21. APPENDIX: Top XBRL Technical Syntax Related Modeling Tips

The following is a summary of the top 10 XBRL taxonomy and XBRL instance creation tips which will help you create quality systems which make use of XBRL, helping a business domain achieve what they are striving to achieve.

Generally business users will never need to deal with these sorts of issues as software will hide the issues from users. However, today software does not hide these XBRL technical syntax related issues. As such, we point them out.

21.1. Define a clear, unambiguous, formally documented financial report model layer

Define a clear, unambiguous, formally documented financial report model layer. At a minimum the XBRL Abstract Model 2.0 should be followed. Alternatively, some model terminology which is consistent with that model should be clearly defined.

For more information see:

http://xbrl.squarespace.com/journal/2012/6/15/xbrl-international-releases-semantic-model-public-working-dr.html

21.2. Define a clear, unambiguous, formally documented information model

Define a clear, unambiguous, formally documented information model. One of the biggest problems XBRL taxonomies have is inconsistent information models. An information model is simply how the relations within a taxonomy are structured. This is of particular importance when extensibility is employed within your system. For example, the US GAAP Taxonomy creates structures such as [Table]s, [Roll Forward]s, and other such structures. They explain how these structures are to be created. You should do the same in order to be able to evaluate how your taxonomy is created and in order to explain how your taxonomy should be extended. Taxonomies are simply not random. Make yours clear, unambiguous, and formally document it so those extending your taxonomy can follow the rules.

21.3. Don't mix dimensional and non-dimensional models

Don't mix dimensional and non-dimensional models; personally I prefer a dimensional model. If you use XBRL Dimensions, then every concept should be attached to a hypercube thus requiring the dimensions of the concept to be explicitly identified. Mixing a dimensional model and a non-dimensional model causes headaches which can be avoided by simply using one model or the other. Since business information is inherently dimensional anyway, I personally prefer a dimensional model, using XBRL Dimensions consistently throughout your XBRL taxonomy. Mixing models also make using XBRL Formula much trickier.

21.4. Make each hypercube unique (use isomorphic hypercubes)

Make each hypercube unique. There are advantages to making each hypercube in an XBRL taxonomy unique. Take a look at this taxonomy. Search for the line items

which say "Statement [Table]". You can see what I am talking about more clearly by looking at this. What is the point of using the same hypercube for each set of dimensions and concepts? Why not use a different unique hypercube name for each hypercube? This has a number of benefits, including making the extended link as any form of semantics unnecessary. The FINREP taxonomy makes each hypercube unique.

21.5. Close all hypercubes

Be sure to require that all hypercubes be closed. All hypercubes you create which have an "all" role should be closed (and all your hypercubes which have a "notAll" role should be open if you happen to use those). Leaving a hypercube open basically lets anything exist in the context. What is the point of that? Be explicit and close all your hypercubes.

21.6. Provide dimension-defaults for each dimension

Each dimension should have a dimension-default. Much confusion exists as to what dimension-defaults actually do and software interoperability can be an issue with dimension-defaults. To achieve maximum reliability, predictability and therefore safety always provide a dimension-default.

The purpose of a dimension-default is to enable one hypercube to intersect with one or more other hypercubes.

21.7. Clearly differentiate members and concepts

Always clearly differentiate dimension values and concepts. When creating an XBRL taxonomy you don't want users of the taxonomy to mix up what is a dimension value (such as a domain or a member) and what is a concept which can be used to report a value. The US GAAP Taxonomy differentiates domains and members by appending "[Domain]" or "[Member]" to such dimension values and assigning those types of elements to a special type value of "domainItemType". You could also use the substitutionGroup to differentiate these two types of XML Schema elements. That way, users don't get confused.

21.8. Use either segment or scenario, there is no real reason to use both

Use either segment or scenario, there is no real reason to use both. Eliminating unnecessary options makes things easier. There is no semantic difference between using the segment context element and the scenario context element. Besides, if different XBRL instance creators use different elements, comparability then becomes an issue. You can avoid both of these problems by simply using one or the other. Which is as easy as tossing a coin really. Using scenario seems to be the best, but the US GAAP Taxonomy suggests segment. You can pick.

21.9. Use XBRL Dimensions or use tuples, don't use both in the same XBRL taxonomy

Tuples and XBRL Dimensions are redundant in that they are basically two syntaxes for doing what amounts to the same thing. Each has its pros and cons. Pick and use one or the other; personally I prefer XBRL Dimensions. The biggest problem with

using both tuples and XBRL Dimensions is explaining when to use one and when to use the other. The primary reason I don't like tuples is because they significantly inhibit extensibility. Basically, tuples add back the XML content model with XBRL worked to remove. XBRL Dimensions can do everything that tuples can do, but tuples are not nearly as functional as XBRL Dimensions.

21.10. Use decimals or precision, don't allow both

Precision and decimals are redundant, pick and use one or the other; personally I prefer decimals. The precision and decimals attribute on a fact value serves the same purpose. There is pretty much universal agreement that only one of these should have been created. Having both causes more work when working with XBRL instance information which contains both. FRTA suggests that decimals be used. So does the US GAAP Taxonomy. I agree and suggest using decimals because it is easier for business users to understand.

21.11. Avoid complex typed members unless you really need them

Don't use complex typed members for a dimension unless you really need them. Complex typed members allow literally any XML you can think of as a possible value, except for XBRL itself. It is way too much to ask for a software application to implement something like this. Further, using it to compare to entities effectively can be quite challenging. You can achieve the same results by using a number of simple typed members, which are much easier to build an interface for and easier to make work. Complex typed members for dimension values are far more trouble than they are worth and should be avoided.

21.12. Be explicit, consistent and concise when expressing taxonomy information

Don't be redundant in expressing taxonomy information. If you express things twice in two different ways, you create work in that you now have to make sure the two things you are expressing are in sync. For example, expressing information in a presentation linkbase and also in a definition linkbase causes such redundant information. The FINREP taxonomy figured this out and does not make a presentation linkbase available with its taxonomy. In the short term this can be a bit of a challenge to effectively do because most software applications rely on the presentation linkbase. Overtime and as software gets better, this will not be an issue. First, realize that you are creating redundant information. Second, if you can, you may want to consider not making this redundant information available in your XBRL taxonomy.

21.13. Consider ditching XBRL calculations

Give serious consideration to using XBRL Formula rather than XBRL calculations. XBRL Formula is several orders of magnitude more powerful that XBRL calculations. Also, XBRL calculations have their idiosyncrasies. More and more people are moving to XBRL Formula. You may want to give strong consideration to abandoning XBRL calculations and using XBRL Formula instead. XBRL calculations can be easier in certain situations. The trade-offs should be understood and evaluated in making your decision.

21.14. Realize that XBRL instance contexts and XBRL Dimensions hypercubes constrain facts differently

XBRL has two mechanisms for defining contextual information and those two ways work differently. The two ways are XBRL contexts and XBRL Dimensions hypercubes. Two specific pieces of an XBRL context, entity identifier and period, must exist on every XBRL Fact. They are unconstrained and not impacted by any context constrains defined by an XBRL Dimensions hypercube. Segment and scenario information not defined by XBRL Dimensions works this way also. XBRL Dimensions hypercubes is another way of constraining information, basically the dimensions or Measures associated with a Fact.