The Anatomy of an XBRL Taxonomy

Jim Richards
Murdoch Business School
Murdoch University
South Street
MURDOCH
Western Australia 6150

Co-Chair, Education Working Group XBRL Australia
Vice-Chair, Education Working Group, XBRL International

Working Paper, May 2002
Contact: j.richards@murdoch.edu.au

1.0 Introduction

To date, two versions of the Specification for XBRL. According to Hampton and vun Kannon (2001, p. 7) the changes made when moving arose from two factors:

- New technology such as XML Schema and XML Linking, and
- Implementation experience

Under Specification 1.0, all of the details for a taxonomy were contained in a single file (taxonomy.xsd). The .xsd indicated that this was a taxonomy/schema file rather than a normal XML file.

An important feature found in Specification 1.0 and, therefore, in the taxonomy/taxonomies created under that specification was that of roll-ups. The details of how the roll-ups were to be performed and the labels associated with each of the elements were contained in the taxonomy file. Further information also contained in the schema were the references to the accounting standards to which each of the elements could be related.

The introduction of Specification 2.0 has seen a taxonomy grow from a single file to six files. Most of this has been possible due to the use of XML linkbases (XMLLink).

Unless otherwise specified, element naming in these notes is consistent with the element naming scheme used in the draft Australian taxonomy released on April 30, 2002.

2.0 A Specification 1.0 Taxonomy Sample Element

Here is the details for a sample element (currentAssets.otherCurrentAssets) from the US-GAAP-CI-2000-07-031.xsd:

```
<element name=”currentAssets.otherCurrentAssets” type=”xrbl:monetary”>
  <annotation>
    <appinfo>
      <xbrl:rollup to=”currentAssets” weight=”1” order=”10”/>
      <xbrl:label xml:lang=”en”>Other Current Assets</xbrl:label>
      <xbrl:reference name”ARB” number=”43” chapter’”3A” paragraph=”6”/>
     </appinfo>
  </annotation<
</element>
```

Most of the contents would be self-explanatory but the rollup section probably needs a little explanation.

2.1 Rollups

These are probably best explained by referring to a simple Statement of Financial Position (Balance Sheet) example. Here is the Current Assets segment of the Statement of Financial Position.

Current Assets		
Cash and cash equivalents	$	520,127
Receivables		95,702
Merchandise inventories		1.060,788
Other current assets		33,660
Total current assets	$	1,710,277

To make the comparison easier to understand the entry in the taxonomy for this small example would be as follows:

```
<element name=”currentAssets.otherCurrentAssets” type=”xrbl:monetary”>
  <annotation>
    <appinfo>
      <xbrl:rollup to=”currentAssets” weight=”1” order=”4”/>
      <xbrl:label xml:lang=”en”>Other Current Assets</xbrl:label>
      <xbrl:reference name”ARB” number=”43” chapter’”3A” paragraph=”6”/>
    </appinfo>
  </annotation<
</element>
```

All that has changed is that ‘order=”10”’ has changed to ‘order=”4”’. This is because our simple example only contains 4 current assets. The value of the order attribute tells the software displaying the output that ”Other current assets” is the fourth current asset to be displayed.

In the report displayed above the last line – Total current assets – would not normally be displayed and the total for current assets would be shown at the top of the list. An HTML output of the above segment might look like this:

-	Current Assets	1,710,277
	Cash and cash equivalents	520,127
	Receivables	95,702
	Merchandise inventories	1.060,788
	Other current assets	33,660

The ”-” sign next to Current Assets indicates that the full detail for Current Assets is being shown. By clicking on the ”-” the details of Current Assets would be hidden and all that would be displayed would be:

+	Current Assets	1,710,277

The ”+” sign indicates that further detail could be found by clicking on the ”+” sign to expand the display back to that shown originally.

The attribute 'weight="1"' simply meant that when the rollup occurred the value for "Other current assets" remained at its original value when the addition occurred to get the total for "Current Assets".

Another point to note here that is also applicable to a Specification 2.0 instance document is that all amounts are entered as positive values. The weight assigned to an element determines whether the amount is added to subtracted to obtain t e value for its parent.

3.0 A Specification 2.0 Taxonomy Sample Element

The entry in the taxonomy for the same element, but using Specification 2.0 and the naming convention from the IAS taxonomy development group, would be as follows:

```
<element
        id="lia.cur.otr"
        name="lia.cur.otr"
        type="xbrli:monetaryItemType"
        substitutionGroup="xbrli:item" />
```

As you can see a lot of the detail has been removed. The "missing" data has now been transferred to other files that make up a complete taxonomy.

The taxonomy file (assumed in these notes to be **example.xsd**) would contain a similar entry for each of the elements in the taxonomy.

4.0 The Anatomy of a Specification 2.0 Taxonomy

According to IAS-GAAP-CI-BASE-2002-04-05 documentation (2002, p. 8), the PFS (Primary Financial Statements) taxonomy consists of a package of interrelated files:

- XML Schema File (.xsd file)
- XBRL Linkbases (.xml files) for:
    - Labels
    - References
    - Presentation
    - Calculations, and
    - Definitions

All of the data contained in the single taxonomy from Specification 1.0 is now found in different files that are all interconnected using XML linkbase technology.

Each of the linkbase files is named using the name of the taxonomy file and the type of file that the data contains. For example, the taxonomy file used in these notes is named **example.xsd**. Therefore, all of the various file names start with "example". This is followed by an underscore ("_") and then the type of linkbase file. This means that the file containing the element labels would be named **example_labels.xml.**

Sample entries from each the files and explanations of those entries are contained in the following sections. The element "Other current assets" will be used for all of these explanations.

Note: All of these notes assume that the taxonomy file (the .xsd file) is named **example.xsd**.

4.1 The Labels File

The labels file contains the full description for each of the elements contained in the taxonomy. The name for this file would be example_labels.xml.

There is a minimum of 4 entries for each element if a single language is supported.

4.1.1 Locator Entry

The first entry is required to locate the element in the labels file. This entry is only required once in the file even if multiple languages are supported by the taxonomy.

The entry is as follows:

```
<loc xlink:type="locator"
        xlink:href="example.xsd#otherCurrentAssets
        xlink:label="example_otherCurrentAssets/>
```

4.1.1.1 The type entry

This line indicates that the xlink type is being used to locate a specific element from the taxonomy file..

4.1.1.2 The href entry

This line indicates the reference to the element name in the taxonomy. The data between the quote marks contains two parts separated by a "#":
the taxonomy file name (example.xsd)
the element name within that taxonomy file (otherCurrentAssets)

4.1.1.3 The label entry

This line indicates the general label (not language specific) for the element within the labels file (example_labels.xml). This will be used in later entries within the file.

4.1.2 Label Entry

The next entry contains the label for the specific element for a specific language. The entry is as follows:

```
<label xlink:type="resource"
        xlink:label="example_otherCurrentAssets_en"
        xlink:role=http://www.xbrl.org/linlprops/label/standard
        xml:lang="en">Other Current Assets</label>
```

This entry requires a bit more explanation.

- The first line indicates that we are now creating a resource entry rather than an entry to find an element.
- The second line contains three parts:
  - example – name of taxonomy
  - otherCurentAssets – name of element within the taxonomy
  - en – two character language code for English.

  This enables the XML processor to find the label for each language.
- The third line refers to what type of label (explained in the Specification) is being created.
- The fourth line refers to the language using a standard two character abbreviation (en) for English. The actual label (Other Current Assets) is between the ">" and the "<".

4.1.3 The Arc Entry from the Taxonomy to the Label

This is the first of two entries required. If only one entry was contained in the file the link would only be in one direction. The two entries mean that we can move backwards and forwards between the taxonomy file (example.xsd) and the labels file (example_labels.xml).

This entry allows the XML processor to replace the element name with the element label contained in the previous label entry. Whenever the instance document refers to the element name, this entry allows the processor to replace it with the element's label.

The entry is:

```
<labelArc     xlink:from="example_otherCurrentAssets"
              xlink:to="example_otherCurrentAssets_en"
              xlink:arcrole=http://www.xbrl.org/linlprops/arc/element-label
              xlink:type="arc" xlink:show="embed" xlink:actuate="onRequest"/>
```

- The first line indicates that we want to link from the element otherCurrentAssets in the example taxonomy file (example.xsd).
- The second line indicates that we want to link to the label entry for otherCurrentAssets using the English (en) label in the labels file.
- The third line indicates the type of entry we are making as defined by the XBRL specification – linking from an element to a label.
- The final line indicates three things:
  - The link is a connect (arc)
  - The label is to replace the element name (embed)
  - The label is to replace the element name only when requested.

4.1.4 The Arc Entry from the Label to the Taxonomy

The final entry is the reverse of the third entry. It fulfils the same role but allows the link to be reversed. The entry is:

```
<labelArc      xlink:from="example_otherCurrentAssets_en"
               xlink:to="example_otherCurrentAssets"
               xlink:arcrole=http://www.xbrl.org/linkprops/arc/label-element
               xlink:type="arc" xlink:show="embed" xlink:actuate="onRequest"/>
```

4.1.5 The Full Entry Required

Here is the entry for an element in its entirety:

```
<loc xlink:type="locator"
       xlink:href="example.xsd#otherCurrentAssets
       xlink:label="example_otherCurrentAssets/>
<label xlink:type="resource"
         xlink:label="example_otherCurrentAssets_en"
         xlink:role=http://www.xbrl.org/linkprops/label/standard
         xml:lang="en">Other Current Assets</label>
<labelArc      xlink:from="example_otherCurrentAssets"
               xlink:to="example_otherCurrentAssets_en"
               xlink:arcrole=http://www.xbrl.org/linkprops/arc/element-label
               xlink:type="arc" xlink:show="embed" xlink:actuate="onRequest"/>
<labelArc      xlink:from="example_otherCurrentAssets_en"
               xlink:to="example_otherCurrentAssets"
               xlink:arcrole=http://www.xbrl.org/linkprops/arc/label-element
               xlink:type="arc" xlink:show="embed" xlink:actuate="onRequest"/>
```

4.1.6 Adding Another Language

If support for multiple languages is required then there could be an entry in the example_labels.xml file for each language. Another approach to multiple languages would be to create separate files for each language. When creating a displaying a particular instance document, then the specific language file would be linked at the time the document is displayed or transformed.

4.1.6.1 Adding a New Language in the Same Labels File

Adding a second language requires three additional entries for each element. The only entry that does not need to be repeated is the "locator" entry – the first one. Each of the other entries contains a reference to the specific language being used for the element. To add an entry to display the same information but in Mandarin (lang="cn") we would need to add the following entries:

```
<label xlink:type="resource"
         xlink:label="example_otherCurrentAssets_cn"
         xlink:role=http://www.xbrl.org/linkprops/label/standard
         xml:lang="cn">流動資金</label>
<labelArc      xlink:from="example_otherCurrentAssets"
               xlink:to="example_otherCurrentAssets_cn"
               xlink:arcrole=http://www.xbrl.org/linkprops/arc/element-label
               xlink:type="arc" xlink:show="embed" xlink:actuate="onRequest"/>
<labelArc      xlink:from="example_otherCurrentAssets_cn"
               xlink:to="example_otherCurrentAssets"
               xlink:arcrole=http://www.xbrl.org/linkprops/arc/label-element
```

xlink:type=”arc” xlink:show=”embed” xlink:actuate=”onRequest”/>

The actual labels file would not be able to display the actual Mandarin characters shown here as all XML files are ascii/text files. However, the text character codes would be stored and the specific font to be used to display the characters would be part of the style sheet used to display the label.

The entry in the example_labels.xml file would look like this: □□□□. However, the use of style sheets specifying the correct Mandarin font would be able to display the label correctly.

4.1.7 How Does it Work

Whenever an instance document, which contains only the elements and the data, an XSLT style sheet would be used to display the data, say, in Internet Explorer. To switch between the languages requires the use of two style sheets. One style sheet would indicate to use the English language while the other style sheet would indicate to use Mandarin. By clicking on an appropriate link in the HTML document generated the user could switch between the two languages. Clicking on the link would simply change the style sheet that is being used to display the contents of the same instance document.

4.2 The Reference File

The reference file contains the data relating to the Australian Accounting Standard (AASB reference) or the International Accounting Standard that specifies that the element needs to be disclosed. In the explanation that follows reference is made to the AASB standard.

The entries are similar to those contained in the labels file, but the links now relate to the authoritative literature.

4.2.1 Locator Entry

The first entry is required to locate the element in the labels file. The entry is as follows:

```
<loc xlink:type=”locator”
        xlink:href=”example.xsd#otherCurrentAssets
        xlink:label=”example_otherCurrentAssets/>
```

This is exactly the same as for the labels file. It allows the XML/XBRL processor to locate details for the specific element in this file.

4.2.2 Reference Entry

This entry is made up of a number of parts that always appear together. This is referred to as a "tuple". A tuple is an element that consists of a number of sub-elements. These sub-elements must be present every time the main element is present. Null values for a sub-element are allowed in an instance document.

The format of the reference is:

<reference xlink:type=”resource”
xlink:label=”example_otherCurrentAssets”
<name>AASB</name>
<number>1040</number>
<paragraph>7</paragraph>
<subparagraph>5</subparagraph>
</reference>

The reference entry is consists of four sub-elements:

- name
- number
- paragraph
- subparagraph

The above entry indicates that otherCurrentAssets are required by AASB1040(7)(5).

4.2.3 The Bi-Directional Links (Arcs)

As for the label arc, two entries are required to allow the user to link in both directions. The entries are:

<referenceArc
xlink:from=”example_otherCurrentAssets”
xlink:to=”example_otherCurrentAssets_reference”
xlink:arcrole=http://www.xbrl.org/linkprops/arc/element-reference
xlink:type=”arc” xlink”show=”embed” xlink:actuate=”onRequest”/>

<referenceArc
xlink:from=”example_otherCurrentAssets_reference”
xlink:to=”example_otherCurrentAssets”
xlink:arcrole=http://www.xbrl.org/linkprops/arc/reference-element
xlink:type=”arc” xlink”show=”embed” xlink:actuate=”onRequest”/>

Using a style sheet, it is possible to link each element with the authoritative literature that requires the disclosure of each element.

4.3 The Presentation File

The presentation file contains information on the order in which elements would normally appear in a financial statement. This is not always the way that information must appear, but it represents the way that the taxonomy creators consider it will normally be shown. It is possible for individual users to create their own presentation files and use their preferred order of presentation.

If we use the Current Assets segment (from section 2.1 Rollups), the Other Current Assets element (to refer to its label) is part of Current Assets. Our example taxonomy has four current assets:

- Cash and cash equivalents
- Receivables
- Merchandise Inventories
- Other Current Assets

If the above order is required then there are two ways it might be achieved. Firstly, the order would be determined by the order in which they appear in the instance document. Secondly, by having a presentation file associated with the taxonomy, the order can be determined by the presentation file, regardless of the order in which the items appear in the taxonomy.

4.3.1 Locator Entry

Using the above example, the first entry for Other Current Assets in the Presentation file would be:

```
<loc    xlink:type="locator"
        xlink:href="example_taxonomy.xsd#otherCurrentAssets"
        xlink:label="example_ otherCurrentAssets "
        xlink:title="example_ otherCurrentAssets "/>
```

Again, this part of the entry allows the XML/XBRL processor to locate the specific item within in the presentation file (example_presentation.xml).

4.3.2 PresentationArc Entries

Again there are bi-directional links within the presentation file for each element in sub-group. For Other Current Assets the entries are:

```
<presentationArc xlink:type="arc"
        xlink:from="example_ otherCurrentAssets "
        xlink:to="example_currentAssets"
        xlink:show="replace"
        xlink:actuate="onRequest"
        xlink:title="presentation: example_ otherCurrentAssets up to example_currentAssets"
        order="4"
        xlink:arcrole="http://www.xbrl.org/linkprops/arc/child-parent"/>

<presentationArc xlink:type="arc"
        xlink:from="example_currentAssets"
        xlink:to="example_ otherCurrentAssets "
        xlink:show="replace" xlink:actuate="onRequest"
        xlink:title="presentation: example_currentAssets down to example_ otherCurrentAssets "
        order="4"
        xlink:arcrole="http://www.xbrl.org/linkprops/arc/parent-child"/>
```

The first link indicates the child-parent link from otherCurrentAssets up to currentAssets. The second link is the parent-child link from currentAssets down to otherCurrentAssets.

In both instances the order is equal to 4. This indicates that under normal circumstances Other Current Assets (element otherCurrentAssets) would be the fourth of all of the Current Assets. It is important that the value 4 be the same in both entries.

As you can imagine, having to do this manually for each element in a large taxonomy is a rather long process. You need to be very aware of the structure of the financial statement being reported and build the necessary links in the presentation file. The process can be made simpler by using software from the various software developers creating taxonomy building software.

4.4 The Calculation File

The Calculation file performs the same function as the roll-ups discussed previously for Specification 1.0. Rather than being included in the taxonomy file, they have also been separated out into a separate file (example_calculations.xml). The process and entries are very similar to that of the Presentation file, so I will not provide a detailed explaination.

The difference between an entry in the Calculation File and the Presentation file is that the "order" attribute is replaced by the "weight" attribute. To add a value to the other values with the same parent, the element would be assigned a weight of "1". To subtract a value from the other elements in obtaining the value for the parent element, a weight of "-1" would be assigned. Remember that values are entered in instance documents as positive values.

Here is an example from a Statement of Financial Performance (P&L Statement) – Gross Profit. In the usual presentation of Gross Profit, the entries are shown as:

	Sales	355,450
-	Cost of Goods Sold	230,320
=	Gross Profit	125,130

In an XBRL display of this same data, it would be shown as:

	Gross Profit	125,130
=	Sales	355,450
-	Cost of Goods Sold	230,320

This indicates that Gross Profit is the parent and Sales and Cost of Goods Sold are the children. The weights in the Calculation file for these elements would be:

			Weight
	Gross Profit	125,130	
=	Sales	355,450	1
-	Cost of Goods Sold	230,320	-1

4.5 The Definition File

The definition file creates all of the parent-child relationships formerly found in single taxonomy file from Specification 1.0

4.5.1 Parent-Child Relationships

Each parent-child relationship in the taxonomy requires 3 entries – a locator entry, a parent-child entry and a child-parent entry

```
<loc xlink:type="locator"
        xlink:href="example.xsd#otherCurrentAssets"
        xlink:label="example_otherCurrentAssets"/>

<definitionArc xlink:type="arc" xlink:show="replace" xlink:actuate="onRequest"
        xlink:from="example_totalCurrentAssets"
        xlink:to="example_otherCurrentAssets"
        xlink:arcrole="http://www.xbrl.org/linkprops/arc/parent-child"/>

<definitionArc xlink:type="arc" xlink:show="replace" xlink:actuate="onRequest"
        xlink:from="example_otherCurrentAssets"
        xlink:to="example_totalCurrentAssets"
        xlink:arcrole="http://www.xbrl.org/linkprops/arc/child-parent"/>
```

4.5.2 Equivalency (Same-As) Relationships

The initial draft Australian taxonomy released in April 2002 replicates concepts from the draft IAS taxonomy and uses different element names for the same concepts. In this instance, the definitions file allows the taxonomy developer(s) to convey to users that the concepts in the two taxonomies are equivalent and either element name can be used provided the taxonomies are correctly referenced.

As an example Australia now refers to the Balance Sheet as the Statement of Financial Position. Internationally it is still referred to as the Balance Sheet. Additionally, the draft Australian taxonomy does not follow the triad naming convention used in the draft IAS taxonomy. Using this example. the element for the Balance Sheet in the Australian taxonomy is "statementFinancialPosition" where as the same concept is the "bst" element in the IAS taxonomy.

Let's take a look at the entries in detail for the Other Current Assets example we have been using. For brevity I have taken a couple of liberties here in naming the elements compared to those used in the file from the Australian taxonomy.

The correct term for these connections is an Equivalency Relationship. For each equivalency relationship there are two required entries (the element entries) and each entry consists of two parts (the locator entry and the definition link entry). Again, the dual entries for each concept allow the users to move in both directions – from the IAS to the Australian and from the Australian to the IAS.

Here are the entries to link from the Australian taxonomy to the IAS taxonomy. The references beginning with "example" assume that this file contains the Australian taxonomy and those beginning with "iascf" refer to the IAS taxonomy.

```
<loc xlink:type="locator"
        xlink:href="example.xsd#otherCurrentAssets"
        xlink:label="example_otherCurrentAssets" />
```

The above entry is to locate element "otherCurrentAssets in the Australian (example) taxonomy.

```
<loc xlink:type="locator"
        xlink:href="iascf.xsd#ast.cur.otr"
        xlink:label="iascf_ast.cur.otr" />
```

The above entry is to locate the element "ast.cur.otr" in the IAS taxonomy.

```
<definitionArc xlink:type="arc" xlink:show="replace" xlink:actuate="onRequest"
        xlink:from="example_otherCurrentAssets"
        xlink:to="iascf_ast.cur.otr"
        xlink:arcrole="http://www.xbrl.org/linkprops/arc/equivalency" />
```

The above entry links from the Australian taxonomy to the equivalent element in the IAS taxonomy.

```
<definitionArc xlink:type="arc" xlink:show="replace" xlink:actuate="onRequest"
        xlink:from="iascf_ast.cur.otr"
        xlink:to="example_otherCurrentAssets"
        xlink:arcrole="http://www.xbrl.org/linkprops/arc/equivalency" />
```

The above entry links from the IAS taxonomy to the equivalent element in the Australian taxonomy.