLabel	To indicate that a balance has been restated.

7.3 Neutral Format Tables

Neutral format tables are a language to express XBRL information without the angle brackets and with a minimum need for technical understanding of XBRL. It is important to understand neutral format tables in order to understand the meta patterns shown below.

Neutral format tables are not part of the physical model.

8 Meta Patterns

Meta patterns can be thought of as templates into which the components of a taxonomy must fit. The advantages of meta patterns are:

1. Consistency within the taxonomy therefore easier to understand taxonomies
2. Software can operate on a taxonomy at the meta pattern level rather than the XBRL level. This makes creating a taxonomy much less complicated.
3. The users who must extend the taxonomy can understand where and how to extended that taxonomy easier.

We will explain the meta patterns that make up XBRLS and then provide a narrative description of each meta pattern. Note that the formal specification of these meta patterns is located in another document.

8.1 Overview of Meta Patterns

XBRLS is comprised of four basic meta patterns and one meta pattern into which each of the other meta patterns fits, the "Schedule". New meta-patterns can be added, but they are added in a controlled way, using a controlled process.

The four basic meta patterns are briefly described in the next sections. For a more thorough explanation of these meta-patterns, please see the XBRLS Meta Patterns Guide. This document focuses on briefly describing the patterns and specifying tests to validate the patterns, not providing a through, detailed users guide or training guide for each meta pattern.

- Schedule
- Hierarchy
- Calculation
- Movement
- Record

8.2 Schedule Meta Pattern

A **Schedule** is the shell into which the other patterns fit. Every component of an XBRLS based taxonomy is contained within a Schedule. The Schedule is used to structure the pattern. A Schedule is equivalent to an XBRL Dimensions hypercube and represented within an XBRL Extended link. This means that every concept which would ever be used within an XBRL Instance document will always be a participant of at least one hypercube or Schedule.

It is worth explaining why every concept must participate within a Schedule (hypercube). XBRL has two approaches to expressing contextual information within an XBRL instance document: using XML Schema based contextual information and using XBRL Dimensions. XBRL Dimensions information is expressed within an XBRL taxonomy.

Mixing XBRL Dimensions and XML Schema based approaches to articulating this contextual information is simply asking for problems. For this reason mixing them is explicitly forbidden by the US GAAP Taxonomy architecture and also in the XBRLS.

By now you are probably asking what this has to do with why each concept has to participate within an XBRL Dimensions hypercube. Well, the reason is that is just as bad to have some concepts participate in no dimensions and some participate in XBRL Dimensions. For example, consider a context which has no XBRL Dimensions information but rather is a default dimension. [CSH: This needs to be better explained, but this issue alone can be used to argue the case as to why everything needs to participate in an XBRL Dimension.]

So this background is to help the reader understand why every concept which would ever be reported must always participate within a Schedule (or hypercube). To explicitly state the dimensions of the concept which are being reported.

Within a Schedule you will find a Hierarchy, a Calculation or a Movement. You would never find another Schedule. This is because XBRL Dimensions does not allow nested hypercubes. Therefore a Schedule within a Schedule cannot ever exist.

Another thing to keep in mind as one looks at these meta patterns is the notion of "shape". What is meant by shape is that data has a shape. Trying to put two different shapes together is not only illogical to a human consumer of the information, it is illogical generally and can make it challenging for a computer application to render data as a human would want to see it. Basically, packing together multiple shapes of data is a bad idea.

8.3 Hierarchy Meta Pattern

A Schedule can contain a **Hierarchy**. A Hierarchy is similar to the sections in a document or indentation in a document. There are no other relations other than the relation between the child concept and its parent. A Hierarchy may contain other Hierarchies, Calculations, and Movement. Again, keep in mind the notion of shape as to when it would make sense for these multiple types of meta-patterns should be combined. [CSH: It may not be logical to allow this.]

[CSH: Note that these relations are implicit. It would not be too hard to make these explicit using different roles in the presentation linkbase.]

8.4 Calculation Meta Pattern

Next, a **Calculation** is a specific type of hierarchy which involves a series of numbers all of which are the same type and a total. The calculation is articulated within the XBRL calculation linkbase. A calculation can contain other calculations, thus nesting the calculations creating what amounts to a subtotal/grand total type of relation.

8.5 Movement Meta Pattern

Next there is the **Movement**. A Movement (or sometimes referred to as a roll forward) is a type of commutation. It is characterized as some concept which has a beginning and ending balance and the reconciling changes between that beginning and ending balance. For example, "Land, Beginning Balance + Land, Changes = Land, Ending Balance". This is not a Calculation because the beginning and ending balance are as of a point in time, whereas the changes are for a period of time. A normal Calculation has all "as of" type concepts or all "for the period" type concepts, but the two are never mixed together.

A Movement always contains a Calculation. [CSH: Actually, that may not be true. I guess it COULD possibly only be the total changes.] For example, "Additions less Disposals plus or minus any Translation Difference equals the Land, Changes" total. That calculation COULD have another nested calculation, for example a breakdown of additions for "Land, Additions Relating to Purchases" and "Land, Additions Related to Acquisitions". Each of those Calculation patterns follows the basic Calculation pattern.

8.6 Record Meta Pattern

Finally we have a **Record**. A record is something which (a) must be bound together to be meaningful and/or (b) it repeats. For example, a "Director name" is useful information and a company may have more than one director. If you disclose a "Director Salary", that information is meaningless unless you know to which director the salary relates. A record binds this information together forming a complex data type. (Note that a tuple is how XBRL does this binding.)

Surprisingly perhaps, but that is it. For example, the entire US GAAP Taxonomy which is comprised of over 14,000 concepts and some 20,000 relations can be boiled down to one or more combinations of these types of patterns. [CSH: It is not because the taxonomy is constructed inconsistently, but it could be.]

So that is the summary. Don't worry of you don't get every aspect of these meta patterns. We will now take a closer look at each of these meta patterns next. Also, keep in mind that all these rules will be deeply embedded within software which will not allow you to create a pattern incorrectly.

8.7 Defining New Meta Patterns

New meta-patterns will be defined as needed. If data needs to be expressed and no meta pattern exists to express that pattern, the following alternatives exist:

1. Define the data as a hierarchy meta-pattern. A hierarchy is a simply pattern which can basically express anything. Later, a new meta-pattern can be added to XBRLS.
2. Define the relations outside of the DTS. Loosely couple the linkbase to your processes after XBRLS validation.

Again, keep in mind that XBRLS is intended to provide the maximum amount of value and not to limit capabilities. The more you can stay within the XBRLS bounds, the more that XBRLS features can be used. But, when you have to leave the bounds of XBRLS you can in many ways and still use the additional capabilities of general XBRL. There are some cases where you cannot do this, for example adding tuples to your schema as that would impact your instance document.

9 XBRLS Rules

The following is the rules which must be followed to comply with the XBRLS Architecture Specification.

9.1 Overall Rules (normative)

The following is a summary of XBRLS compliant taxonomy design rules.

9.1.1 XBRLS taxonomies and XBRLS instances MUST be XBRL compliant.

The following is a summary of the versions of the XBRL specifications to which XBRLS complies in this version:

- XBRL 2.1 Specification Recommendation dated 2003-12-31, plus corrected errata dated 2006-12-18
- LRR 1.0
- XBRL Dimensions 1.0
- FRTA 1.0
- FRIS (public working draft)
- XBRL Formulas 1.0

9.1.2 Elements within an XBRLS taxonomy MUST NOT be of or derived from the substitutionGroup xbrli:tuple.

Tuples are not allowed.

9.1.3 Elements within an XBRLS taxonomy MUST NOT be of the substitutionGroup="xbrldt:dimensionItem" and have an attribute of xbrldt:typedDomainRef.

Typed dimensions are not allowed.

9.1.4 The <scenario> context element MUST NOT be used.

All hypercubes use the context element, rather than forcing the user to pick between the two options. There is no semantic difference between the two options.

9.1.5 The precision attribute MUST NOT be used on fact values.

Precision and decimals are two ways of doing the same thing. One can automatically convert a decimal to precision; but one cannot convert a precision to decimal. Rather than forcing users to decide and rather than causing comparison issues and XBRL Formula issues, precision will simply not be used. This reduces the number of decisions a business user must make and improves comparability.

9.1.6 The decimals attribute value of all fact values MUST be INF.

Explicitly stating the value for a fact is achieved by putting that value within the instance and setting the decimals attribute value to INF. A counter example will help explain the issue. If a fact value of say "47.32" is reported and the decimals attribute value is set to "2", that means

that the value can be assumed to be anywhere between 47.315 and 47.325. This is generally not the desired result. If the same number is reported and the decimals is reported as INF, the an XBRL processor will interpret the value as exactly 47.32.

9.1.7 The weight attribute within calculation linkbases MUST be of the value “1” or “-1”.

There is no real reason to use either a weight of 1 which means “add” or weight of “-1” which means subtract.

9.1.8 Every schema MUST contain an annotation documentation element which has a value explaining the contents of the schema.

This provides a means for a user to understand what is contained in each file and for taxonomy creators to provide that information. This eliminates the need for a user to use a file name to identify a schema.

9.1.9 Every linkbase MUST contain a documentation element.

This provides a means for a user to understand what is contained in each file and for taxonomy creators to provide that information. This eliminates the need for a user to use a file name to identify a linkbase.

9.1.10 Every element within an XBRLS taxonomy MUST participate within a hypercube whose xbrldt:contextElement="segment" and xbrldt:closed="true".

Mixing fact values in an instant which participate in XBRL Dimensions with those which do not participate in XBRL Dimensions is a dangerous practice and reduces comparability. It also further forces the user of the information to imply what is meant by having an empty context. Rather, being explicit is better.

9.1.11 The arcrole within a definition linkbase MUST NOT have the value of “notAll”.

The notAll is not used in multidimensional analysis applications.

9.2 Rules for Types

The following are specific rules for all type attribute values of concepts.

9.2.1 Concrete concepts MUST NOT use the xbrli:stringItemType.

The string type is not allowed, rather the xbrls:textItemType or xbrls:textBlockItemType should be used to be more explicit. Note that this applies only to concrete concepts which will be used within an instance. This does not apply to abstract concepts.

9.3 Rules for Extended Link Role Definitions

The following are rules specific for extended link role definitions.

9.3.1 All extended link role definitions MUST have a "usedOn" attribute value which enables use on presentation, calculation, and definition links.

The safe thing to do is to allow extended link roles to be used on all linkbases. Not defining that it can be used means that a user who finds a use for it on another linkbase which is not in the list of usedOn means that the user must define a new extended link role.

[CSH: What about formulas and generic links.]

9.3.2 All extended link roles MUST have a definition.

This is an ease of use feature for users, makes it so they don't have to view the URI.

9.4 Rules for Concepts

The following are rules specific for concepts.

9.4.1 Concrete concepts MUST NOT use the type of xbrli:stringItemType.

The string type is not allowed, rather the xbrls:textItemType or xbrls:textBlockItemType should be used to be more explicit. Note that this applies only to concrete concepts which will be used within an instance. This does not apply to abstract concepts.

9.5 Rules for Labels

The following are specific rules for labels.

9.5.1 All label resources MUST have an xml:id attribute which allows for prohibition of the label.

This enables labels to be prohibited if needed.

9.5.2 A label MUST NOT have leading spaces, trailing spaces, or double spaces.

Leading, trailing, or double spaces cause unnecessary issues.

9.6 Rules for Documentation

The following are rules specific to documentation. Documentation is a specific type of label role used to provide a definition for concepts.

9.6.1 Each non-abstract taxonomy concept MUST have documentation if a reference is not provided.

Every concept must be documented in some manner. If a reference is not provided, then documentation must be provided.

9.6.2 A documentation label MUST NOT have leading spaces or trailing spaces.

Leading and trailing spaces cause unnecessary issues. It is the case that double spaces could be used within documentation. For example, between the period at the end of a sentence and the beginning of a new sentence.

9.7 Rules for References

The following are rules specific to references.

9.7.1 All reference resources MUST have an xml:id attribute which allows for prohibition of the reference.

This enables labels to be prohibited if needed.

9.7.2 New reference parts MUST NOT be defined.

Every conceivable reference part exists with the parts defined by FRTA. It is more complicated for applications to provide any reference part or define new reference parts. The marginal cost vastly exceeds the benefit.

9.8 Rules for Presentation

The following are rules specific to any presentation.

9.8.1 Each extended link MUST have exactly only one root concept which MUST be a [Schedule] or an [Abstract] concept.

There is no rule for ordering of root concepts in XBRL. It is better to be explicit and have one root concept per extended link, rather than multiple root concepts.

9.9 Rules for Calculations

The following are rules specific to calculations.

9.9.1 No summation-item should ever have a source or target that is to a concept which is abstract.

Abstract concepts are not allowed to be used within the calculation linkbase.

9.10 Rules for [Abstract] Concepts

The following are rules specific to concepts.

9.10.1 All concepts in the substitutionGroup xbrls:abstractGroup are deemed to be [Abstract].

The substitutionGroup of xbrls:abstractGroup is used to define that a concept is of this class and must have the characteristics of the class.

Note that [Abstract] concepts are never used within a meta-pattern; but rather they are used above the [Schedule] meta-pattern to allow for the organization of [Schedule]s. Within the meta-patterns, the specific purpose of the abstract concept is indicated, rather than overloading the meaning of [Abstract] to have more than one use.

9.10.2 All concepts with the alternative marker of [Abstract] MUST comply with all rules for the xbrls:abstractGroup.

Alternatively, a concept may use the marker [Abstract] to indicate that a concept participates within the xbrls:abstractGroup. The marker and the substitutionGroup must always be consistent.

The substitutionGroup can be used without the marker; the marker can be used without the substitutionGroup; or both the marker and the specific substitutionGroup could be used.

9.10.3 All [Abstract] concepts MUST specify the abstract attribute with a value of true.

Concepts which are [Abstract] must have an abstract attribute value of true.

9.10.4 An [Abstract] MUST have a periodType attribute value set to "duration".

The periodType of an abstract concept is irrelevant as abstract concepts will never appear in an instance document. However, all XBRL concepts must have a periodtype. Rather than requiring a user to decide and expend effort to do something which is basically meaningless, all [Abstract] concepts will consistently have a periodType value of duration.

9.10.5 An [Abstract] MUST have a type of xbrli:stringItemType.

The type of an abstract concept is irrelevant as abstract concepts will never appear in an instance document. However, all XBRL concepts must have a type. Rather than requiring a user to decide and expend effort to do something which is basically meaningless, all [Abstract] concepts will consistently have a type value of xbrli:stringItemType.

9.10.6 An [Abstract] MUST NOT use a preferred label role.

There is no reason to have a preferred label role on an [Abstract] concept.

9.11 Rules for [Schedule]s

The following are rules specific to the use of the [Schedule] meta-pattern.

9.11.1 All concepts in the substitutionGroup xbrldt:hypercubeItem must be a [Schedule].

9.11.2 All concepts with the alternative marker of [Schedule] MUST comply with all rules of the xbrls:scheduleGroup.

9.11.3 A [Schedule] MUST have one or more [Axis] concepts followed by exactly one [Line Item] concept.

9.12 Rules for [Axis]

The following are rules specific to the use of the [Axis].

9.12.1 All concepts in the substitutionGroup xbrldt:dimensionItem are [Axis].

9.12.2 All concepts with the alternative marker of [Axis] MUST comply with all rules of the xbrldt:dimensionItem.

9.12.3 An [Axis] MUST have exactly one child which is a [Domain].

9.12.4 An [Axis] MUST NOT appear other than as a child of a [Schedule].

9.13 Rules for [Domain]s

The following are rules specific to the use of the [Domain]

9.13.1 All concepts in the substitutionGroup xbrls:domainGroup are deemed to be a [Domain].

9.13.2 All concepts in with the [Domain] marker MUST comply with all rules of the xbrls:domainGroup.

9.13.3 All children of a [Domain] MUST be a [Domain Member].

9.13.4 A [Domain] MUST be a child of an [Axis].

9.14 Rules for [Domain Member]s

The following are rules specific to the use of the [DomainMember].

9.14.1 All concepts in the substitutionGroup xbrls:domainMemberGroup are deemed to be a [Domain Member].

[CSH: This conflicts with the domain rule, need to fix. Seems to me that we need to separate the current substitution group of "domainMemberGroup" into "domainGroup" and "memberGroup".]

9.14.2 A [Domain Member] MUST be a child of a [Domain] or a [Domain Member].

9.15 Rules for [Line Item]s

The following are rules specific to the [Line Items] marker.

9.15.1 All concepts in the substitutionGroup xbrls:lineItemsGroup are deemed to be [Line Items].

9.15.2 All concepts with the [Line Items] marker MUST comply with all rules of the xbrls:lineItemsGroup.

9.15.3 A [Line Items] concept MUST specify an abstract value of true.

9.15.4 A [Line Items] MUST have one or more children which MUST be one of: [Hierarchy], [Calculation], [Movement], or [Record].

9.15.5 A [Line Item] MUST only appear as the child of a [Schedule].

9.16 Rules for [Hierarchy] Meta-patterns

The following are rules specific to the [Hierarchy] meta-pattern.

9.16.1 All concepts in the substitutionGroup xbrls:hierarchyGroup are deemed to be [Hierarchy].

9.16.2 All concepts with the [Hierarchy] marker MUST comply with all rules of the xbrls:hierarchyGroup.

9.16.3 A [Hierarchy] concept MUST specify an abstract property value of true.

9.16.4 A [Hierarchy] MUST contain either a concept, [Hierarchy], [Movement], [Calculation], or [Record].

9.16.5 A [Hierarchy] MUST appear as the child of another [Hierarchy] or as the child of a [Schedule].

9.17 Rules for [Calculation] Meta-patterns

The following are rules specific to the [Calculation] meta-pattern.

9.17.1 All concepts in the substitutionGroup xbrls:calculationGroup are deemed to be [Calculation].

9.17.2 All concepts with the [Calculation] marker MUST comply with all rules of the xbrls:calculationGroup.

9.17.3 A [Calculation] concept MUST specify an abstract value of true.

9.17.4 A [Calculation] MUST contain only concrete numeric concepts of the same type or another [Calculation].

9.17.5 A [Calculation] MUST contain a last child which is concrete.

9.17.6 A [Calculation] MUST have summation-item arcs which define the calculation within the calculation linkbase of the same extended link role name as the presentation linkbase.

9.17.7 A [Calculation] MUST have as its last child concept the summation item of the set of summation-item arcs.

9.18 Rules for [Movement] Meta-patterns

The following are rules specific to the [Movement] meta-pattern.

9.18.1 All concepts in the substitutionGroup xbrls:movementGroup are deemed to be [Movement].

9.18.2 All concepts with the [Movement] marker MUST comply with all rules of the xbrls:movementGroup.

9.18.3 A [Movement] concept MUST specify an abstract value of true.

9.18.4 A [Movement] MUST have exactly four children: a beginning balance, the period increase (decrease), the ending balance, and a Boolean concept used to hold the XBRL Formula for the movements reconciliation business rule.

9.18.5 The concrete concepts used for or within a [Movement] for the beginning balance, ending balance, and changes MUST be of the same type.

9.18.6 The first child of a [Movement] MUST be of periodType value of instant.

9.18.7 The first child of a [Movement] MUST have a preferred label role of "beginning balance".

9.18.8 The second child of a [Movement] MUST be a [Calculation].

9.18.9 The type of all concepts participating within the [Calculation] within a [Movement] MUST be of the same type.

9.18.10 The third child of a [Movement] MUST have periodType value of instant.

9.18.11 The third child of a [Movement] MUST have a preferred label role which is different than the first concept.

9.18.12 The fourth child of a [Movement] MUST be concrete and MUST be derived from the type xbrli:booleanItemType.

[CSH: I think we may be able to drop this. I think there can always be an explicit formula for a movements, and that we don't need to necessarily capture the value as a Boolean value. That is just how I have done it in the past. Need to discuss.]

9.18.13 The fourth child of a [Movement] MUST be of periodType value duration.

9.19 Rules for [Record] Meta-patterns

The following are rules specific to the [Record] meta-pattern.

9.19.1 All concepts in the substitutionGroup xbrls:recordGroup are deemed to be [Record].

9.19.2 All concepts with the [Record] marker MUST comply with all rules of the xbrls:recordGroup.

9.19.3 A [Record] concept MUST specify an abstract value of true.

9.19.4 A [Record] MUST appear as the child of a [Line Item].

[RVE: I need an example for this]

9.20 Rules for [Comment]

The following are rules specific to the [Comment] substitutionGroup. The [Comment] is not a meta-pattern and it generally will never be a component of a released and final taxonomy. It is simply a mechanism to place useful information useful to a reader of a taxonomy within the taxonomy without breaking any of the meta-patterns. This information can be used to help review a taxonomy which is under construction or to highlight information in a final version.

A [Comment] is basically an abstract concept (so no value can exist in the instance document) which can be ignored and not considered when creating or analyzing any meta-pattern.

9.20.1 All concepts in the substitutionGroup xbrls:commentGroup are deemed to be [Comment].

9.20.2 All concepts with the [Comment] marker MUST comply with all rules of the xbrls:commentGroup.

9.20.3 All [Comment] concepts MUST specify an abstract value of true.

9.20.4 All [Comment] concepts MUST be ignored within and by other rules.

9.21 Rules for Other Purposes

The following are other rules.

9.21.1 The preferred label roles of ‘beginning balance’ and ‘ending balance’ MUST NOT be used outside a [Movement].

There is no reason to use a preferred label role of beginning balance or ending balance outside a movement analysis.

9.21.2 The preferred label role of ‘total’ MUST only be used within a calculation and only on the last child of a calculation.

There is no reason to use a total label role outside a calculation.

10 Extension Points and Extensibility Rules

The following is a discussion of extension points and extensibility rules.

10.1 Extension Points

There are no explicit rules for extension points as extension points are inherent within each meta-pattern. Basically, an extension to a meta-pattern must follow the rules of the meta-pattern being extended. No further rules are necessary.

However, it is helpful to briefly discuss the logical extension points which are allowed and the pathological extension points which are not allowed. This will be done for each meta-pattern.

10.1.1 Hierarchy

A [Hierarchy] can be extended anywhere. You must comply with the rules of what is and what is not allowed within a [Hierarchy].

10.1.2 Calculation

There are two points at which it is logical to extend a [Calculation].

1. Adding additional concepts to the [Calculation]; basically as a child of the [Calculation] concept but before the total.
2. Adding additional details of an existing concept to the [Calculation] on level 1; basically creating a sub-calculation at level 2 by creating a new [Calculation], putting the existing concept from level 1 to the [Calculation] at level 2 as that calculation's total, and adding additional concepts which add up to that total.

It is pathological to:

A. Add a second total concept.

B. Add concepts under the total.

10.1.3 Movement

There is one points at which it is logical to extend a [Calculation].

1. Extending the increases (decreases) calculation; which is basically exactly like extending a [Calculation] which is described above. (Additional detail for existing increases (decreases) set of concepts are [Calculation] extensions.)

It is pathological to:

A. Add a second set of increases (decreases).

B. Add a second instant concept (beginning and ending balance).

C. Adding a second reconciliation concept.

10.1.4 Record

A [Record] is similar to a [Hierarchy] in that it is rather flexible in terms of where it can be extended.

10.1.5 Schedule

There following are the logical to extend a [Schedule].

1. Add a new [Axis].
2. Add a new [Domain Member] to an existing [Axis] [Domain].
3. Add a new meta-pattern as a child of the [Line Items].

It is pathological to:

A. Add a second set of [Line Items].

10.2 Extensibility Rules

The following explains how extensibility rules work. Basically, if no information is communicated to the contrary, every logical extension point is allowed to be extended.

If taxonomy designers desire to turn off one of those extension points they would create a definition linkbase which parallels the presentation linkbase (and using the same extended link role) with one of two arcroles being used in place of the of the "parent-child" arcroles of the presentation linkbase:

http://xbrls.org/2008/xbrls/arcrole/extensionAllowed-true

http://xbrls.org/2008/xbrls/arcrole/extensionAllowed-false

[CSH: I am just making this up as I go along here, but it seems that this would work and is quite simple.]

11 XBRLS Best Practices

The following are suggestions on how to best implement certain features of XBRL within an XBRLS taxonomy or instance and other practices which help create higher quality taxonomies. These rules should be implemented as warnings.

11.1 Overall Taxonomy Design Best Practices

The following is a summary of overall best practices.

11.1.1 XBRLS taxonomies SHOULD comply with a style guide.

Consistent use of one way of spelling words is a good practice. For example, using "Long Term Debt" and "Long-Term Debt" and "Long-Term-Debt" interchangeably within a taxonomy is a bad practice.

11.1.2 All information other than documentation that has no processing semantics SHOULD be removed from schemas, linkbases, and instances.

11.1.3 Unnecessary import and schema references that force users to become aware of unnecessary information SHOULD be avoided.

11.2 Extended Link Role Best Practices

The following is a summary of best practices which relate to extended link roles.

11.2.1 All extended link role definitions SHOULD follow a consistent pattern.

11.2.2 All extended link role definitions SHOULD exist in one or more separate files, rather than be included within the base schema of concepts.

It is better to not force a user to make use of a schema in order to use the extended link role definitions. It is a better practice to separate extended link role definitions and concept definitions into separate physical files.

11.2.3 Extended link role definitions SHOULD be given values that facilitate proper ordering by software applications.

XBRL provides no mechanism specifying the order of extended links inside the taxonomy. The best practice used by many taxonomies is to use an alpha-numeric scheme to enable software to order extended links within the application.

11.2.4 Each extended link role SHOULD be used at least once within the presentation linkbase.

There is no point in defining an extended link role and then not making use of that role somewhere within the taxonomy.

11.3 Labeling Best Practices

The following is a summary of best practices which relate to creating labels.

11.3.1 The label of a [Line Items] SHOULD be consistent with the [Schedule] in which it is contained.

Consistency is a good practice. This make it easier and less confusing for users of the taxonomy.

11.3.2 The label of a beginning balance SHOULD be consistent with the standard label of that concept appending ", Beginning Balance" to the end.

Consistency is a good practice. This make it easier and less confusing for users of the taxonomy.

11.3.3 The label of an ending balance SHOULD be consistent with the standard label of that concept appending ", Ending Balance" to the end.

Consistency is a good practice. This make it easier and less confusing for users of the taxonomy.

11.3.4 The label of a [Calculation] SHOULD be consistent with the label of the concept which is the total of the [Calculation].

Consistency is a good practice. This make it easier and less confusing for users of the taxonomy.

11.3.5 A [Calculation] MAY use the 'total' preferred label role on the last child of a [Calculation]; if this is done, it should be done consistently throughout the taxonomy.

Consistency is a good practice. This make it easier and less confusing for users of the taxonomy.

11.3.6 The label of a total SHOULD be consistent with the standard label, appending ", Total" to the end.

Consistency is a good practice. This make it easier and less confusing for users of the taxonomy.

A Errors and Warnings

The following is a summary of errors and warnings.

Validating Taxonomies

The namespace `xbrlse` is defined as `http://xbrls.org/2008/xbrls/taxonoomy/errors`

Taxonomy Error	Meaning	Ref.

Taxonomy Error	Meaning	Ref.
[XBRLS Error, 1] xbrlste:PresentationNotSchedule	This Extended Link does not look like a [Schedule].	X.X

The namespace xbrlse is defined as http://xbrls.org/2008/xbrls/taxonoomy/warning

Taxonomy Warning	Meaning	Ref.
[XBRLS Warning, 1] xbrlstw:GeneralUnusedRole	Extraneous Role defined in taxonomy, but not used anywhere.	X.X

Validating Instances

The namespace xbrldie is defined as http://xbrls.org/2008/xbrls/instance/errors

Instance Error	Meaning	Ref.
[XBRLS Error, 1] xbrlsie:PresentationNotSchedule	This Extended Link does not look like a [Schedule].	X.XX

The namespace xbrldie is defined as http://xbrls.org/2008/xbrls/instance/warning

Instance Warning	Meaning	Ref.
[XBRLS Error, 1] xbrlsiw:PresentationNotSchedule	This Extended Link does not look like a [Schedule].	X.XX

B XBRLS Terminology Glossary

The following is a convenient summary of XBRLS terminology. The intent is to allow for hiding technical terminology from business users where possible.

XBRLS Term	Meaning
Concept	The XML Schema term "element" is used within an XBRL taxonomy to define XBRL concepts. The term "concept" should be used, rather than the commonly used terms "element". A concept is a business concept. That concept is expressed within XBRL as an XML Schema element.
Fact Value	A fact value in an XBRL element within an instance document. That value has a concept associated with it. XBRL refers to these as items.

XBRLS Term	Meaning
Network of Relationships	A network of relationships is two or more concepts which are organized within one or more XLink Extended Links which are of the same type (presentation, calculation, definition) and have the same extended link role. How XBRL uses extended links and how XLink uses extended links are different and this difference needs to be well understood. A network of relations is after all DTS processing has taken place, it is a result of that processing. A **base set** of arcs is the preprocessed version of the network.
Schedule	XBRLS meaning for what XBRL calls a hypercube.
Domain	Total of all the members.
Member	A value of an Axis or dimension.
Axis	XBRLS meaning for what XBRL calls a dimension.
Line Items	Line items is the collection of primary items which are allowable within a hypercube.
As of (or As at)	Instant periodType. For example, a balance sheet is as of (or as at) a point in time such as December 31, 2007.
For period (or For Period Ended; For period Ending)	Duration periodType. For example, an income statement is for a period of time such as for the year ended December 31, 2007.
Record	XBRL term tuple. A complex concept comprised of one or more simple concepts which can repeat (i.e. there can be more than one). For example, a "Related Party" is a [Record] because there can be more than one related party as compared to say a "Cash and Cash Equivalents Policy" of which there may be only one (i.e., it repeats). Director Compensation is a record because a Director Salary and Director Bonuses is meaningless unless it is associated with a Director Name.
Hierarchy	A [Hierarchy] is the simplest form of a relation as it has no calculations associated with it and it does not repeat. It is similar to a simple categorization or sub categorization.
Calculation	A form of a computation where all the concepts are of the same context. A [Calculation] is a simple summation with all concepts within the same context.
Movement	Reconciliation of concept from one instant to another instant. There are two types of movements. One is where there is a beginning balance, changes, and an ending balance; the only thing changing is the period. The second is a reconciliation of a balance to another balance and something other than the period changes; for example the reconciliation of retained earnings when there is a prior period adjustment, the balances are the originally stated value and the restated value, the changes are the adjustments to the originally stated balance giving rise to the restatement.
Business Rule	XBRL Formula, generally used where you cannot make an XBRL Calculation work because the computation is complex or because the fact values participate in different XBRL contexts.
[CSH: Need terms for these three]	Parent
	Child

XBRLS Term	Meaning
	Sibling

C Comparison of Existing Taxonomies

The following is a comparison of what are considered high quality public taxonomies created using best practices. The purpose of the comparison is to glean clues as to what XBRLS should be comprised of in terms of architecture.

Characteristic	**XBRLS**	COREP	FINREP	IFRS	Netherlands	ABS	US GAAP
Uses XBRL Dimensions	**Yes**	Yes	Yes	No[1]	No	Yes	Yes
Uses tuples	**No**	No	No	Yes	Yes	Yes	No
Uses typed dimensions	**No**	Yes	Yes	No	No	???	No
Every concept participates in a hypercube	**Yes**	Yes	No	No	No	No	No
Allows XML Schema-based contextual information	**No**	No	????	Yes	????	???	No
Uses weights other than 1 and -1	**No**	No	No	No	No	???	No

1. Does not disallow use of XBRL Dimensions, but the taxonomy does not make explicit use of XBRL Dimensions.

D Document Conventions

The following formatting is used for non-normative examples in this document:

The following formatting is used for non-normative counterexamples (examples of poor, discouraged, or disallowed usage) in this document:

Non-normative editorial comments are denoted as follows and removed from final recommendations:

The use of italics is for emphasis and has *no* normative impact.

E References

[XBRL] Phillip Engel, Walter Hamscher, Geoff Shuetrim, David vun Kannon, Hugh Wallis.
Extensible Business Reporting Language (XBRL) 2.1 Recommendation with corrected errata to 2006-12-18
http://www.xbrl.org/SpecRecommendations/

[DIM] Ignacio Hernández-Ros, Hugh Wallis
XBRL Dimensions 1.0, RECOMMENDATION dated 2006-09-18
http://www.xbrl.org/SpecRecommendations

[XBRLS] Rene van Egmond and Charles Hoffman
XBRLS: How a simpler XBRL can make a better XBRL
http://xbrl.squarespace.com/xbrls

[PATTERNS] Charles Hoffman, et. al.
XBRLS Meta-Patterns Guide
http://xbrl.squarespace.com/xbrls

F Document History

Date	Editor	Summary
2007-11-03	Hoffman	First version of this document.
2008-01-26	Hoffman	Adjusted and synchronized document to prototype and database repository prototype.
2008-03-24	Hoffman	Built out rules
2008-04-17	van Egmond	First round of comments