

1. Fragment Arrangement Patterns

1.1. Fragment arrangement patterns

Flow is the notion of relations between Networks and/or [Table]s for the purpose of ordering or sequencing information contained in a digital financial report. Creating schemes for generating the desired flow of information contained by a digital financial report can be impacted by metadata available. Fragment arrangement patterns¹ explain different approaches to representing the flow of a digital financial report.

1.1.1. XBRL-based financial reports to SEC report fragment ordering scheme

One example of using networks to order or sequence the contents of an XBRL-based digital financial report can be seen in how the SEC achieves sequencing. Consider the following example:

Note C. Inventory, By Component (As Reported February 12, 2011 [Member], Parent Company [Member], USD \$) In Thousands		
	Dec. 31, 2010	Dec. 31, 2009
Inventory, by Component [Roll Up]		
Inventory, Finished Goods	\$ 1,000	\$ 1,000
Inventory, Work in Process	1,000	1,000
Inventory, Raw Materials	1,000	1,000
Other Inventory, Supplies	1,000	1,000
Inventory, Net, Total	\$ 4,000	\$ 4,000

The above is a fragment of a model financial report rendered within the SEC interactive data previewer. The XBRL extension taxonomy created by the public company drives the SEC interactive data viewer.

Networks drive the rendering of the table of contents on the left. Each network can be broken into three components which drive the sequencing of the rendering framework:

- **Number** such as "101000" within the first network.
- **Sort Category** such as "Document", "Statement" or "Disclosure"
- **Title** or other part of the networks definition.

The category is used to put the different networks into one of the yellow categories in the SEC example, the number determines the order within the category, and the balance of the description is the label that a user sees.

¹ Fragment arrangement pattern examples, <http://xbrl.azurewebsites.net/DigitalFinancialReporting/FragmentArrangementPatterns/2017-05-07/>

This approach is workable, but it means that all information must be broken out by network and anything smaller than the network itself cannot be broken out any further. For example, table information is not used for rendering information at all.

You can examine this in more detail by examining the reference or model implementation of an SEC XBRL financial filing.

1.1.2. Metadata provided impacts ordering

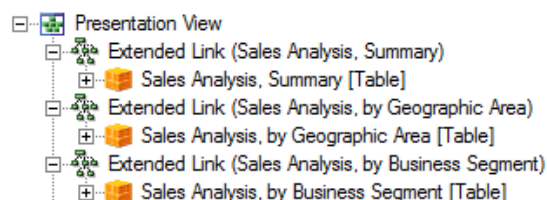
Certain metadata is required by the XBRL technical syntax. Other metadata is determined by how a taxonomy is expressed. The following is a summary of the constraints imposed by approaches used to express metadata within a taxonomy and how those constraints impact ordering.

- **Networks** – Networks are always required to be unique so as such, networks can always be used to order a taxonomy. However, if networks alone are used many times not enough granularity is achievable. Also networks cannot be articulated within a hierarchy.
- **Networks plus Non-unique Tables** – Tables can be used with networks to order information. However, depending on whether the tables are expressed are unique governs the role a network must play in allowing a table to be specifically identified.
- **Unique Tables** – If every table within a taxonomy is unique, then networks no longer need to be relied upon to uniquely identify and locate a table, the table alone will allow such identification.

1.1.3. Tables organized into a list

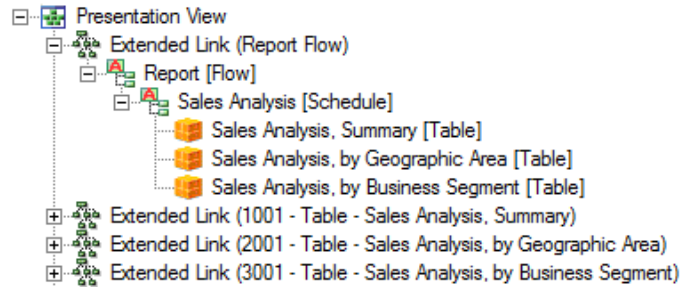
Another approach to articulating sequencing information can be seen by comparing the *Pivot Table* business use case with the *Flow* business use case.

Consider the screen shot below of the Pivot Table business use case:



There are three networks with three tables. Each network and table is unique. Suppose you wanted to articulate the ordering you would prefer for working with this information, how would you do that? You could request the information in the physical order in which it exists within the XBRL taxonomy or you could request the information in alphabetical order, that is about all the options you might have.

Now consider the *Flow* business use case below. The this taxonomy has a network called "Report Flow". Within that network, a hierarchy of the [Table]s which exist in the taxonomy for this financial report is provided.



As such, a software application can read that hierarchy and use it within the application to show the summary first, the geographic table second, and the business segment third.

Alternatively, the numbering of the network could be used to achieve the same goal as with the SEC example.

The [Table]s alone can be used, and the networks totally ignored, because each table is unique. By contrast, if each table were called “Sales Analysis, Summary [Table]”, then to identify which [Table] you were looking for, you would also need to rely on the network.

1.1.4. Notion of the “Implied [Table]”

In the section which discusses the report elements which make up a digital financial report we explain that everything within a digital financial report exists within a [Table], be that [Table] explicitly articulated using the “[Table]” report element, or the table is implied.

Basically, everything expressed within a network which is not contained within some explicit [Table] can be thought of existing within a pseudo or implicit table called “No Table [Table]”. Because you might have more than one “No Table [Table]”, you must rely on the network to uniquely identify which “No Table [Table]” you would like to work with. As such, using implicit tables requires you to work with tables just as though you created non-unique tables.

1.1.5. The “Statement [Table]”

Another approach to defining [Table]s can be seen by examining the “Statement [Table]” within the US GAAP Taxonomy or even better, the “Hypercube [Table]” of the FINREP taxonomy.

The FINREP taxonomy took the most extreme route using one [Table] and one [Table] only throughout their entire taxonomy. They did this to specifically push all semantics of the meaning of a group of information onto the network which contains the hypercube. One can be sure that the network describes the information 100% of the time because (a) each [Table] is called exactly the same thing and (b) because each network could only possibly contain one [Table] because using the same [Table] name within a network would cause modelling conflicts (and remember, all [Table]s have the same name). The bottom line here is that the network carries all semantics for describing the information, there is no confusion.

By contrast, the US GAAP Taxonomy has the “Statement [Table]” which is used on the balance sheet, the income statement, the cash flow statement, and the statement of changes in equity. As such, one can only know which “Statement [Table]” you are working with by using the network.

Further, most but not all other [Table]s in the US GAAP Taxonomy are unique. What is more, not everything is modelled as an explicit [Table] and therefore there are many “No Table [Table]s” (see the preceding section).

1.1.6. Which Approach is Best?

All this distils down into three possible options:

- **Use explicit unique [Tables].** This option works well, and in fact it is the option which I believe is the most reasonable. By taking this approach you can ignore networks altogether, relegating networks to the role of syntax needed only for avoiding modelling conflicts. And because you can ignore the network, you can be sure the [Table] describes the information set and each [Table] being unique, each information set is unique.
- **Use explicit but only one [Table] for everything.** This option works well also because it is clear that the network carries all semantics for describing a set of information. The down side is that you have to create metadata such as the “number” and “category” used by the SEC to help organize those networks.
- **Mixed model.** If [Tables] are not unique and if [Table]s are not explicit (i.e. you have “No Table [Table]s”), you have to rely on both networks and tables to identify which information you need to work with. This can be both cumbersome for software and for users. A mixed model such as this does not appear to make much sense and should be avoided, all things considered.

There are no real benefits of having [Table]s names which can be used in more than one place, yet there are significant benefits of unique [Table] names.