Table of contents

1. Introduction....2
1.1. Terminology....2
2. Criteria....3
2.1. Consistent....3
2.2. Pragmatic/ Intuitive....3
2.3. Efficient....3
2.4. Extensible....4
3. Document structure....5
3.1. xtab element....5
3.2. table element....5
4. Table Geometry....6
4.1. ‘Cube’....6
4.2. Dimensions....6
4.3. Pagination of edges....10
5. Table Contents....11
5.1. Table structure....11
5.2. Statistics....13
5.3. Statistic types....15
5.4. Cell values....15
6. Names and titles....24
7. Handling of text....25
8. Metadata....27
8.1. Operational metadata....27
8.2. Project and table metadata: controls and controltypes....27
9. Results of significance texts....29
9.1. Detailed significance results....44
10. Deferred and out-of-scope issues....46
10.1. Canonical values....46
10.2. Rich text....47
10.3. Styles....48

1.Introduction

This document describes the decisions leading to the format for XML tables (hereafter referred to as XtabML) defined formally by:

- DTD (XtabML.dtd)
- Implementation notes (implementation_notes.doc)

and based on the requirements set out in:

- Requirements (requirements.doc)

We first set out the criteria we have used when selecting from alternative approaches, and then show how these apply to the major features of the format.

In other words, the document attempts to answer for people reading the XML design: "why on Earth did they do it this way?"

1.1. Terminology

There are two software domains concerned with cross-tabulations; survey tabulation systems and OLAP systems, particularly the MS Office manifestation of OLAP in pivot tables. This table lists key tabulation concepts and the names used for them in XtabML and their (approximate) equivalents in other system

XtabML term	**Concept**	**Alternative names**
table	2, 3 or higher dimensional cross-tabulation	Cube (OLAP), table report (Pulsar)
dimension	One side of the tabulation vector, matrix, cube or hypercube	
element	Categories for which data are tabulated (components of a dimension)	Row/Column (TabsML), element (Quantum)
edge	Hierarchical structure of categories for presentation (an organisation of a dimension for presentation)	Hierarchy (OLAP), axis, RowGroup/ColGroup (TabsML)
statistictype	A type of aggregated value shown in a table, or a value derived from aggregated values	Statistic (Pulsar), Value (Star), Nxx (Quantum), Measure (OLAP)
statistic	Usage of a **statistictype** in a table	
control	A text accompanying the data in a table	Various
controltype	A type of control text, e.g. footnote.	TX (Star)

2.Criteria

There are several criteria that we have tried to use to motivate the design decisions. They are not mutually consistent; we can't satisfy them all. They don't have any absolute hierarchy: we shouldn't sacrifice one completely to make a small gain in another.

Where we have noticed alternatives in creating the design we have tried to explicitly evaluate them against these criteria, and take a balanced decision. This is necessarily subjective.

2.1. Consistent

Where an approach is used for one part of the design, the same approach should be used elsewhere. This applies to, for instance, use of elements versus attributes and naming conventions, and completeness; where a feature appears in one context, it should appear in all similar contexts.

Consistency also lies in reusing styles and names that have been used in existing standards.

XML elements are reused wherever possible, as this facilitates the reuse of XSLT templates.

2.2. Pragmatic/ Intuitive

MR tables are a situation dominated by one standard case: a two-dimensional table with a few very different statistics used (typically no more than 5 or 6) and some of these used only small parts of the table. Empty cells (sparse tables) are very much the exception rather than the rule.

The standard should support this case well and it should not create an overhead for ordinary tables in its provision for the more unusual cases of 3D and higher dimensioned tables.

2.3. Efficient

We want a standard that is practically useful. There are two main concerns for efficiency.

2.3.1. Bandwidth: Economy of size

The documents will be transmitted from server to client and should be of a magnitude that makes this practicable.

One very simple expedient we have used extensively is to adopt single character element names for all the most common elements. This reduces both document size and clutter that obscures content with mark-up.

2.3.2. Processing: XPath friendly

When designing the structure of the document we must consider likely operations that applications will undertake, and ensure that the XPath expressions will be simple to write and fast to execute with real XSLT and XQuery processors.

XSLT is universally supported in browsers today, but it is likely that in the near future XML processing will be principally through XQuery.

The release of SQL Server 2005 is imminent at the time of writing. The major new feature of SQL Server 2005 is its support for XML as a native data type and XQuery as a retrieval mechanism for XML data.

XQuery (http://www.w3.org/TR/xquery/) is a W3C standard that is finding its way into XML databases and processors in general, not just SQL Server.

XQuery is far closer in appearance to a traditional programming language than XSLT but shares XPath as the basis for query syntax.

XPath-friendliness remains a key requirement for any XML format for the foreseeable future.

2.4. Extensible

We should not include features that are not capable of extension to support a more sophisticated model of a cross-tabulation in future. In other words, the standard is limited by what is omitted, not by what is included.

3.Document structure

The requirements specify only that an XML table document should contain one table.

This was motivated primarily by the expectation that processing large documents would be slow. However initial experiments showed that individual tables in XtabML format occupy only a few kilobytes, and this shrinks by a factor of 10 using compression (as in a web server offering gzip compression in its HTTP responses).

Consequently the XtabML design was altered to allow authors to put a report consisting of multiple tables in one document.

XML element names are shown in bold throughout this document, e.g. **statistictype**.

3.1. xtab ***element***

The document has an **xtab** root element. The root element identifies the version of the XtabML format used in the document.

Within the root element are elements common to all tables:

- operational metadata: optional **origin**, **date** and **time** elements
- **controltype** definitions
- **statistictype** definitions
- **control** elements with text common to all tables, e.g. project title.
- **language** elements describing languages appearing in the document

and one or more **table** elements with information specific to one table.

3.2. table ***element***

Within each **table** element are:

- **edge** elements: describing rows, columns, and planes etc as appropriate.
- **control** elements with text specific to the table, e.g. filter title.
- **statistic** elements: one for each statistic type used in the table
- **data** element containing all the data values for the table.

4.Table Geometry

4.1. ‘Cube’

An XtabML table may have one, two, three or more dimensions. A one dimensional table has several columns. A two dimensional table has columns within rows. A three dimensional table has columns within rows within planes; a four dimensional table has columns within rows within planes within cubes etcetera.

So there is an underlying data ‘cube’ like in an OLAP system and the XtabML format is storing the cells of the cube one-by-one in a specific order.

XtabML models a ‘cube’ that has several dimensions (rows, colums, planes and hyperplanes) and several statistics in each cell. There are several statistic types associated with the cube and any of them may be present in each cell.

Within the dimensions there are element and summary items that collect data. The word ‘element’ is used with the same meaning as it has in OLAP systems (dimensions are made of elements) and in Quantum (axes are made of elements). Pulsar Web and STAR do not have an equivalent term. ‘category’ comes close, but this is within a single or multiple variable; a quantity variable or a total within a Pulsar edge is not a category, but it is an element in the XtabML sense. STAR also has a term ‘element’ which roughly corresponds to a group in XtabML.

Unfortunately in the XML context ‘element‘is also used for the region of a document enclosed by an element tag. We persist with the use of the term element because it is well established with this meaning in existing major tabulation systems (OLAP in general, and Quantum).

4.2. Dimensions

There is enough similarity in the data needed to describe the dimensions (row, column, plane etc.) to justify a single **edge** XML element type, with one instance defining each dimension. The row edge is identified as the edge with the attribute **axis=’r’**; similarly **c** for columns, **p** for planes. Fourth and higher dimensions have edges with **axis=’p’ and level=’4’** or **‘level=5’** and so on.

If you inspect a typical cross-tabulation the rows and columns are usually structured into groups.

For example, a breakdown may involve several variables such as age, sex and region. This implies a group relating the categories for each variable that is needed for formatting purposes:

- the groups require headings, centred above the columns of the group.
- a pagination algorithm should try to avoid splitting a group between two pages

Similarly, a table may present several scaled variables on a page with the frequency distribution of each variable and mean scores below. There may also be a subtotal for each variable because not each respondent gave a score for every variable. In this situation we still wish to keep the rows of a variable together on a page, but if the variable has to be split between pages we wish to reprint the subtotal on each page.

Therefore, as well as defining the individual rows and columns we need to record the group structure of the dimensions.

In XtabML this is done by creating a hierarchy of **group** elements within each **edge**. **group** elements contain several members, each being one of:

- **group**: groups may be nested
- **summary**: a special row, column or plane that summarises the members of the group in some way
- **element**: a row, column or plane containing ordinary data.

summary XML elements may have a **type** attribute specifying the way in which they summarise the group. The sorts of summary we anticipate supporting eventually might include:

- **total**: everyone potentially qualifying for the group
- **net**: everyone who actually qualifies in one or more of the elements
- **other**: everyone who qualifies for the group but in a way that doesn't merit an individual element
- **none**: everyone who potentially qualifies for the group but hasn't qualified for any element of the group

These distinctions will be especially useful once XtabML supports canonical values and table manipulation, though if an application can include them in XtabML output this is still valuable as sometimes different types of summary motivate different formatting.

Presently the **type** of **summary** elements is an optional attribute with only one defined interpretation: **type="xs:base"** is used to distinguish summaries that are the base for percentages (as used, for example, in the XtabML standard statistic types).

In XtabML **summary** is really just a special kind of **element**. However **summary** elements are very important to distinguish in table formatting. We use a different XML name so summaries are conspicuous both in documents and in the source code of applications processing documents.

4.2.1. Application of groups in an edge used for columns:

```
<edge axis="c">
      <group>
            <summary type=”xs:base”>
                  <t>Total</t>
            </summary>
            <group>
                  <t>Region</t>
                  <element><t>West Coast</t></element>
                  <element><t>South</t></element>
                  <element><t>Mid West</t></element>
                  <element><t>Mid Atlantic</t></element>
            </group>
            <group>
                  <t>Age</t>
                  <element><t>-35</t></element>
                  <element><t>36-55</t></element>
                  <element><t>56+</t></element>
            </group>
            <group>
                  <t>Education</t>
                  <element><t>-Coll. Grad.</t></element>
                  <element><t>Coll. Grad.+</t></element>
            </group>
            <group><t>Income</t>
                  <element><t>-$45K</t></element>
                  <element><t>$45K+</t></element>
            </group>
      </group>
</edge>
```

Notes:

- this edge is associated with columns (**axis=’c’**)
- this edge like any other contains one top-level group
- the top-level group contains one summary total category followed by four sub-groups
- the edge contains 1 total column, 4 region columns, 3 age columns, 2 education columns and 2 income columns, i.e. 12 columns in all.
- the columns are stored in the data section of the table in this order and (by default) will be printed in this order.
- XtabML is concerned with structure and content not presentation. A rendering engine will use the structural information to choose an appropriate presentation. A conventional presentation of the edge in a formatted table will use spacing to separate the groups, and will render the group heading centred above the columns of the group, probably emphasising the heading by emboldening and forcing to upper case.
- in a Pulsar Web context, this is an edge with a total followed by four categorical variables.

4.2.2. Application of groups in an edge used for rows

This example is presented twice, once to show the potential of the summary concept, but requiring the XtabML export to know the details of the incrementation instructions, and another that can reasonably be produced by an XtabML export that is working only with information about the accumulated table.

The edge presents a scaled variable. It has a total row before the individual categories and a statistics row at the end that will show the total answering, mean score and other statistics.

The edge definition does not specify a selection of statistics to be shown for each element or summary. This information comes in the data section of the XtabML table. The **edge** element just provides a context to show whatever statistics have been accumulated. This is necessary because complex table designs can have different sets of statistics in different columns – for example a breakdown that is a mixture of categorical and quantity variables.

Simple version:

```
<edge axis="r">
      <group>
            <summary>Total</t></summary>
            <element><t>Read all</t></element>
            <element><t>Read most</t></element>
            <element><t>Read some</t></element>
            <element><t>Read none</t></element>
            <element><t>Don't remember</t></element>
            <summary><t>All answering</t></summary>
      </group>
</edge>
```

Enhanced version:

```
<edge axis="r">
      <group>
            <summary type="xs:base"><t>Total</t></summary>
            <element><t>Read all</t></element>
            <element><t>Read most</t></element>
            <element><t>Read some</t></element>
            <element><t>Read none</t></element>
            <summary type='none'><t>Don't remember</t></element>
            <summary type='net'><t>All answering</t></summary>
      </group>
</edge>
```

The enhanced version uses summary types to express all the relationships between the rows; the '**xs:base**' row includes all respondents, the '**none**' row includes only respondents not qualifying for any element, and the '**net**' row includes all respondents qualifying for any element.

XtabML for now does not mandate the use of a **type** attribute with **summary**. Implementors choosing to support summary types (and they can be useful to distinguish in pagination and formatting) are expected to use summary types consistently across all documents, as with statistic and control types. We believe that

the names suggested here are reasonable starting points – but only **xs:base** has been standardised.

The simple version has the same structure but merely distinguishes the first and final rows as summaries.

In some countries it is conventional to present totals as the final rather than first row of a table and XtabML supports this; summary items may appear at any position in a group including the beginning and end. It is typical to have more than one summary item for a group, as in this example. The initial summary item might show statistics unweighted base, weighted base and vertical percent (uniformly 100%). The final summary item might show the mean and standard deviation.

4.3. Pagination of edges

The group structure of edges provides information required for pagination. The basic algorithm for paginating an XtabML edge is as follows:

- show complete groups where possible.
- When groups have to be split, it should preferably be between subgroups.
- If a group has to be split between elements, then all pages of the group should contain all summary items of the group if possible, in the defined order (in other words, reprint total rows whether they are at the top or the bottom of the group).

These rules have to be supplemented with heuristics to avoid unbalanced page sizes. The intention in XtabML is to provide enough information about the relationships within rows and columns for a satisfactory pagination to be implemented, without the exporting application having to make any decisions about pagination.

5.Table Contents

5.1. Table structure

Cross tabulations are normally represented inside tabulation software by a multidimensional array. Arrays are not naturally hierarchical, and so the question arises in what order to write the cells to a sequential format such as XML.

5.1.1. Linear

OLAP XML formats have taken the path of writing out individual cells within an all-embracing data element, identifying the cells by their subscript in the array. This has two very big drawbacks:

- subscript arithmetic is required to identify the row/column/plane of a cell. This is awkward and especially inconvenient in XPath.
- The data element is huge and selecting sub-elements will be very slow.

This approach has the advantage that sparse tables are represented quite efficiently; however sparse tables are uncommon in market research reports; where they do appear it is usually rows that are empty rather than columns.

TabsML uses a linear format explicitly identifying the row and column for each cell.

5.1.2. Nested

In practice the table is presented in a row/ column format, and this format is repeated for each plane. So there is already a natural ordering of the cells of the table, and if this is reflected in the XML design by nesting column elements within rows within planes, that will be efficient because the identification of a row is not repeated for each column in the row, and the identification of a plane is not repeated for each cell in the plane.

XtabML stores the cell values for a table within a **data** element. In a two dimensional table, this has one **r** (for row) element for each data item in the edge used as the **r** axis, containing all the statistic values for one row of the table. Within this **r** element there is one **c** element for each statistic type defined in the table. Finally within each **c** element are individual value elements (described below) giving the data values for that statistic for each column of the row, i.e. for each data item in the edge used as the **c** axis.

There are several reasons for this approach:

- **efficient access**. each row, statistic and column (subject to optimisation, below) is in the same position in its sequence of elements wherever it appears in the table. Therefore XPath expressions (or 3GL code) can select children by efficient positional access rather than slow searching for a child with a particular property.
- **opportunity for compression**. We have a convention that if all the statistics at the end of a row column have the same value, the application generating the XtabML can omit the data values at the end that repeat the final value. The main use of this is in the common situation where a statistic is undefined in

every column of a row, or is the same in every column (e.g. 100%). These rows are represented by their first value only.

- **readability.** Humans are not the primary audience for XtabML files but one great benefit of XML, HTML and HTTP is that they are text formats and you can inspect them by eye to see if their contents make sense rather than relying on a program. This is especially valuable in the debugging stage of interfacing two programs through a document. In XtabML you can see a whole row of data at a glance much like looking at a printed table.

5.2. Statistics

Many OLAP systems treat the statistics as just another dimension, with elements and labels, that can be presented in a report as rows, columns or planes, or nested within the row or column dimension.

Pulsar adopts this philosophy, to a limited extent. The statistics dimension in Pulsar Web is slightly privileged compared to the other dimensions; it may be nested in the row or column dimensions, while the other dimensions can't be nested within an axis (as they can in desktop Pulsar).

Consider a weighted column percent table including a scale distribution with means, standard errors and T-tests (in the conventional letter-for-significant difference format). There are six different statistics shown for this table:

1. unweighted total
2. weighted total
3. column percentage
4. mean
5. standard error
6. T-test

This an important example because very many market research reports use no more than these six statistics, and only a minority of tables in these reports will include the mean, standard errors and T-tests. Unweighted surveys of course demand one statistic fewer.

There are several features that make the statistics dimension different from the other dimensions of a table such as demographics.

- The statistics each have their own different meanings, data types and required presentation formats; whereas the difference between a male cell and a female cell, say, is just a difference in content.
- The same statistics crop up in many different tables, whereas the other dimensions are much more volatile.
- It is not useful to talk about a 'total' category for the statistics dimension
- It is not useful to propose ranking for the statistics dimension.
- Grouping doesn't arise in the statistic dimension.
- Statistics are often only meaningful for parts of the table; for example, means
- Statistics are often only useful for parts of the table; for example, unweighted counts

For these reasons this XML format defines the table statistics separately from the other dimensions. The other dimensions are defined in **edge** elements, while statistics are defined by **statistic** elements. Since the same statistic type may appear in many different tables, XtabML defines statistic types globally and these types and then applied in one or more tables. Each statistic type has a name and a display title (within

a t sub-element). Some names (but not titles) for common statistics are standardised by XtabML.

5.3. Statistic types

There is a wide variety of statistics generated by different tabulation systems. While there may be little debate about the meaning of a column percentage, there are differences between systems in the calculation of more complex statistics such as chi-square significance in the case of small samples. Rather than attempt to impose a catalog and mathematical definition of all statistic types, we have taken the view that each tabulation system generates its own repertoire of statistic types and has is own names for them that it can use consistently in the XtabML documents that it creates. Each XtabML table should contain a description of the different statistics to be found in the table. The description will include a name for the statistic and a title (within a t element so that the title can be exported in multiple languages if required).

In addition to this we have provided a set of standard names and calculation methods for common table statistics that hopefully have a universal meaning. Where an exporting application calculates statistics in the same way as the standard specifies then that application may (and is strongly encouraged to) use the standard name.

This approach means that a generic formatting system can present the statistics correctly labelled without needing to know their meaning.

The same statistic name should be used consistently in all XtabML documents generated by any one system. Therefore, where two statistic types are known to be the same in two different systems an XSLT can easily translate the names used by one system to the names used by another, even though the standard names have not been used.

5.4. Cell values

Within each combination of plane/row/column (a table cell) Pulsar Web calculates and displays the table statistics. As discussed, most tables will contain just two or three statistics. Very few will contain more than six.

5.4.1. Special situations

Applicability

Not all statistics are meaningful in all parts of a table. For example, in a table with a row edge containing a single variable and a quantity, the ‘mean’ statistic is not meaningful or ‘applicable’ in the rows associated with the single variable.

Visibility

Some statistics are not required in all parts of the table where they are meaningful. For example, unweighted totals are meaningful throughout a weighted table but not normally required except in the total row. In the formatted table these statistics should be ‘hidden’. Pulsar Web has business rules that hide statistics in most of the situations where they are inapplicable.

Another context where statistics may be invisible is blank or small row suppression. An exporter of XtabML may choose to delete suppressed rows entirely from the row edge definition, in which case of course the numbers are not

available to an application reading the XtabML. Alternatively, the statistics may be exported as 'hidden' values. An application processing the file can use or ignore the values as it chooses, but is aware of their 'hidden' status.

Undefined

Applicability and visibility are related to the structure of the table, independently of the actual data values arising. Certain statistics cannot be calculated for certain data values. For example, a percentage of a zero base requires a division by zero that has an undefined result. Some statistic values in some tables will be undefined for this or similar reasons.

All four combinations of visible/invisible and applicable/inapplicable arise in practice. Undefined values may arise in either visible or invisible contexts, though only in applicable contexts.

The following table from Pulsar/Web shows all these different contexts:

New cross report

			Total	Q22F : How would you rate your overall impression ?					Q22F : How would you rate your overall impression ?
				Excellent	Good	Average	Poor	Don't know/not applicable	
Total		Base	720	376	252	52	36	4	720
Q22E : How would you rate the general range and standard of the facilities available ?	Excellent	Base	288	232	52	4	0	0	288
	Good	Base	288	128	140	12	4	4	288
	Average	Base	96	8	40	32	16	0	96
	Poor	Base	32	0	12	4	16	0	32
	Don't know/not applicable	Base	16	8	8	0	0	0	16
Q22E : How would you rate the general range and standard of the facilities available ?		Base	720	376	252	52	36	4	720
		Mean	3.111	3.532	2.857	2.308	1.667	3	N/A
Q38 : Gender	Male	Base	188	92	80	8	8	0	188
	Female	Base	532	284	172	44	28	4	532

This table has 'Base' and 'Mean' statistics defined. In the rows there are two questions; one is a rating scale, the second is gender. The columns show one question, also a rating scale.

- Pulsar/Web represents the analysis of a rating score as a row or column following the rating question. 'Mean' values are only applicable in these special rows or columns. Pulsar Web hides all the inapplicable mean values for Q22E and shows means for Q22E only in the special row.
- Looking at the column for Q22F, this is where we might expect to find the mean rating of Q22F. The mean rating of Q22F is meaningful in all the rows except the special analysis row for Q22E. Pulsar Web has hidden the mean values in all the rows of this column; instances of values that are applicable but hidden.
- Finally, the mean of Q22F in the special row for Q22E is marked as inapplicable ('N/A'). This is an instance of an inapplicable but visible data value.

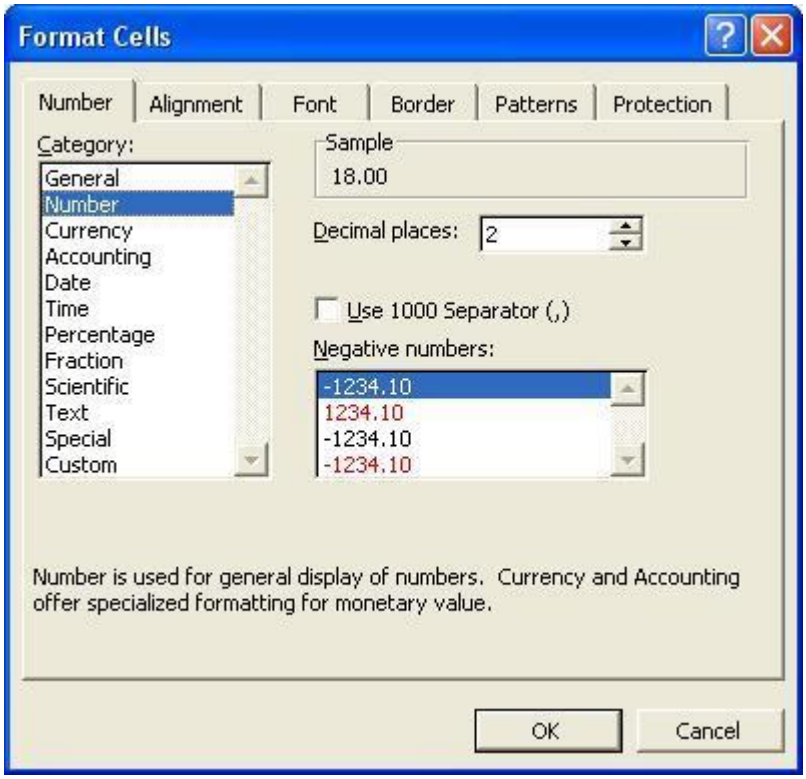
Arguably one might make different decisions about what to hide and what to show. XtabML allows an exporting program like PulsarWeb to include values that are hidden in its normal print format, marking them as hidden so that an importing program can maintain the exporters' decisions about hiding, or if required can modify them.

5.4.2. Representation of numbers

Tabulation systems prepare a table by accumulation from source data to produce 'raw' aggregate numbers that are then formatted for human inspection in a report. The formatting rules cover several areas, including:

- Specification of precision (decimal places)
- Conditionalisation (print negative numbers in red; print numbers with a base less than 25 as blank…)
- Locale-specific formatting conventions (decimal point character, digit grouping character, position of sign, use of percent character…)
- Scale factors

A good starting point to assess the scope of requirements to represent formatting is the Format…/Cells… dialog in Microsoft Excel.



In an XML representation of a table we may choose to store the ‘raw’ value and/or the formatted value. This gives rise to several considerations:

Canonical values

If the raw value is stored, then there needs to be a specified format to use in the XML. This is non-trivial, for several reasons:

- Statistics have data types: integer, real and (moot point) text that should be represented.
- We want the format to be efficient, e.g. the integer value one should preferably be stored as ‘1’; and the number of digits written should not exceed the accuracy of the value (i.e. not 0. 333333333333333!).
- There must be a representation for special values such as ‘undefined’ for a mean or percentage with no sample. This is a different concept from ‘not available’. ’not available’ is a property of the context.in the table; ‘undefined’ values are a property of the numbers in the context.
- Conventional presentation of certain statistic types, notably the results of t-test comparison of means in columns of a table, results in a string appearing that is really a representation of many statistics, i.e. the comparison of each column with each other columns.

In this document we refer to a text denoting the raw value of a statistic as the **canonical** value, as opposed to the **formatted** value that is generated by the application creating the tabulation, using its formatting rules. A canonical value should have a representation that is unchanged whatever the formatting rules or locale. For now XtabML does not provide a capability to store canonical values.

Table manipulation

It is likely that some uses of the XML format will involve arithmetic operations on the table values. This is only practicable if canonical values are stored, because:

- formatted numbers lose accuracy in rounding
- recovering the accurate value from formatted numbers involves running the formatting business rules in reverse; this is not easy and sometimes not possible.

Reformatting

If only canonical values are stored, then the only way an application can recreate the formatted values is by using the same formatting rules as the original tabulation system. This implies both that the XML document can model the rules whatever their complexity and that applications processing the XML document can interpret and apply them. This seems impracticable.

The implication of these remarks is that formatted values need to be stored in the XML to support applications that will perform operations like pagination; and

canonical values must be stored to support manipulation of table data. A format that can support all applications must allow for both formatted and canonical values.

XtabML allows the exporting application to present statistic values in formatted or canonical form, or both. Certain statistics (like t-test results) are complex to represent in canonical form – some suggestions are made about this in xxx.

5.4.3. Statistic value format

The XML representation of cells will constitute the bulk of an XML table. Consider a 10 x 10 table; there will be 100 cell representations, but only 10 column and 10 row representations. Larger tables are even more dominated by cell representations; therefore great effort should go into selecting an efficient representation.

Rationale for choosing the statistic value representation

For each cell of the table we may need to store the following information for each statistic value:

- Statistic type: We have observed that typical tables use only a few statistic types.
- Formatted value of the statistic
- Canonical value of the statistic
- Status: Meaningful/not meaningful in this cell
- Visibility: Show/ hide

There are several mechanisms available in XML for the representation of these information items in the cell:

- omit the information
- convey by position in a sequence of elements
- convey as an XML element name
- express as an XML element content
- express as an XML attribute value

The following table shows the possibilities to be considered:

	Unrepresented	Implied by element position	Element name	Element value	Attribute value
Formatted value	Forbidden	n/a	n/a	**Selected**	Alternative
Canonical value		**Selected**	n/a	Alternative	Alternative
Status		n/a	**Selected**		Alternative
Visibility		n/a	**Selected**		Alternative
Statistic type	n/a	**Selected**			Alternative

Our XML design needs to choose a method to convey each item of information, i.e. one selection in each row of the table. The ‘n/a’ entries show where a mechanism is inappropriate, e.g. we cannot use an element name to represent a formatted value, or we would need as many element names as there are possible formatted value. Similarly it is not possible to represent status as implied by element position; this

would mean that we were saying something like “the third value element in a cell always has status of meaningless”.

The cells containing “selected” reflect the XML design chosen. The cells containing “alternative” are another eligible set of choices (from the 2 x 2 x 4 x 4 x 4 or 256 possibilities) that was considered and rejected, but is worth examining to put the actual selection in perspective.

5.4.4. Example showing statistics values for a table

Statistictypes definition

```
...
<statistictype name="xs:rt">
      <t>Unweighted Total</t>
</statistictype>
<statistictype name="xs:t">
      <t>Weighted Total</t>
</statistictype>
<statistictype name="xs:mean">
      <t>Average</t>
</statistictype>
...
```

This appears in the **xtab** element and defines three of the statistic types that may appear in tables within the document table: unweighted total, weighted total and average.

This example shows the row edge and statistics definitions for a table within the document. The column edge is the same as the example earlier and contains 12 columns.

Row Edge definition

```
<edge axis="r">
      <group>
            <summary><t>Total</t></summary>
            <element score=”4”><t>Read all</t></element>
            <element score=”3”><t>Read most</t></element>
            <element score=”1”><t>Read some</t></element>
            <element score=”0”><t>Read none</t></element>
            <element><t>Don't remember</t></element>
            <summary><t>Mean score</t></element>
      </group>
</edge>
```

This defines seven rows, the first being a summary used to present the ‘all respondents’ statistics, and the final being another summary used to present a mean.

Statistics definition

```
<statistic type=”xs:rt”/>
<statistic type=”xs:t”/>
<statistic type=”xs:mean”/>
```

This tells us that the data section contains values for three statistics, i.e. there will be three **c** elements within each **r** element in the order specified here.

The **statistic** elements are not qualified with **datatype** attributes; this tells us that the values we will find are formatted and not canonical – i.e. we should not attempt to interpret them as data.

Table data

```
<data>
	<r i="1">
		<c>
			<v>54</v><v>12</v><v>9</v><v>16</v>
			<v>17</v><v>12</v><v>24</v><v>16</v>
			<v>18</v><v>35</v><v>24</v><v>17</v>
		</c>
		<c><x /></c>
		<c><x /></c>
	</r>
	<r i="2">
		<c><x /></c>
		<c>
			<v>30</v><v>6</v><v>4</v><v>11</v>
			<v>9</v><v>8</v><v>13</v><v>8</v>
			<v>9</v><v>20</v><v>11</v><v>11</v>
		</c>
		<c><x /></c>
	</r>
	<r i="3">
		<c><x /></c>
		<c>
			<v>6</v><v>0</v><v>1</v><v>2</v>
			<v>3</v><v>2</v><v>3</v><v>1</v>
			<v>3</v><v>3</v><v>4</v><v>2</v>
		</c>
		<c><x /></c>
	</r>
...
	<r i="7">
		<c><x /></c>
		<c><x /></c>
		<c>
			<v>3.18</v><v>2.82</v><v>3.00</v><v>3.50</v>
			<v>3.19</v><v>3.33</v><v>3.22</v><v>3.00</v>
			<v>3.27</v><v>3.11</v><v>3.00</v><v>3.35</v>
		</c>
	</r>
</data>
```

Statistic values for the table appear in the **data** element within the table element.

In a two dimensional table the children of the **data** element are row data or **r** elements. In a three dimensional table the **data** element contains plane or **p** elements and each **p** element contains **r** elements.

There is one **r** element for each **element** or **summary** component of the edge associated with the **r** axis, in the other they appear in the **edge** definition. The r elements have optional index or **i** attributes that are purely for the convenience of a human reader of the document. This 'syntactic sugar' costs little space but is very useful in a diagnostic situation. Rows 4 through 6 are omitted in the listing.

Within the **r** element we have the data for the corresponding row. There is then one column data or **c** element for each statistic type defined in the table, in the order they appear in the **statistic** elements of the table. Within the **c** element there are one or more child elements, up to a maximum of 12 (the number of columns). These give the formatted value of the statistic for each column in the current row, as the content of the **v** element.

Some statistics are not available in some column or some rows. These are denoted by an empty **x** element, meaning both not available and hidden.

Some rows end with several values the same. Only the values up to the last distinct value are included.

The combination of these features means that rows that have no value for a statistic have **<c><x/></c>** in the statistic position. There is only one child of the **c** element, meaning that all columns after the first have the same value as the first, and that value is **<x/>,** i.e. hidden and not available.

This data section tells us that only the first statistic (unweighted total) is available in the first row, only the second statistic (weighted total) is available in rows 2 and 3, and only the third statistic (average) is available in the final row, row 7.

Hidden values

Hidden values are those accumulated by the system but not intended for printing in all cells, usually because they contribute little information except in a few contexts. For example, an unweighted total might be hidden except in a summary row.

It is of course possible to output hidden values as not available using **x** elements, but sometimes the hiding reflects a decision after accumulation; for example, suppressing rows with a small frequency, and an application processing the table might wish to reverse it. This can be achieved by using an **h** instead of an **x** element, enclosing the data value that has been hidden.

Values that are not hidden and also 'not available' are represented by the n element. This appears, for example, where Pulsar Web puts the string 'N/A' in a table cell.

The following table summarises the elements that may denote statistic values:

	Hidden	Not hidden
Available	**<h>**The value**</h>**	**<v>**The value**</v>**
Not available	**<x/>**	**<n/>**

6.Names and titles

An XtabML document describes many objects of various types:

- projects
- tables
- edges
- edge components (element, summary)
- groups of edge components
- control types
- statistic types

Objects are described by the XML elements defined for XtabML and may have names specified by a **name** attribute and titles specified by a **t** element. Both name and title are optional and should not be included unless they are meaningful.

The **t** element provides multi-lingual support for the display of the object. The **name** is there to be a language neutral identifier for the object that may be used to refer to the object in other contexts, either within the same document or in other documents.

Names are usually optional and there are two reasons why an object might be given a name:

- the name was originally chosen by a human user (e.g. age_break)
- the name is generated automatically but is related to the same name in other contexts.

For example, an edge group might be named with the name of a questionnaire variable defining the group, and **element**s within the group might be named based on the code number of the categories. This will allow rows or columns associated with those names to be combined across tables or documents.

Names are useful for combining documents, or tables within a single document. They should never be generated without a good reason, because this complicates the document with no benefit for users of the document. For example, a GUID can never be a good XtabML name.

7.Handling of text

There are three main issues arising with text handling:

- multiple natural language versions of the text
- multiple versions of the text for different applications (screen, print, abstract etc.)
- rich text: selection of text decoration, font, color etcetera

Only the multiple natural language issue was in scope for the first version of the document. However rich texts do arise in tabulation systems and some comment is necessary.

7.1.1. Multiple natural languages

The handling of natural language is designed based on the recognition that the vast majority of XtabML documents will be in one language only. This case should be simple, both to generate and to process.

Therefore the inclusion of anything to do with multiple languages has been made optional.

Almost all text in XtabML documents, and all that is not language-neutral, appears inside **t** elements. A document using only one language will contain **t** elements with text content.

Where multiple languages are to be stored in the document, the creator of the document should distinguish one of the languages as the 'base' language. This reflects the norm that documents are authored in one language and then translated into others; and that there should be a text-of-last-resort to use if translations are incomplete. Each text will be available in the base language even if not all translations to other languages have been made. The **t** elements have the base language version of the text as their text content.

The translations of the text into other languages feature as the text of **a** (for alternative) elements that are children of the **t** element. Each **a** element has a **lang** attribute specifying its language. We have not used the **xml:lang** attribute as this implies properties of inheritance that would complicate processing the documents. However the names of languages are well standardised (unlike table control or statistic types) and so we mandate the ISO language names used for **xml:lang** values as the values for the XtabML **lang** attribute also.

This approach means that a simple importing system that doesn't care about language will still work with a multi-language document; it will only use the principal language text.

When multiple languages are in use within the document, the root (**xtab**) element of the document should contain two or more **language** elements, one for each language used including the base language. These specify:

- the standard ISO code for the language (e.g. EN-GB)
- whether the language is the base language

- a label to use for the language in reports (as the text content of the language element). This enables, for example, an application to offer a menu of languages to an end-user. We haven't considered it necessary to offer multiple translations here, so this label is not in a t element.

It is also possible to include a single language element when only one language is in the document. Its function then is to identify the principal language.

In summary, this approach is intended to provide the necessary flexibility for multi-language documents while retaining the KISS principle for commonplace documents.

7.1.2. Rich text

All tabulation systems have some mechanism to express text editing for labels. The instructions to be expressed most commonly include:

- forced line-breaks
- non-breaking spaces
- preferred line-break points
- alignment
- font characteristics
- text decoration

Historically each system has developed its own method to convey these instructions, typically on the basis of 'escape' characters.

In the 21st century the world seems to have converged on HTML as a representation for rich text; in particular there is a subset used in the 'rich text' edit boxes in Internet browsers. It will probably be appropriate to adopt this subset of HTML (in its XML manifestation, i.e. XHTML) in a later edition of the XtabML standard; certainly it would not be appropriate to re-invent it. XHTML has been used in this way in the Microsoft XML spreadsheet design (see http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnexcl2k2/html/odc_xmlss.asp). Microsoft support only <b>, <i>, <s>, <font>, <u>, <sup> and within rich texts in spreadsheets.

In the initial version of XtabML there is no provision for rich text support. Instead there is a mandate to exclude all mark-up from the text exported to XtabML. In particular texts should not be exported into **t** elements as CDATA sections because these create tricky problems for consuming applications (see http://www.xml.com/pub/a/2003/08/20/embedded.html for more discussion).

Although this does not yet address the rich-text issue, it does avoid creating potential difficulties in the future (on the "constrain by omission rather than inclusion" principle).

Another reason to postpone the definition of these features is the need to distinguish between inherent properties of the text and those that are formatting choices, better conveyed by concepts of style and class. Time constraints prevented a close analysis of this issue.

8.Metadata

There are several types of ‘data about data’ to be found in cross-tabulation reports.

- Operational metadata: originating software, run time
- Project metadata: survey title, copyright notice …
- Table metadata: table title, filter title …

Metadata items in XtabML are text strings relevant to the table but not part of the table geometry, i.e. not row and column labels.

There is no consensus amongst tabulation systems that would allow us to construct a comprehensive list of metadata items. So we have taken the position that each system has its own set of table metadata item types that it will use consistently across the documents that it generates.

The design is intended to make it possible to write generic applications to process and publish tables without knowing the detailed meaning of controls.

Providing systems use **controltypes** with internal consistency it will not be a problem to adapt documents from one system to another, e.g. “job title” in one system may be “project title” in another, but it will be simple for a generic XSLT to translate documents from one naming convention to another.

8.1. Operational metadata

We settled on three key operational metadata items:

- **origin**: identification of the software creating the XtabML document
- **date**: date document was created
- **time**: time document was created

These are stored in the **xtab** element and we do not use **t** elements to represent them, because they are language-neutral, and also because they are very generic and should be easily accessible. These same elements are the operational metadata in tripleS XML.

8.2. Project and table metadata: controls and controltypes

In the root (**xtab**) element the exporting application lists the different metadata item types it will use. Searching for a less obscure and clumsy term than metadata item we chose the term control.

The various metadata item types are described within **controltype** elements. Each **controltype** has a **name**, **title** and **status**.

- The **name** is used later in the document to classify individual control texts.
- The **title** is used to label the control text in tables, e.g. “Filter”.
- The **status** (**primary/ secondary**) reflects whether the control type is a vital part of the identity of the table, or instead is merely descriptive. For example, job title is vital and primary, copyright notice is descriptive and secondary.

> Table title is primary, statistical interpretation notes are secondary. The status is intended for applications to dispose the title appropriately; for example, only primary texts would appear in a table of contents; primary texts appear at the top of the page and secondary texts appear as footnotes.

Probably each system (Pulsar Web, STAR etc) will write the same list of control types into each XtabML document that it generates.

The control texts themselves appear as **control** elements, either in the **xtab** element, if they are characteristics of the project, or in the **table** element, if they are characteristics of an individual table.

9. Canonical values

We use the term 'canonical value' to describe a value written to the XtabML document primarily for the purposes of subsequent calculation.

This implies that the value is both:

- as precise as the data justifies (as determined by the exporting program) and
- written in a standard format that is parseable by standard numeric input libraries (XSLT, C and its descendants, VB etc.).

The format for canonical values should be locale-independent, i.e. the same value will be written whatever the originating locale, and it will be read with the same interpretation whatever the input locale.

9.1. General principles

We need to consider whether XtabML needs to support both canonical values and the formatted values that would have appeared in the printed tables. It is self-evident that the formatted values alone can not provide the requirements for use in subsequent calculations. They are often written with only limited decimal places, in a locale-dependent format, and sometimes with special treatment for some values. But can having just canonical values also satisfy the requirements for formatted tables?

9.1.1. Formatting

If only canonical values are stored within XtabML, then the only way that an application can recreate the formatted values is by using the same formatting rules as the original tabulation system. This implies that the XtabML document can model these rules, and that the processing application can interpret and apply them. So what sort of formatting rules do we need:

- Specification of precision (decimal places)
- Scale factors (displayed in 000's)
- Special values (incalculable, almost zero, …)
- Conditionalisation (print negative numbers in parentheses, do not print statistics if base less than 10, …)
- Locale-specific conventions (decimal point character, use of percent character, …)

We considered including formatting directives that would handle some of these rules. But even if the most common situations are covered, then there will always be other formatting requirements that were outside the standard. Providing a full set of

formatting directives was considered too elaborate a requirement to include within a standard like XtabML.

9.1.2. Conclusion

The implication, from the above, is that XtabML should be able to store both formatted values (to support applications that perform layout operations) and canonical values (to support manipulation of table data). However there should be no requirement that both types of values are present.

In some cases the same value can serve as both the canonical and the formatted value, and XtabML should try to use this redundancy to avoid unnecessary bulk in the document.

9.2. Precision

We considered de/prescribing the precision of canonical values in the description of the statistic type, but this presents some difficulties, as follows.

9.2.1. Distinction from accuracy.

There is an important distinction between precision and accuracy.

A system may choose to write out a column percentage (say) to 2 decimal places. But if the survey has a sample size of 100 even 1 decimal place is misleadingly precise; the estimate is not even accurate to the nearest whole number percentage because of sampling error. It wastes space and is misleading to generate documents with greater precision than accuracy.

An application originating an XtabML document has (potentially) some awareness of accuracy. Subsequent processing systems do not unless accuracy is indicated in the source document in some fashion. Therefore a processing document cannot decide whether to output calculated values to one decimal place or ten without external instruction.

9.2.2. Compliance

If we provide some mechanism to indicate precision, we can anticipate a difficulty similar to that of the "standard name" feature in triple-s. There, an exporter can specify that standard names (complying to some restrictions) have been used in the document. The problem is that this requires the exporter to check whether this is actually the case, and in practice exporting software in general does not. Therefore an importing application cannot take the standard name assertion at face value and has to check each name anyway. We feel that an indication of precision will suffer from the same problem.

9.2.3. Scaling

The values of some statistics within a table may have an intrinsically large magnitude (i.e. viewing figures, company financials) and are often reported as scaled figures (e.g. 000’s, $M’s). In order not to imply excessive precision and to reduce the file size, it would be convenient if the canonical values were also stored in a similarly

scaled form. So that a viewing figure of 5.80 million would explicitly imply a precision of 10000, whilst 5800000 could only imply that the precision was better than 100000.

9.2.4. Conclusion

We decided not to include any explicit specification of precision for canonical values. The precision of a number in a table cell is implicit in its representation in the XtabML document; each number in the document will have been written with a particular number of digits and decimal places. However, it will be important for exporting programs to be aware that an application processing the canonical values may make assumptions on precision based on the values in the XtabML document. This implies that:

- An exporting application does not output more significant figures or decimal places than its internal precision warrants.

- Trailing zeros are output as decimal places if they are part of the precision (i.e. 5.4 indicates only one decimal place of precision, but 5.4000 indicates four decimal places of precision).

Large numbers may present spurious accuracy, e.g. an estimated population figure of 7 million implies more than 6 digits of accuracy but in fact the true accuracy is perhaps four digits. Therefore the canonical values for any table statistic may be output with a scale factor defined for that statistic (default 1). So that:

true value = scale factor * stored value

and in this example a scale factor of 1000 might be used and the number presented in thousands to one decimal place.

9.3. Special values

Not all statistic values are available to be rendered in the standard manner in all parts of the table 'cube', for several different reasons.

9.3.1. Not available

The statistic may not have a meaningful interpretation in some regions of the table, for example a mean where there is no associated quantity. This is a consequence of the structure of the table and is independent of the specific data in the table. Occasionally a value may be meaningful in a region but the data to calculate it do not exist (for example, unweighted counts might be accumulated only in the total row of a table). Statistic values like these that are not available in a particular part of the table, regardless of the actual data in the table, are represented in XtabML with x or n elements (hidden and visible, respectively).

9.3.2. Incalculable

These are values where the intended calculation cannot be performed because of issues in the data. Examples include percentages on a zero base, or statistical tests that cannot be applied because the sample size is too small. Often incalculable values are

shown in formatted tables as '-'.

9.3.3. Unrepresentable

This is more applicable to formatted than canonical values, and arises where the absolute value of a number is not zero but is too small to be represented with the number of decimal places used. Often unrepresentable values are shown as '*' in formatted tables

9.3.4. Zero

Sometimes zero is represented by blank or '- in formatted tables'. Like unrepresentable values, this is more an issue for formatted values.

9.4. Visibility

Often canonical values are required for calculation purposes but not included in formatted output. Sums of squares of quantities are a typical example. XtabML includes the ability to suppress values by using "h" instead of "v" elements. The motivation for this was to distinguish values that are not intended for publication because, for example, they fail to meet sampling criteria, or create unnecessary bulk in printed output, like blank rows in sparse tables. 'Hiding' values creates the possibility for consuming applications to make their own interpretation of the hiding rules.

The requirement to suppress canonical values tends to be all-or-nothing for each table rather than a cell-by-cell consideration, so we use a special indicator on statistic declarations to distinguish canonical values.

9.5. Data type

The 'data type' of a statistic is the set of possible values it may have (as opposed to its interpretation: for example, all percentages have the same data type). There are several data types that commonly arise in cross-tabulations:

9.5.1. Integer

A statistic such as an unweighted frequency that has a whole number value.

9.5.2. Decimal

A statistic such as a weighted total that has (potentially) a fractional value. Decimal values are used rather than floating point because floating point notation is never used in survey reports, and it is more bulky for storing values of typical magnitude.

9.5.3. Percentage

Percentages arise very frequently in survey reports. Applications such as XSLT and Excel, important candidates for use with XtabML, regard percentages as fractions and multiply by 100 for display, so we adopt the same policy for XtabML. The canonical value of a percentage is stored as a fraction, while its formatted counterpart might be a whole number suffixed with a percent sign.

9.5.4. String

String data types are uncommon in tables. They are most commonly used as the formatted representation of a probability or a set of probabilities (e.g. comparing columns in T-tests).

9.5.5. Others

We also considered other possible data types, including:

- Probabilities, which are a ratio constrained to the range 0 to 1.
- List, which is an ordered set of simple data types.

9.6. Representation of canonical values

We need to extend the existing XtabML specification to allow for representing one or both of formatted and canonical values within the table. There would seem to be a number of possible approaches:

- Use different 'statistic types' for the canonical and formatted values
- Use multiple instances of a 'statistic type' to represent the two values.
- Make one value an attribute of the element defining the other value, or even make both values attributes.

9.6.1. Different 'statistic types'

Pros: Retains the existing XML structure.

Cons: Would need three statistic types for each statistic (i.e. formatted, canonical, and both). Would only be able to determine which are the formatted values by inspecting the names of the statistic types.

9.6.2. Multiple instances of a 'statistic type'

Pros: Would allow the possible statistic types to be defined at the higher level (cf control types). The statistic instance definitions could provide additional attributes describing that statistic in the context of the particular table – in particular whether it is canonical, formatted or both.

Cons: Adds more statistics to the data section of the table, although only where canonical and formatted values both exist.

9.6.3. Attribute of the value element

Pros: Would not increase the number of value elements.

Cons: Visibility of canonical and formatted values is poor because their presence can

only be deduced by examining the value elements. Different status of the two values unless both are made attributes. Makes the table data part of the XML more complex.

9.6.4. Conclusion

We considered all the above, plus some combinations that took features from more than one approach. The conclusion was that we use the multiple instances of a ‘statistic type’ as the way to allow for canonical and formatted values.

10. Standard statistics

10.1. Rationale

XtabML takes the view that less is more; the greater the number of statistic types the greater the burden on implementors. Statistic types that do not have the prospect of a universally practiced calculation method have been excluded from standardisation. For example, 'sum of squares' is included; 'weighted standard error' is not. As a further, more subjective judgement, statistics that have no prospect of widespread use have been excluded from the list. Applications are free to define their own statistic types where no appropriate statistic type exists in the standard..

10.2. General principles

10.2.1. Optional

Exporters don't have to use the standard types or the standard namespace. But if they do, it must be with the defined meaning.

10.2.2. Inter-calculable

There are defined arithmetic relationships between standard statistic types. Therefore applications may calculate e.g. column percentages if only totals have been supplied; or omit derived statistics when outputting a transformed table without destroying information. At this time we do not attempt to describe these relationships in the document itself, as this is a very complex undertaking (broadly equivalent to the multi-dimensional OLAP extensions to SQL). However the relationships between standard types are documented as part of the standard type description so applications are free to exploit them as they see fit, for example by deriving standard values that were not included explicitly in the XtabML document from those that were included.

The documented relationships between the values make it possible to construct an XSLT template for each standard statistic that will use the stored value in the document, if available, or else if possible construct the value from those that are available in the document.

10.2.3. Title

Applications declare a title for the standard value types just like for private ones, with optional language alternatives. There are suggested titles but applications are not mandated to use them (it is unlikely that application developers will be willing to depart from the statistic titles that their users are accustomed to). The definitive link between statistic type declarations in different documents is the type name, and not its title.

10.2.4. Row/column percentages?

Arguably the representation of tables should be as an abstract hypercube whose dimensions are the table edges, and this is independent of how this is expressed in reports as pages, rows and columns. In this way of thinking 'row' and 'column's lose their meaning and so do the corresponding percentages; standard statistic types would

need to be expressed in terms of dimensions rather then presentation attributes like row and column. In XtabML, since different pragmatic arguments have dictated a nested representation for the data section, it is convenient to regard the most deeply nested dimension as the 'column's, the next most nested as the 'rows', and the next as 'pages', and interpret 'row percent' etcetera as standard statistics on this basis.

10.2.5. Projection/universe

Weighted tables, especially in continuous surveys, are often projected to the total of the population (or universe) being sampled. In table processing, both the projected and unprojected totals may be required; projected totals for publication, and unprojected totals for assessment of sampling error. Since these are related by a constant multiplication factor, there is potential redundancy in including both numbers in all cells. No accurate calculation is possible on formatted values, so in general, if both projected and unprojected weighted formatted totals are required, both must be output.

10.2.6. 'Publication' totals

Some surveys are weighted, others not. In each case a typical presentation of tables will have totals and column percents in the data rows. If weighted total statistics only appear in weighed tables, then consuming applications will need two variants, one for weighted tables, and another for unweighted. This is unfortunate when in common practice the only difference in processing between weighted and unweighted surveys is that weighted surveys include an unweighted total line.

Hence the possibility of a 'publication total' whose value is the unweighted total in unweighted tables and the weighted total in weighted tables, and associated 'publication' row and column percentages. Similarly, where a table has an associated quantity that is being analysed for summary statistics such as mean, that table may be weighted or unweighted, and again the use of weighting does not greatly influence the style of presentation. It will be inconvenient if processing code has to deal with unweighted and weighted cases separately. Pragmatically, if weighting is present then unweighted percentages and means are rarely used. If weighting is absent, they are all that can be used.

In XtabML the standard statistics defined are what are normally used for publication, and include or exclude weighting appropriately in each case. This is to avoid creating a large number of statistic types that serve only to complicate the processing of tables in the overwhelming majority of cases.

10.2.7. Cross-tabulation repertoire

Cross tabulations although similar to OLAP cubes are typically rather simpler. An OLAP cube may contain many measures in each cell such as sales, units, gross margin, net margin etcetera. Cross tabulations regard rows, columns and planes as each having a qualification or filter constraining the data that are summarised (and tables as a whole may have further constraints or filters).

So each cell of a cross-tabulation is associated with a qualification. The most basic statistic is therefore the unweighted total, or count of qualifying respondents in an elementary edge component (represented in XtabML by an 'element'). Each cell may

also be associated with a weight value, and one or more quantity variables (the distinction between weights and quantities being explained above).

Complex survey tables may have many different weight and quantity variables operating in different regions of the table. However, this is a matter for the table accumulation software and in XtabML we are modeling only the result of the tabulation process. It is important to know that a weight operates in the context of a cell, but not to know the details of its definition; just as we know that there are qualifications for each row and each column but we don't need to know, at the XtabML stage, what they are in terms of the data set that was tabulated.

10.3. Guiding principles

10.3.1. Important distinctions

Data type

Each standard value type has an XtabML data type i.e. integer, string, or decimal. The data type determines the set of possible values for data of that type (e.g. non-negative integers). The statistic type, on the other hand, determines the interpretation of the values.

Summary elements

Within a table, some rows and columns contain summaries of other rows and columns, most conspicuously "total" rows and columns. It is important to note that these rows and columns differ from others only in the qualification for respondents to appear in them. There is a meaningful unweighted total, weighted total, column percent etcetera in total rows just as in 'regular' rows. So "total" for edge components is a different concept from "total" as a statistic type.

Weights and quantities

We distinguish between weights and quantities.

Weights are numbers calculated for each respondent in a survey to correct for sampling bias. When weighting is used, it is normally used in all tables, and normally applies to the whole of a table.

Quantities are numeric attributes of the respondents that we wish to analyse to establish total volumes (e.g. pints of beer drunk), averages etcetera. Quantities feature in only some tables, and often only in some parts of those tables.

Frequently quantities used in tables are implicit; if the answers to a question (typically expressed as rows) have an associated score value, say from 1 to 5, then there is an implied quantity for each respondent in the range 1 to 5 and summaries of this quantity may be included in one or more rows of the table.

10.3.2. Rigor

The objective of standardising types is that numbers of the same type produced by different systems in different tables may be combined. Tables produced by the same application may be combined in any case because XtabML requires that each application use statistic names consistently across all the XtabML documents it generates.

If the motive for combining numbers is anything except republishing formatted values, then this implies that the values for the same type from different systems will be compared and combined arithmetically. This can only be meaningful if the values have been derived on the same basis; for example, two systems might indicate a significant value with a single asterisk, and use the same standard name for the statistic type. If however, one system used a 5% probability level as its criterion, and the other used 2%, then combining these numbers in one table would be very misleading. Arguably this can be solved by having a parameter for the value type indicating the significance. One system would specify significance(5) and the other significance(2). It is not clear how this provides any usable benefit for consuming applications, and it is certainly more complex.

The problem is compounded when the parameters of the value type need to be qualitative rather than quantitative: for example, what is the meaning of standard error in a weighted table? How many different algorithms are used by different applications to calculate this statistic? We cannot mandate a specific one (developers will not output one number called standard error in XtabML documents and a different number called standard error in their regular tables, and of course they will not suddenly change the way they compute standard errors either). If we attempt to parameterise the value type to express the diversity of calculation methods then consuming applications are presented with a very complicated task to make any useful comparison of value types between documents; and when they find that two value types have the same name but different parameters, what use is that information to them?

The conclusion we reach is that only statistic types that have a high probability of a universally consistent derivation should be included in the standard. Our assertion is that these will be just the 'basic' statistics that can only be calculated from the disaggregated data, e.g. unweighted and weighted totals, and very simple statistics derived from those like percentages and means. it is even possible that percentages will be subject to controversy in their definitions - what is the value of a percentage with a zero base? The advantage of encouraging producers to output the basic statistics is that consumers can then derive all the statistics they want using the calculation they want, whatever the normal behavior of the originating applications in producing these statistics.

10.3.3. Rule of thumb

To determine whether a statistic is a suitable candidate for standardisation, consider a survey data set analysed with several different tabulation systems to produce a set of tables containing a diversity of statistics. Those statistics that we can be confident will have the same values in all the reports should be standardised; those that will vary due to different algorithms should not.

10.4. Proposed solution

Standard statistic types fall into two classes, basic statistics that are accumulated from the individual data, and derived statistics that are calculated after accumulation. The standard attempts to include statistics that are commonplace in survey tabulations, and have hopefully an uncontroversial definition. Many other statistics are occasionally used but have not been included in order to keep the size of the list small for the first version of the standard, while we determine whether there is truly a useful consensus on the definition of standard statistics.

10.4.1. Basic statistics

These are the statistic types created by accumulation from the raw data and are hopefully not subject to divergent interpretation; as in the 'rule of thumb'.

Qualifications

The meaning of a cross-tabulation is that each row and column, and each plane, or hyper-plane in a higher-dimensional table, has an associated qualification (or filter) that restricts the respondents that are eligible for inclusion in the table. Additionally, there may be qualifications applicable to the table as a whole. So for a respondent to count in a cell s/he must meet all the qualifications applying to the cell. There is no requirement that qualifications must be exclusive, indeed frequently there will be more than one row with the same qualification: for example an "all answering" row above a distribution and a summary statistics row below it showing dispersion statistics such as mean and standard deviation, both with the same respondents qualifying.

Unweighted total

The unweighted total is simply the count of all respondents qualifying for the cell, i.e. meeting the qualification for each edge.

Weights

When a weight is applied to a table, each cell now potentially has a weighted total as well as an unweighted total. Weights are usually fractional, so although the unweighted total statistic is an integer, weighted totals are decimal. Note that adding weighting to a cell adds a further qualification to the cell, i.e. the cell will include only respondents who have a weight value. If the weight value for a respondent cannot be determined, then nothing is added to the weighted total nor is the respondent added to the unweighted total for the table.

- **Weighted total**. The weighted total is the sum of weights for all respondents qualifying for the cell.

- **Sum of squared weights**. The sum of squared weights is required to calculate effective sample size.

- **Population**. Sometimes weights in the disaggregated data include a factor to

make their sum equal to the population from which the respondents are drawn. This process is called 'projection', the factor is the 'projection factor', and the population total is the 'universe'. When projection is used, each weighted total is an estimate of the population of respondents in each cell of the table. Very often the population is known for some cells from census data, particularly for totals of demographic qualifications, and some reporting applications will include the population as a statistic value so readers of the report can check the performance of weighting by comparing the estimated population found in the sum of weights with the actual population. So 'population' is a statistic type in itself.

10.4.2. Quantities

Very often a table contains an analysis of some numeric (quantitative) property of respondents. The numeric property is analysed in some or all cells of the table to generate statistics such as mean, median and standard error. As with weights, adding a quantity to a cell has the effect of extending the qualfication for the cell as a whole so that only respondents who have a value for the quantity are included in the unweighted total of the cell (or the weighted total if weights are present). This is necessary because otherwise statistics such as the mean will be calculated on an incorrect base.

- **Quantity total**. The most fundamental quantity statistic is the quantity total

- **Sum of squared quantity**. This statistic is required to obtain the full repertoire of derived dispersion statistics.

- **Exotic quantity scenarios**. There are some less common situations where more than one quantity may be associated with a cell. XtabML does not attempt to model these with standard statistics in the first version.
    - **Cross-products**. Sometimes there may be a quantity variable associated with each row and each column of a table, i.e. a row and a column quantity associated with each cell. The most common example is for dependent sample T-tests comparing two means. In this scenario as well as the row quantity sum and sum of squares we need the column quantity sum and sum of squares, and the sum of products of the two quantity variables. The cell qualification is extended to include only those respondents who have values for both quantities. Another application of cross-products is the generation of correlation coefficients, that are the basic inputs to factor analysis.
    - **Units and Price**. In conducting a retail audit, for example, for each product in a store the survey may capture a stock level, in units, and a price. Statistics required in cells corresponding to products may include average unit price, calculated as the sum of units times price divided by the sum of units. This scenario is comparatively rare and we have not modelled this and other specialised statistic types in XtabML standard statistics for now.
    - **Media evaluations**. When reporting the results of media surveys, there are two key aggregations for individual media used to derive other reported statistics: reach (or cume) and exposure (total listening,

GRPs). Derived statistics include share, reach %, average exposure/hours listening and others.

10.4.3. Derived statistics

These are statistic types where there may be a diversity of interpretation (such as weighted standard error). Here the standard provides a definition of these statistics in terms of the basic statistics. These definitions are intended to reflect industry 'best practice'. XtabML producer applications, if they provide these statistics, should calculate them according to the standard. This means that any document including basic statistics and standard derived statistics can have the derived statistics replaced by a fresh calculation according to the standard in the definition and their values should not change (apart from issues of precision). If a producer application uses a non-standard algorithm to calculate a statistic, the exporting module should output the applications' own evaluation of the statistic with a non-standard name. It may of course also output the standard evaluation of the statistic as a separate statistic type using the standard name. Note that although exporting applications are obliged to use the standard definition for those statistic values that they include, they are not obliged to include the values throughout the table. For example, effective sample size requires the sum of squared weights. If the exporting application only accumulates the sum of squared weights for part of the table, it need only export the effective sample size statistic to the cells in that region of the table, leaving the statistic value as ‘not available’ elsewhere..

Percentages

Percentages are by far the most common derived statistics, and column or vertical percentages are the most common among percentages. This terminology implies that the data in the table are organised into rows, columns (and planes etcetera).

XtabML documents have explicit row, column and plane dimensions, so it is clear that a row percentage for (say) weighted totals in a cell is calculated with respect to a base row in the table. However, this begs the question, "What is the base row?"

Presentation of bases varies widely; in most markets bases appear in the topmost row, but in some (Sweden for example) they are conventionally the final row. Occasionally, totals are not printed at all, although this is widely regarded as bad practice. When bases are available, in principle percentages can be calculated from the canonical values of totals in the current row/column and the base row/column.

XtabML provides a method to identify the base elements within edges, using standard summary types.

10.4.4. Standard summary types

XtabML has an optional **type** attribute for **summary** components of edges. Edge components are organised into a hierarchy of **group** XML elements. Groups may contain one or more **summary** components, in any position. If a **summary** has a **type** attribute with the value '**xs:base**', this implies that this summary contains the base totals for that group. Note that it is not a requirement of XtabML that summary components exist. However, if they do not exist percentages cannot be recalculated

from canonical values of totals, because it is not possible to identify the base values. There may be only one base summary in each group. The base summary of the outermost group, if it exists, is the overall base for the edge. XtabML has standard statistics for both overall and local percentages, where local percentages for an edge component (row/column) are based on the base summary for the current group. Local percentages allow XtabML to represent a common format where there is an "all respondents" row (the base summary of the outermost group), and an "all answering" row (where there is an enclosed group associated with response to a particular question). The 'all answering' row has a percentage based on the overall base, while the individual answer rows have percentages based on the 'all answering' row.

10.4.5. Dispersion statistics

The formulae used in XtabML for variance etc. are for the 'sample' variation of the calculations, not the 'population' variation. I.e. the divisor for variance is the sample size minus one, rather than the population count.

10.4.6. 'plane', 'row' and 'column' statistics

Where there are more than two dimensions there comes a combinatorial explosion of possibilities to select a percentage base.

For example, in a 3D table cells are distinguished by three separate qualifications for rows, columns and planes. Percentage bases can be constructed by removing any combination of these qualifications. Row percentages are constructed by removing the column qualification only, and column percentages by removing the row qualification.

By extension of this principle, plane percentages would be constructed by removing the plane qualification. Taking an example, if rows represent age categories, columns gender and planes regions, then a plane percentage in a particular age/sex/region would be calculated from the respondents in that age/sex/region based on all respondents of that age/sex independent of region. However, in practice, 'table' or 'global' percentages in a 2D table are based on all respondents, i.e. the base is found by removing both the row and the column qualifications.

In the 3D situation, there are six ways you can form percentage bases because there are six ways one or more qualifications may be removed from a cell. In a 4D table, there are 14 ways and so on.

XtabML supports only three of these possibilities in the repertoire of standard statistics, the three that in practice constitute the overwhelming majority of published percentages.

- **Row percentages**. Row percentages for a cell are based on a cell with the column qualification (only) removed.

- **Column percentages**. Column percentages for a cell are based on a similar cell with the row qualification (only) removed.

- **Plane percentages**. Plane percentages for a cell are based on a similar cell

with both the row and column qualifications removed.

10.4.7. Indices, moving averages etc.

Sophisticated tables often contain statistics that are ratios calculated between derived statistics.

For example, the 'index' for a cell is calculated as the ratio of the row percentage in the cell row to the row percentage in the total row. Other indices are often calculated, e.g. year-on-year change where years might be represented as planes, adjoining column groups, adjoining columns etcetera. Moving averages are used to smooth variation over successive reporting periods, and again these may be reflected in rows or columns, and may be taken over any number of periods.

There are too many variations in these statistics to be reflected in a set of standard statistic types. These would be more appropriately represented by some 'formula' based system, but this is out-of-scope for XtabML that does not attempt to address all the complexities that created the genre of OLAP systems.

10.5. Score values

Where statistics are calculated for a table by applying a score to selected rows (or columns), it is appropriate to model this in the XtabML. Therefore **element** XML elements in edges may have a **score** attribute with a decimal value to record the score associated with a row. **summary** elements may not be scored. Edge elements without a score are excluded from statistical calculations. Typically scored elements will be grouped, and the **group** XML element will include a **summary** element where the statistics that have been based on the scores are presented. This is not mandated; scores may be documented in the XtabML without exporting any statistics based on those scores. Similarly, the derived statistics may be presented without the use of score attributes to record the scores used.

If the score values are included then it is possible in principle for consuming applications to calculate the scored sums and sums of squares required for dispersion statistics without these statistics being explicitly included in the document; or to calculate their own values of the statistics based on a different choice of scores

11. Results of significance texts

This section shows how the results of significance tests may be stored in XtabML.

Certain statistical tests yield results that are probabilities. Although these can be presented as percentages, it is conventional to mark a result as ‘significant’ if the probability is below a certain threshold.

A common scenario is that some quantity, such as a rating, is analysed by various demographics. Tests are performed that compare every pair of columns; so that, for example, if there are 11 columns there will be 11 x 10 tests. Each test yields one of three results:

- no significant difference in the quantity between the columns
- the quantity is significantly greater in the first column than the second
- the quantity is significantly greater in the second column than the first.

Usually the great majority of tests yield no significant difference.

There is a widespread convention for the presentation of these results. Each column is assigned a letter from the alphabet sequentially, i.e. first column is A, second is B etcetera.

Each column of the row for the statistic shows the results of testing whether the mean in that column is significantly greater than that in each other column. It shows the results by listing the letters denoting each of the columns with a significantly smaller value of the quantity. So, if the third column (denoted C) has a significantly greater value than the first and fourth columns, the test result is shown as ‘AD’.

In XtabML this can be simply represented by two statistictypes:

```
<statistictype name=”statlabel”>Label</statistictype>
<statistictype name=”Ttestresults”>T-test</statistictype>
```

The statlabel statistic is hidden and inapplicable everywhere except in (say) the total row, where is has the values A, B, C etcetera in successive columns. The data compression convention in XtabML means this carries only a small overhead.

The Ttestresults statistics are visible and applicable in each row and contain the test results as a string such as AD in each column.

There is a simple extension of this concept for two different significance levels, one associated with capital letters and the other with lower case.

11.1. Detailed significance results

But what if we are interested in the 110 probability values associated with each row, and not just in whether they are significant?

In order to calculate the probabilities in general it is necessary to make a cross-tabulation of each column by each other column for every row of the table. There is a special situation where the columns are mutually exclusive (S variable) and these tables reduce to a diagonal, but we need a solution for the general case.

The solution is to represent the table in XtabML as a 3D table, where a copy of the column edge is used for planes as well as columns. This provides a cell for each combination of columns in each row, where the probability and any other useful statistics such as the sum of squares may be stored.

This may be combined with the summary reporting with column labels. The values in the extra planes may be stored as hidden so the default presentation of the table would show only the summarized values, but the detailed probabilities are still available in the document for usages that require them.

12. Deferred and out-of-scope issues

The requirements excluded treatment of several major features from XtabML. Since the standard is expected to evolve over time, we have made some attempt to anticipate extensions to support these features.

12.1. Canonical values

12.1.1. Requirements

A canonical value is intended for processing table data values as numbers. A standard that includes canonical values should address the following issues:

- Specification of data type
- Specification of precision (avoiding wasting space and processing time with excessive precision).
- Minimise redundancy when both canonical and formatted values are present.
- Provide for special values for e.g. percentages where the base is zero

12.1.2. Recommendations

statistictype elements require some extra attributes to support canonical values:

- **datatype**: probably the integer, decimal and float types may be adopted from the XML schema standard (http://www.w3.org/TR/xmlschema-2/ section 3.2.3-5), as well as text types. This standard has provision for undefined or 'Nan' values, though only for floating-point types.
- **iscanonical**: 'y' or 'n'. If y, then this might imply that the formatted value is also the canonical value. For example, if the formatted value is an integer represented without grouping characters then the canonical and formatted values would be the same so the canonical attribute may be omitted.
- **places**: for non-integer data types, records the number of decimal places used. This is necessary, for example, in an application that adds two reports together and outputs the result. This may necessitate recalculating some percentages. The application needs some indication of the accuracy of the input data to use the appropriate accuracy in the output.

Canonical values may be stored in **h** or **v** elements as the value of a new **v** attribute. **n** and **x** elements do not require a separate canonical value attribute.

12.2. Rich text

One approach to rich text support may be to allow the contents of **t** and **a** elements to contain elements from the XHTML namespace (http://www.w3.org/1999/xhtml), e.g.

```
<t>This XtabML text contains some<xhtml:br/>marked-up text;
   <xhtml:b>This is bold</xhtml:b>
</t>
```

This will obviously be very convenient if text is to be subsequently rendered by a system that understands XHTML markup, as Internet Explorer and Microsoft Word do. Conversely it will be a difficult and unrewarding exercise to design some new competitor to XHTML mark-up.

There are some subtleties however:

- Where text is coming from a legacy tabulation system, the system-specific mark-up will have to be translated to XHTML equivalents. Of course this translation will have to happen somewhere in any case if these tables are to be rendered on a modern system.
- It may seem simpler if the texts are already expressed in HTML format, for example mark-up in answer labels from FIRM scripts. However, the onward system must protect itself from HTML injection attacks, so the HTML code will have to be censored in some way to generate only ‘harmless’ tags without script. HTML is also not necessarily valid XML, so some utility like HTML ‘tidy’ will be needed to prepare the text for inclusion in XtabML.

12.3. Styles

In a publication quality table all texts with a particular function will usually be rendered in a consistent style, in the sense of font-style, size and decoration such as underlining, forcing to upper case etcetera.

The modern approach to dealing with this situation is to classify texts according to function and then associate a specific appearance with each class at the time the text is rendered.

Considering an HTML document for example, any HTML element may have a class attribute associating a class name with that element. A separate CSS style sheet associates specific CSS instructions with each class name.

An XSLT script processing an XtabML document and generating HTML can very easily insert a class name into the elements corresponding to each context. An external styles sheet would define a CSS class for each **statistictype**, for each **controltype**, for row and column **group** headings etcetera.

There may be contexts where the required style is known to the application creating the XtabML document. For example, Pulsar Web has a style editor allowing users to define CSS styles for features of tables online. It may well be useful to allow these definitions to be included in the XtabML document.