Understanding the Moving Pieces
Impacting Dimensional Aggregation

This document is a summary of information relating to the pieces of the puzzle which appear to have an impact on dimensional aggregation within XBRL when XBRL Dimensions is used. At this point the information is more of a stream of consciousness dump than a well flowing tutorial, it will be organized better later.

Consider the following business use case:

Sample Company
For Period Ending December 31,
(thousands of dollars)

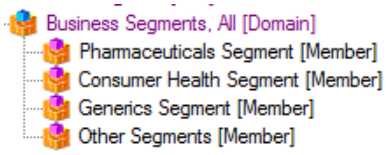
	2010	2009	2008
Sales, all Business Segments, all Geographic Areas	32,038	35,805	32,465
Breakdown by Business Segment:			
Pharmaceuticals	20,181	18,150	15,275
Generics	2,433	1,973	1,823
Consumer Health	6,675	6,514	5,752
Other Segments	2,749	9,168	9,615

The screen shot shows a breakdown of sales by business segment and a total for sales for all business segments. This is an example of dimensional aggregation. The concept "Sales" is part of a hypercube which has a dimension "Business Segments" with the domain "All Business Segments" which represents a total of the members Pharmaceuticals, Generics, Consumer Health, and Other Segments.

Said using terminology of US GAAP/SEC taxonomy: The concept "Sales" is part the [Line Items] which is part of a [Table]. The [Table] has an axis "Business Segments [Axis]" which has a domain "All Business Segments [Domain]". That [Domain] represents a total of the members Pharmaceuticals [Member], Generics [Member], Consumer Health [Member], and Other Segments [Member].

The above might be expressed in the presentation relations of an XBRL taxonomy following the US GAAP Taxonomy/SEC modeling style and look something like this:

When I refer to the business use case of dimensional aggregation, I am referring to this specific portion of the modeling:



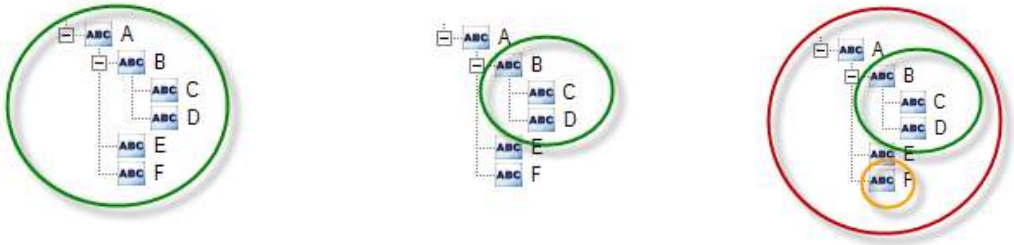
Intuitively, it is not a huge jump to make to believe that the sum of the [Member]s should add up to the [Domain]. However, there are other use cases which must be considered. Also, there are other issues.

First off, the XBRL Dimensions specification does not address dimensional aggregation at all. As you can see by looking at the specification, there is no section in the XBRL Dimensions specification (http://www.xbrl.org/Specification/XDT-REC-2006-09-18+Corrected-Errata-2009-09-07.htm) which addresses dimensional aggregation.[1]

Secondly, there are many different business use cases which one might run across which relate to what one might think of as dimensional aggregation. Let me define a couple of terms and then I will explain the use cases.

To be clear, let me define the term **dimensional aggregation** as I am using it here. A dimensional aggregation is an aggregation or summing of information across the members of a domain within a dimension. I am focusing on explicit members as defined by the XBRL Dimensions Specification, these are allowed by the US GAAP Taxonomy. I am specifically excluding typed members from this discussion as the US GAAP Taxonomy Architecture specifically prohibits the use of typed members and I am focusing on dimensional aggregation in SEC XBRL filings with explicit members at the moment.

Another term worth defining because there is generally a lot of confusion relating to it is the term domain. I am guilty of using this term incorrectly myself. Basically a **domain** is a set of members. Consider this example:



[1] If you go to the XBRL Dimensions specification and look at the Document History toward the end of the document you can see that aggregation across dimensions was considered and removed. Not the addition of and subsequent removal of the "aggregator-contributor" arcrole and the summable attribute.

Assume that the above trees are the [Member]s of an [Axis]. In the diagram, **A** is a domain with members A, B, E, F, C and D. Also, **B** is a domain with the members B, C and D. And I also believe that **F** is a domain with the only member being itself.

The US GAAP Taxonomy requires one root level member referred to as a [Domain]. That [Domain] then has [Members] which could be organized as a flat list, could contain additional hierarchy, and which have some sort of relations between the [Members].

So a domain (a set of members) can have different types of relations, similar to how the [Line Items] can have many different types of relations. In the US GAAP Taxonomy, a set of [Line Items] can express a [Hierarch], a [Roll Forward], a [Roll Up] or a few other relation patterns, which I call an information model.[2]

For example, a domain can contain a **complete** set or a **partial** set of members. A domain may **aggregate** (add up) or it may not aggregate and simply be a **hierarchy** of members.

Different taxonomies have used different terminology for these types of relations. For example, the CEBS project uses these two definitions:

- A **complete tree** is one where each parent node could be calculated as the aggregation of its children.
- An **incomplete tree** is one where each parent node can be said to be equal or greater than the aggregation of its children.

Another way to describe the relations between a domain and its members is to use **set theory**. For example, set theory defines the term patrician. (see http://en.wikipedia.org/wiki/Partition_of_a_set). A **patrician** is both collectively exhaustive and mutually exclusive, per the definition of the patrician of a set. A domain is basically a set of members.

There is no "standard" terminology at this time for these types of relations, all the terminology is taxonomy specific. There is no correct or incorrect terminology. This is because neither the XBRL Dimensions specification nor anything from XBRL International defines any standard terms. The US GAAP Taxonomy nor the SEC filing manual does not provide an information model for these sorts of relations.

However, although XBRL Dimensions does not define how members of a domain aggregate or if they aggregate at all, you can use XBRL Formulas to clearly define such aggregation if they exist. This XBRL Formulas definition both articulates the aggregation scheme and can also be used to validate XBRL instances against that scheme. XBRL Formulas can handle quite complex models.

[2] For more information on information models see http://www.xbrlsite.com/US-GAAP-2011/LogicalModel/SEC-XBRL-Primer-2011-02-15.pdf

But, since the SEC does not allow XBRL Formulas to be submitted with an SEC XBRL filing, these filings can have aggregation schemes which are inconsistent with aggregation schemes you may come up with or different than how you might interpret the XBRL taxonomy. SEC XBRL filers can still create a valid scheme of aggregation, test any XBRL instances created against it in their SEC XBRL filing but not submit that XBRL Formula set with their SEC XBRL filing. One way or another, SEC XBRL filers should prove that their XBRL instances do in fact follow their defined scheme by validating their XBRL instance.

Here are some example financial reporting use cases which show different types of dimensional aggregation schemes:

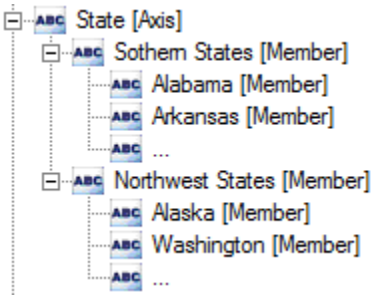
- **[Axis] with no aggregation**: Subsequent events or property, plant, and equipment policies would be an example of where you would have an [Axis] with no aggregation. Subsequent events are rarely, if ever, added up to arrive at some total amount for subsequent events of an entity. The same with PPE policies, they are not added up.

- **[Axis] with basic aggregation**: Nonmonetary transactions are a good example of a basic aggregation. There is no real grouping of nonmonetary transactions, it is simply a flat list. That list has an amount of for each nonmonetary transaction, and the total for nonmonetary transactions is commonly reported. That total does not tie to any other location in the financial report, it stands alone.

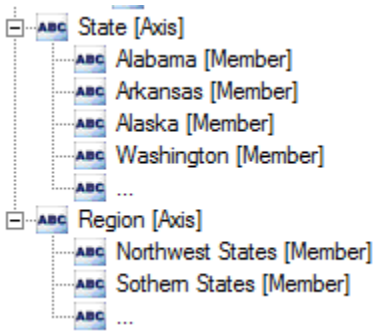
- **[Axis] with complex aggregation**: Business segment reporting is a good example of potentially complex aggregation. You may have a Parent Holding Company [Domain], which has consolidation eliminations, and multiple business segments. You may add to this a breakdown of continuing and discontinued segments. This could become even more complex with asset groups, reporting units, and disposal groups.

When you model a business use case there are at least two approaches you could choose to model members: as an [Axis], or as hierarchy within a set of [Member]s. An example will explain what I mean. Suppose you wanted to breakdown some set of information by state and region. Here are two approaches.

One approach is to, within a [Table], create one [Axis], modeling states and regions together within one [Axis]:



An alternate approach is to, within a [Table], create two [Axis], one for the state, and another for the region:



There really is no right or wrong answer here; how you would model your business use case depends on the dynamics of what it is you are modeling. The primary point I am making here is that if there are multiple ways to model the same information; then what criteria do you use to determine the most appropriate modeling approach?

In order to maintain comparability between filings created by different SEC XBRL filers, these criteria need to be established.

What I have pointed out here is similar to the question of whether it is better to model something as a concept and part of a set of [Line Items]; or to model that same thing as a [Member] of an [Axis]. This also impacts comparability.

Or, does it really impact comparability across filings? Are these alternative approaches semantically equivalent, only differing in the features realized from modeling the information using the selected modeling approach as compared to the characteristics of the alternative approach? Sure, it may be a little more complicated to do a comparison because the approaches have to be mapped together. Could this mapping be achieved in an automated manner?