Data Point Model database

August 2013

Introduction

When developing, using the Data Point Modelling methodology[1], the draft Data Point Model (DPM) for the draft Implementing Technical Standards (ITS), put forward in consultation papers CP 50 and CP 51, the EBA decided to further enhance the implementation of the methodological approach, by introducing a relational database as the repository for the DPM metadata, instead of relying solely on MS Excel data structures. For convenience reasons, MS Access was chosen to support this database.

One of the main advantages of this technical component is to impose a series of logical constraints on the model, and enabling the realisation of a series of automatic consistency checks that would not be possible otherwise, thus contributing decisively to shorten the time needed to achieve the desired level of quality, on a DPM that categorises over 30,000 data points.

Another considerable benefit from the database is the possibility of defining many different views on the same metadata content, according to the needs of the user who is trying to understand the reporting framework, and the link between the business templates and the dimensional data points, which are now explicitly defined in the DPM.

The database model is a meta-model, in order to be used in any reporting domain other than COREP/FINREP, with a relatively low level of abstraction, focusing directly on the main concepts that are used in data point modelling (e.g. *framework, table, table cell, dimension, member, domain …)*. As regards the dimensional concepts, they basically share the same definitions found in analytical systems, which makes possible a very straightforward connection between both ends of the reporting chain.

Compared to the introductory version of the meta-model expressed in the database, released in May 2012, this release of the model has been enriched and refined, expressing several additional concepts, and making clearer the link between the pure dimensional analysis view of the tables and data points, and a likely expression of that model in e.g. data transmission or analysis systems.

[1] See e.g. http://archive.xbrl.org/24th/sites/24thconference.xbrl.org/files/TSMT9AndreasWeller.pdf , http://www.eurofiling.info/finrepTaxonomy/DPM-Formal-Model.pdf , http://www.eurofiling.info/dpm/

The meta-model is however not bound to any particular technology, and therefore XBRL specific constraints, for example, are not reflected in the DPM if they would reduce the clarity of the model. In order to streamline the process of automatic translation from the DPM to XBRL taxonomies, however, some additional model elements have been added, and a layer of XBRL properties (e.g. *namespaces*) will likely be added to the database in future.

Other enhancements will follow to address additional issues, such as versioning of metadata. Both the templates and the data points' categorisation are expected to change in the future, and keeping track of history of the unique data points is a fundamental requirement for data warehousing and time series analysis.

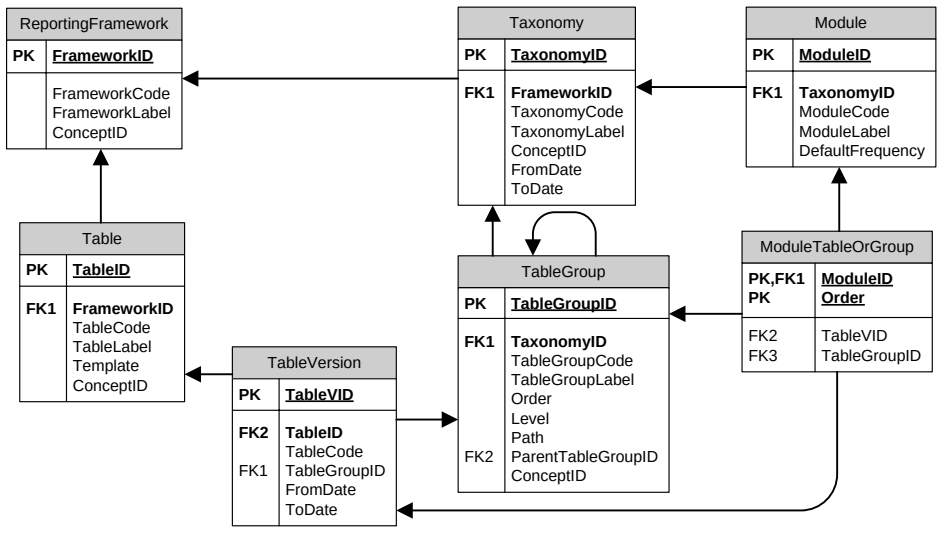
Structure of the database

The meta-model is basically structured around the representation of the templates' metadata, the dimensional concepts used to categorise the data, and the links between them, which is the actual categorisation.

Tables and table groupings

Tables belong to a Reporting Framework (currently either COREP or FINREP); most of the time the concept of *table* will be the same as business *template*, except when, for modelling reasons, a template had to be normalised and split into two or more tables (e.g. C 09.01 (CR GB 1) becoming C 09.01.a and C 09.01.b).

(In the following diagrams, the arrows represent relationships that must be read as "belongs to a" - i.e. indicating a *many-to-one* relationship, pointing from the many to the one).



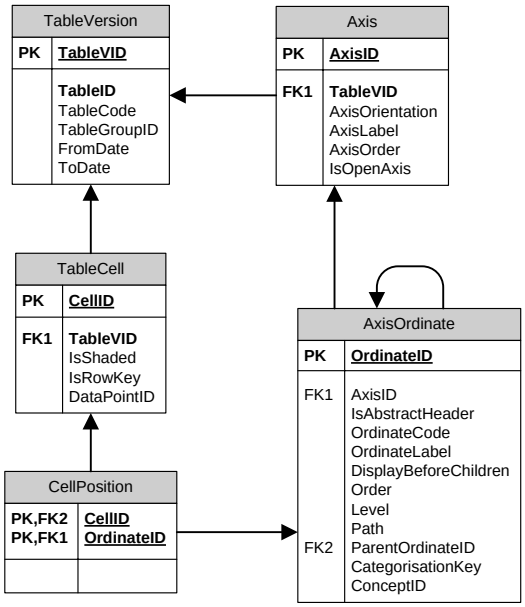
The reporting framework and the tables within them are relatively stable concepts, that may persist over several variations and updates to the specific practical reporting requirements or technical implementations.

In contrast a specific description of the classification of these tables and the data points within them at a particular point/period in time is referred to in the model as a Taxonomy, with the specific description of a particular table within a taxonomy being represented by a TableVersion, several of which may represent the evolution of a particular conceptual Table over time.

Within a Taxonomy, TableVersions may be grouped, for information, into TableGroups. Those tables representing parts of a template are grouped into a table group representing the overall template. Tables/templates are grouped together into subject areas (e.g. Capital Adequacy, Credit Risk etc.)

Modules represent the major units of reporting, collecting together a set of tables that could potentially be reported in a single submission. The tables included in a particular module are indicated by the ModuleTableOrGroup relationships, which either links a specific TableVersion, or a TableGroup (and any contained TableGroups or TableVersions) to module.

Table layout



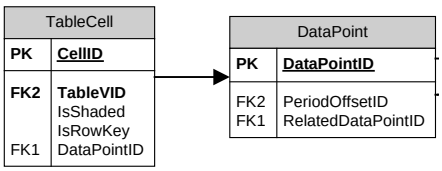
The physical layout of a table (TableVersion) is described in terms of axes. An Axis represents either the rows, columns, or sheets of a table, which are "X","Y" and "Z" axes respectively (the possible values of AxisOrientation). Each possible value along each axis (i.e. the individual row, column, or sheet) is called an AxisOrdinate.

This decomposition of tables is key to the modelling process, which categorises each individual value on an axis (5,903 in total), instead of each individual cell (around 82,000 table cells).

There are in the framework several different kinds of tables. Most have a fixed structure, with one single sheet, while others can have multiple sheets with the same structure (e.g. C 08.01 (CR IRB 1)), or even a variable number of sheets (e.g. C 09.01 (CR GB 1)). Also, some tables have a 'list' format, that is, an open structure where rows are identified by typed key data, and repeating an indeterminate number of times, depending on the data being reported (e.g. C 14.00 (CR SEC Details))[2].

The TableCells are generated in the database, by crossing the ordinates of the axes of each table. Around 50%[3] of all table cells are grey-shaded, either because data is not being required, or because the *row x column* combination has no logical meaning.

Each table cell (not considering the grey-shaded) corresponds to one, and only one, informational fact, called a DataPoint; however, there are some data points represented in multiple table cells. In the latter case the table cells contain exactly the same piece of information, and so share exactly the same categorisation in the DPM.



[2] This is indicated by the axis property IsOpenAxis

[3] Although nearly 30% of notional table cells are trivially "grey-shaded" since either their row or column is what is called an abstract header, i.e. just a descriptive/informational part of the table layout, not representing any intended data entry, and that entire row/column is expected to be either not shown or shown as grey.

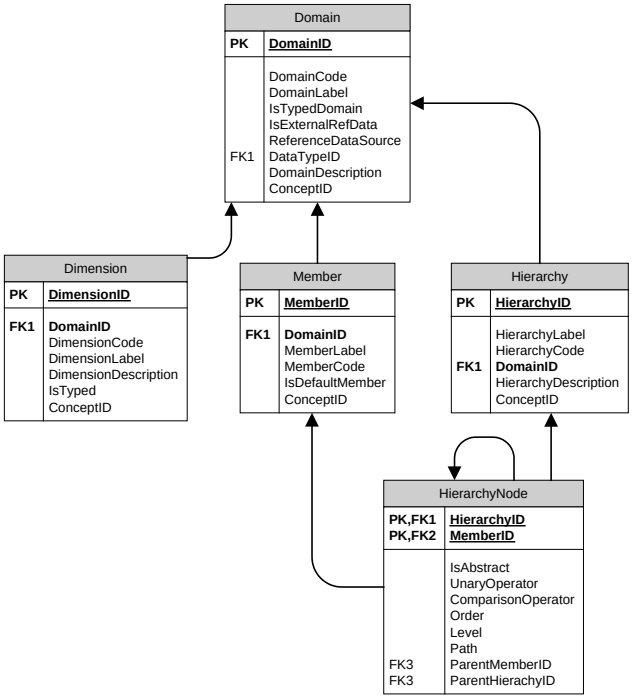
Dimensional Data Model

The dimensional concepts represented are Domains, Dimensions, Members, and Hierarchies.

Dimensions are the different categories used to describe the data points (e.g. *Counterparty sector*), and Members are the actual instances of those categories (e.g. *Central banks*).

For instance, the cell in FINREP table 8.01.a, row 090, column 010, is categorised in the DPM by the following five pairs of *[dimension].[member]:*

[Base].[Liabilities]
[Metric].[Carrying amount]
[Main category].[Deposits. Redeemable at notice]
[Accounting portfolio].[Financial liabilities held for trading]
[Counterparty sector].[Central banks]



All members of a dimension must belong to the same Domain. A domain groups members of the same type, corresponding to concepts with similar semantic nature, either abstract like *Type of risk*, or more concrete like *Currency*. Some (most) domains are "closed", i.e. have a predefined and restricted number of members (e.g. *Countries*), and others are "open", since we cannot enumerate all possible instances (e.g. *Legal Entities*).

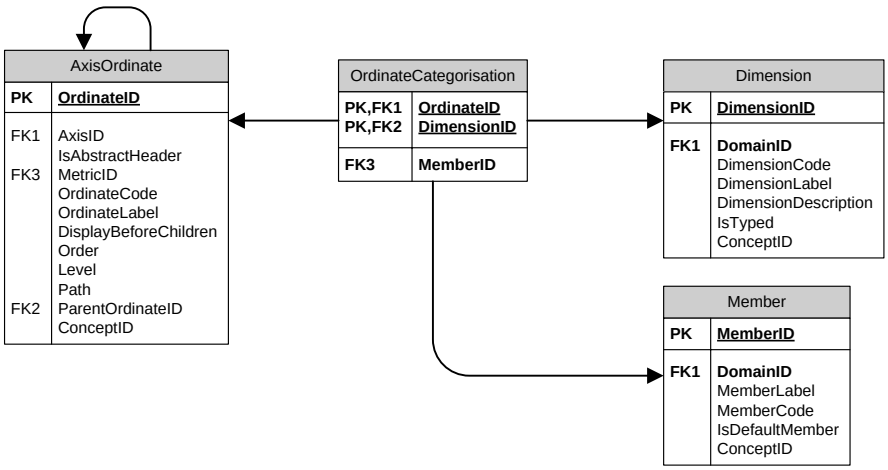
Dimensions are not always simply direct one to one equivalents of domains, because they may represent a particular *role* in the model. For instance *Residence of the counterparty, Location of the activities*, and *Country of the market*, are all different dimensions that take their members from the *Geographical area* domain. Thus the same member can belong to different dimensions, and two different dimensions from the same domain can categorise the same data point[4].

Hierarchies indicate how members relate to each other, define subsets of members, and can also define the aggregations from lower to upper levels in the hierarchy.

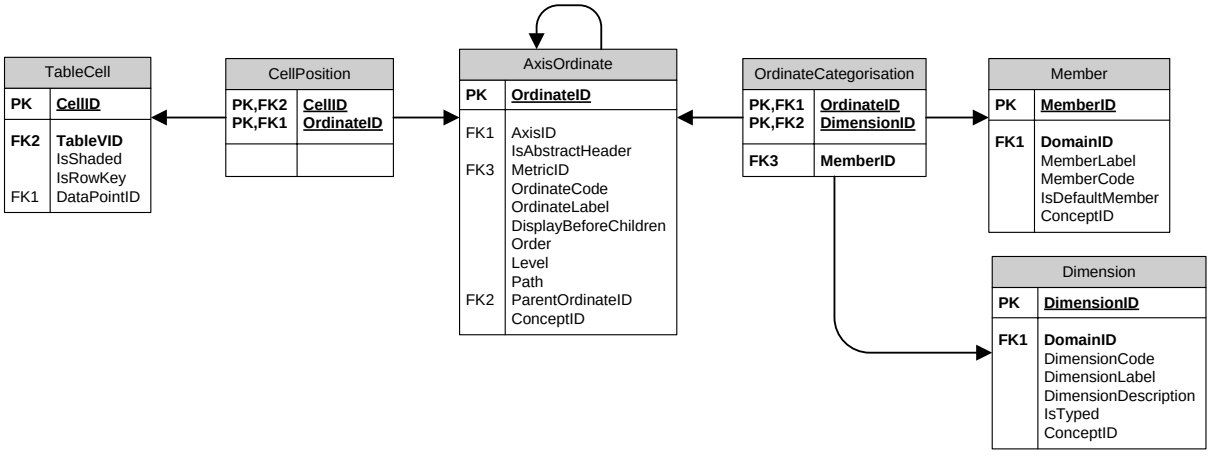
[4] But of course, the same dimension cannot be used to categorise a data point twice.

Dimensional Analysis of templates

When describing the templates, business experts define the set of (pairs of) dimensions and members that categorise each row and column. If there is a 'z-axis', multiple sheets are generated, and the dimension and members associated with each sheet as a whole must also be specified.



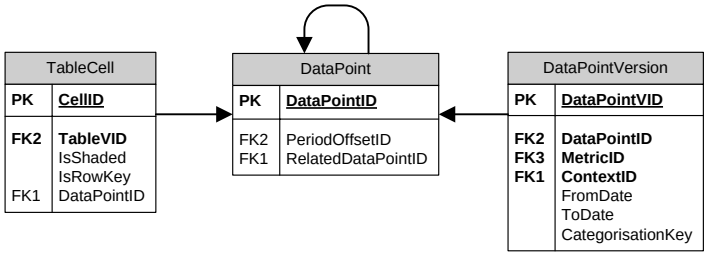
It is possible to trace from a table cell, through the axis ordinates of the cell to identify the complete categorisation of each individual table cell, resulting from the union of the categorisation of its axis ordinates (sheet, row, and column).



Data point modelling is an iterative process. In each iteration a set of consistency checks is applied to all the cells of the framework, to validate the model from a logical perspective, checking for cases of missing mandatory dimensions[5], or duplicate dimensions, for instance.

[5] All table cells that are expected to contain reported data must have as a logical minimum a "metric" dimension value, indicating the fundamental nature of the value being reported. In this data point model, it is also expected that most data points are also categorised in terms of the "Base" and "Main Category" dimensions.

Finally the data points are “discovered” by identifying the unique combinations of pairs *[dimension].[member]* throughout the complete set of categorised cells[6].



The outcome of this automatic process needs finally to be analysed by the business experts, as there are two possible reasons for different cells pointing to the same data point: either the cells contain exactly the same business information, and so the result was expected (or then realised as true), or there is a mistake that needs to be corrected in the model, requiring a new template analysis.

Conversely, there is also the possibility of two cells that are known *a priori* to be the same data point not showing the same categorisation, which also calls for another look at the analysis.

A DataPoint represents a specific item of business information. In future versions of the DPM it is possible that the specific categorisation used to describe/identify this business information might change – for example new more specific members may be added to a domain, or a dimension might be realised to be expressing two independent concepts, and be broken into two dimensions. To indicate that the underlying meaning of the business concept is not changing, just the categorisation of it within a taxonomy[7], DataPoints are related to one or more DataPointVersions, which represent the categorisation of a DataPoint which is/was valid during a specific period of time. Clearly in this first version all DataPoints map directly to a single DataPointVersion.

Validation Rules

Data validation rules are also represented in the Data Point Model database, please see the accompanying “Eurofiling – Validations.pdf” file for a presentation from the 2013 Eurofiling event in London, which gives a brief outline of the representation of these rules.

[6] For illustration, a string consisting of the concatenation of all dimension-member pairs, sorted alphabetically (a “CategorisationKey”) is given for each DataPointVersion, this should be unique for all DataPointVersions in force at a particular date. A similar string is given for each AxisOrdinate, indicating the dimensional attrbiutes contributed by that ordinate.

[7] And so indicate, for example, that time series comparisons of the value of this data point including values from before and after the re-categorisation are valid.

Appendix 1 – The representation of "Open" Tables

The model contains two major types of tables, "closed" – where each axis of the tables has all its required values explicitly listed, and therefore the precise size of the reported table is known, and "open", where one or more axis is "open" (allows a variable number of entries, chosen either from a restricted list (e.g. counterparty sectors), or of a particular type (e.g. any integer).

Closed tables fall into two main types, those with X and Y axes only (which are simply plain tables), or those that also specify a Z axis, and are made up of multiple sheets, each with a complete copy of the table, one for each ordinate on the Z axis.

Open tables also fall into two types – lists and table with optional sheets. Tables with optional sheets are simply normal tables with an open z axis, for which a copy of the table should be completed for each applicable values. The open z-axis for these tables typically allows values chosen from an explicit domain (e.g. a sheet per currency or per country where significant exposures are present etc.).

List tables are slightly more complex, they represent a table where a series of rows with identical columns must be entered, one for each item of a particular kind reported (e.g. a row per entity in a banking group, a row per security held, a row per transaction etc.). These tables have an open y-axis.

For both open z and open y-axis tables, the unknown number of entries on the axis is represented in the DPM database by an AxisOrdinate with OrdinateCode of "999".

For open y axis tables, the anticipated rendering involves a special column, into which the identifier of each particular row is entered. In many cases this identifier will simply be a (otherwise meaningless) number (e.g. a line number). In the database this is represented by the row AxisOrdinate ("999") being associated with an open dimension describing the nature of this identifier (such as the code of the security or obligor grade), and a Member with MemberID 999 (to indicate that the entire row is associated with a (unknown) value from the dimension). The columns axis of these tables includes an AxisOrdinate which is also associated with this open dimension and has the field IsRowKey set to true, which represents the column into which this code will be entered.

Data points for open tables

For "open" tables (either tables with open z (C 09.02) or y (C 27.00) axes), the number of potential values is unknown. Since each copy of the basic repeating unit of the table (i.e. either sheet or row) is identical in attributes to the others, except for the value for the open dimension, in the DPM this repeating unit is represented only once, with one set of datapoints. This means that in an actual report, each there will be many facts (one for each sheet/row) that are associated with the same data point (distinguished by the associated value for the open dimension).

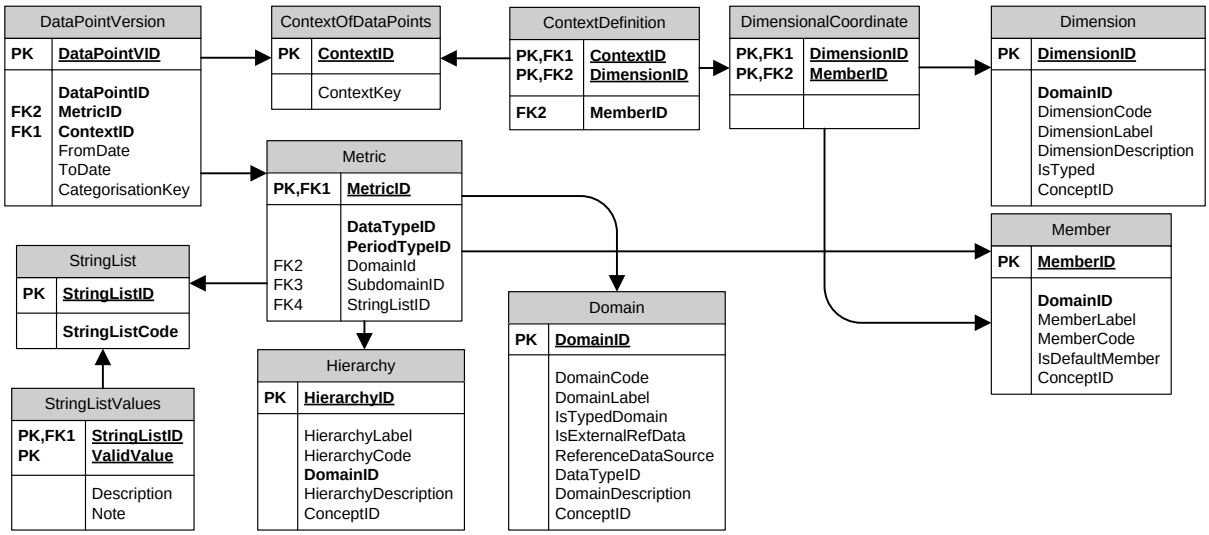
In contrast, "closed" tables with multiple sheets (e.g. C 07.00.a) have a distinct, known dimensional attribute set for each sheet, the cells on different sheets can be distinguished, and so each is identified in the model as a unique data point.

Appendix 2 - Physical Concerns – towards XBRL

In order to illustrate the link to the physical implementation of this Data Point Model for second level reporting from National Supervisory Authorities to the EBA, which is expected to be implemented in XBRL, some additional tables have been included.

Data facts in an XBRL document are identified via a combination of "primary item", which corresponds to the metric of the DPM, and a reference to a "context" which contains a set of dimension-members pairs[8]. One of the DPM dimensions, *Metric*, has been chosen to map directly to the primary item[9], and the remaining dimensions that categorise a data point will be represented in the context.

To represent this, each DataPointVersion is linked both to a Metric entry, which expresses the (fundamental) meaning, data type[10] and "period type"[11] of the data point (and is linked to a member of the *Metric* dimension), and to a ContextOfDataPoints entry, which represents the unique set of the remaining dimension member pairs[12].



The identifier of members used in XBRL are given by the MemberCode field, and the canonical namespace prefix for a domain is given by the DomainCode of the associated domain.

Restrictions on the allowed values for a metric (i.e. for the cell(s) where that metric is used) are given by the data type, and potentially further indicated

---

[8] As well as a date or date-range and an identifier of the entity the fact relates to.

[9] Hence each "metric" option is also a member of the Metric domain. For simplicity the MemberID is used also as the MetricID).

[10] Money, string, date etc.

[11] Whether the value is "as of" an instant in time, like a balance or stock, or measures something between two dates, like a change or flow.

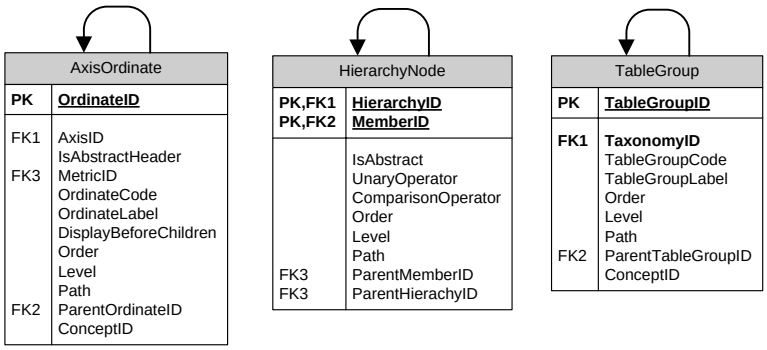
[12] Again for illustration a field called ContextKey is included in this table, which is produced in a similar fashion to the Categorisation Key, but omitting the *Metric* dimension. For the avoidance of doubt, it is not expected that the values of either the ContextKey or the ContextID field from the database will appear anywhere in the XBRL instance files, it is instead the unique set of dimension member pairs that would be matched to a specific ContextOfDataPoints to (along with the metric) identify the datapoint reported.

- for strings – by an associated StringList, which enumerates the allowed values.
- for “codes” (data type “e”) – the values reported for cells using this metric are required to be one of the members of a specific domain (indicated by the DomainID field), and may be further restricted to one of the members of a particular hierarchy[13], indicated by the SubdomainID field.

[13] which may be included purely for the purpose of enumerating this set of values

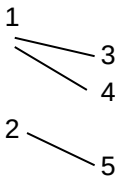
Appendix 3 - Tree structures in the Model

Three tables in the DPM database represent “tree” / “parent-child” structures (i.e. where an entry can have multiple child entries “below” it, each of which can also have children etc.) – Hierarchy Node, AxisOrdinate and TableGroup.



In each case the tree structure is represented in the database in two ways, by a link to a “Parent” entry from each child (i.e. as an “adjacency list”), and with a “Path” field giving the full path from a child through its ancestors to the first level (“path enumeration”), both of which use the Order field to indicate the overall order of the nodes within the tree (and so also the order within each sibling group).

i.e. an arrangement such as

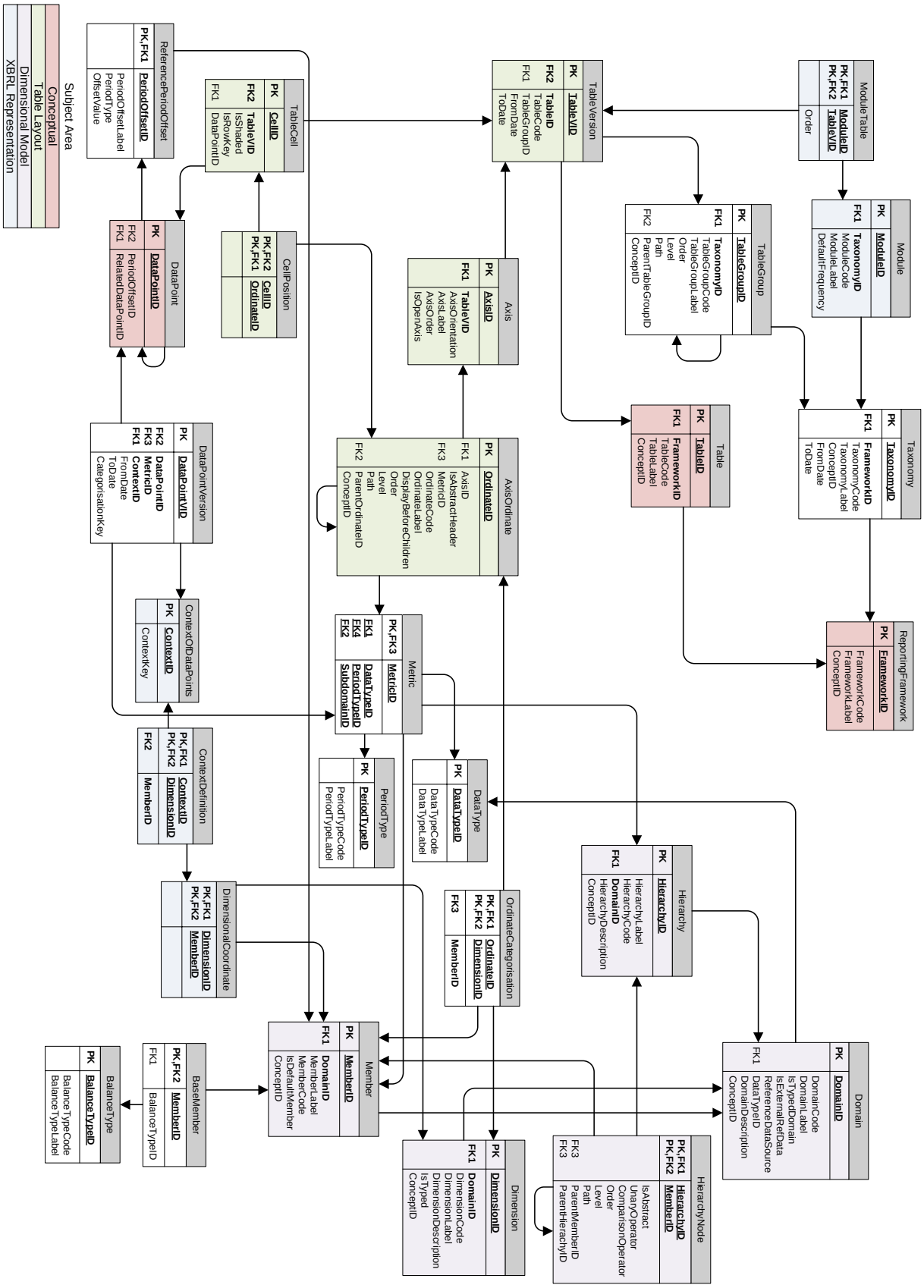


Would be represented as

Node	Parent	Path	Order	Level
1		1.	1	1
3	1	1.3.	2	2
4	1	1.4.	3	2
2		2.	4	1
5	2	2.5.	5	2

These representations are redundant, each conveys the same information. Both are included merely for convenience, as each representation is more convenient to work with when using certain tools/technologies, or for different purposes.

Appendix 4 – Complete data model



Appendix 5 – Table and Field Descriptions

Axis

Represents either a row, column or sheet of a particular table.

Field Name	Type	Size	Description
AxisID	Long	4	Artificial ID
TableVID	Long	4	Table to which this axis belongs
AxisOrientation	Text	1	Either X,Y or Z for row, column or sheet respectively
AxisLabel	Text	255	Descriptive label (English). Most relevant for Z axes, where it can be used e.g. to label a text or dropdown box for the user to enter/choose the z-axis value
AxisOrder	Integer	2	For multiple Z-axes, indicates in what order the axes should be shown (i.e. in what order any text or dropdown boxes used to represent the axes should be displayed)
IsOpenAxis	Boolean	1	An "open" (vs. "closed") axis allows a variable number of entries, either chosen from a list of options or of a type of value. Used e.g. for vertical list tables, where a "line number" is used, and for "sheet per country/currency/sector" type tables.

AxisOrdinate

Represents a specific position on a "closed" axis (or a generic placeholder for an "open" one). Tree structure of ordinates represent indenting / nesting of rows or columns (used for e.g. "of which" type breakdowns)

Field Name	Type	Size	Description
OrdinateID	Long	4	Artificial ID
AxisID	Long	4	Axis to which this ordinate belongs
IsAbstractHeader	Boolean	1	If true, this ordinate does not represent any "reportable data", e.g. it may either be displayed as a completely "grey row", or as just a heading with no column for values etc.
MetricID	Long	4	Fundamental nature of the values reported against this ordinate (i.e. in this column, row or sheet), if applicable.
OrdinateCode	Text	4	Short code
OrdinateLabel	Text	255	Descriptive label (English)
Order	Long	4	Position of this ordinate within its set of siblings, if any (Tree structure information)
Level	Long	4	Level of this ordinate, lower level numbered ordinates "contain" higher numbered ones, i.e. lower levels are nearer the root (Tree structure information)
Path	Text	255	Path from the root of the axis to this ordinate (Tree structure information)
ParentOrdinateID	Long	4	Parent of this ordinate, if any - i.e. the level immediately above (Tree structure information)
DisplayBeforeChildren	Boolean	1	Hint for display. If yes/true then this ordinate is intended to be displayed above or to the left of any child ordinates, if false it should be shown

			below or to the right of them.
ConceptID	Long	4	Reference to concept (change, owner and translation) information - to be implemented

BalanceType

Field Name	Type	Size	Description
BalanceTypeID	Long	4	Artificial ID
BalanceTypeCode	Text	1	Short Code
BalanceTypeLabel	Text	255	Descriptive label (English)

BaseMember

Field Name	Type	Size
MemberID	Long	4
BalanceTypeID	Long	4

CellPosition

Links a cell in a table to its position on the axes of that table

Field Name	Type	Size	Description
CellID	Long	4	Cell that is described
OrdinateID	Long	4	Position on an axis of a table

ContextDefinition

A specific dimension member pair used to categorise one or more data point versions.

Field Name	Type	Size	Description
ContextID	Long	4	Context (of data point version(s)) to which this categorisation applies
DimensionID	Long	4	Dimension being considered
MemberID	Long	4	Categorisation in that dimension.

ContextOfDataPoints

A specific combination of dimension member pairs (excluding the metric dimension) used to categorise one or more data point versions. Intended to illustrate the intended approach to mapping to XBRL.

Field Name	Type	Size	Description
ContextID	Long	4	Artificial ID
ContextKey	Text	255	Concatenation of the [DimensionCode MemberID] pairs, excluding the metric dimension, that categorise one or more data point versions. (For illustration only)

DataPoint

A unique item of information. Indicates conceptual identity independent of the specific categorisation in any particular taxonomy.

Field Name	Type	Size	Description
DataPointID	Long	4	Artificial ID
PeriodOffsetID	Long	4	The relationship between this and the underlying data point. (Some data points are intimately related to others, representing the same information at a different time, i.e. "balance now" and "balance last year").
RelatedDataPointID	Long	4	The underlying data point. (Some data points are intimately related to others, representing the same information at a different time, i.e. "balance now" and "balance last year").

DataPointVersion

The categorisation of a DataPoint which is/was valid during a specific period of time.

Field Name	Type	Size	Description
DataPointVID	Long	4	Artificial ID
DataPointID	Long	4	Data point for which this is a specific version of the categorisation
MetricID	Long	4	Fundamental meaning, data and period type of this data point
ContextID	Long	4	Remaining dimensional categorisation
FromDate	Date	8	Date from which this data point version is/was valid
ToDate	Date	8	Date from which this data point version is/was valid
CategorisationKey	Text	255	Concatenation of the [DimensionCode MemberID] pairs that categorise this data point version (including metric). (For illustration only)

DataType

Field Name	Type	Size	Description
DataTypeID	Long	4	Artificial ID
DataTypeCode	Text	1	Short code (one character)
DataTypeLabel	Text	50	Descriptive label (English)

Dimension

Category/aspect used to describe and differentiate data points, each relates to one specific feature. Allowed values are taken from a domain. If these are explicitly listed they are called members.

Field Name	Type	Size	Description
DimensionID	Long	4	Artificial ID
DomainID	Long	4	Domain from which the allowable values for this dimension are taken
DimensionCode	Text	3	Short code
DimensionLabel	Text	255	Descriptive label (English)
DimensionDescription	Memo	0	Longer description (English)
IsTyped	Boolean	1	"Typed" dimensions allow any value of a particular form (i.e. any string of certain length or pattern, any number, a date etc.), "explicit"

			dimensions only allow a choice from a given list of members.
ConceptID	Long	4	Reference to concept (change, owner and translation) information - to be implemented

DimensionalCoordinate

A specific dimension and member pair.

Field Name	Type	Size
DimensionID	Long	4
MemberID	Long	4

Domain

Groups values of a particular kind/addressing a particular concept. May have an explicit list of allowable values (members), or else specify values of a particular type or pattern (a "typed" domain). Provides the allowable values for a dimension.

Field Name	Type	Size	Description
DomainID	Long	4	Artificial ID
DomainCode	Text	3	Short code
DomainLabel	Text	255	Descriptive label (English)
IsTypedDomain	Boolean	1	"Typed" domains allow any value of a particular form (i.e. any string of certain length or pattern, any number, a date etc.), "explicit" dimensions only allow a choice from a given list of members.
IsExternalRefData	Boolean	1	Indicates if the domain value list is obtained primarily from an externally published authoritative list.
ReferenceDataSource	Text	255	Indicates where the value list is obtained from (if not defined by the domain owner).
DataTypeID	Long	4	Indicates the allowed type of values (for Typed domains).
DomainDescription	Memo	0	Longer description (English)
ConceptID	Long	4	Reference to concept (change, owner and translation) information - to be implemented

Hierarchy

Hierarchies specify how members relate to each other, and can also define the aggregations from lower to upper levels in the hierarchy.

Field Name	Type	Size	Description
HierarchyID	Long	4	Artificial ID
HierarchyLabel	Text	255	Descriptive label (English)
HierarchyCode	Text	50	Short code
DomainID	Long	4	Domain this hierarchy relates to
HierarchyDescription	Memo	0	Longer description (English)
ConceptID	Long	4	Reference to concept (change, owner and translation) information - to be implemented

HierarchyNode

Represents a node in a hierarchy of members, specifying how members relate to each other, and can also define the aggregations from lower to upper levels in the hierarchy.

Field Name	Type	Size	Description
HierarchyID	Long	4	Hierarchy to which this node belongs
MemberID	Long	4	Member this node represents
IsAbstract	Boolean	1	
ComparisonOperator	Text	2	Indicates the relationship between this node and the aggregation of its children
UnaryOperator	Text	1	Indicates the contribution of this node to the aggregation of its siblings
Order	Long	4	Position of this node within its set of siblings, if any (Tree structure information)
Level	Long	4	Level of this node, lower level numbered nodes contain higher numbered ones, i.e. lower levels are nearer the root (Tree structure information)
Path	Text	255	Path from the root of the hierarchy to this node, only MemberIDs are listed (Tree structure information)
ParentHierachyID	Long	4	Must always be the same as HierarchyID (included purely due to limitations of MS Access).
ParentMemberID	Long	4	Indicates the parent of this node, if any - i.e. the level immediately above (Tree structure information)

Member

An explicit possible value within a domain.

Field Name	Type	Size	Description
MemberID	Long	4	Artificial ID
DomainID	Long	4	Domain to which this member belongs
MemberCode	Text	50	Short code
MemberLabel	Text	255	Descriptive label (English)
IsDefaultMember	Boolean	1	
ConceptID	Long	4	Reference to concept (change, owner and translation) information - to be implemented

Metric

The fundamental conceptual meaning of a piece of information.

Field Name	Type	Size	Description
MetricID	Long	4	Artificial ID - Matches a MemberID from which descriptive labels may be obtained
DataTypeID	Long	4	The type of data present
PeriodTypeID	Long	4	The time characteristics of the information, is it for a specific point in time or time period
SubdomainID	Long	4	

Module

Field Name	Type	Size
ModuleID	Long	4
TaxonomyID	Long	4
ModuleCode	Text	255
ModuleLabel	Text	50
DefaultFrequency	Text	255

ModuleTable

Field Name	Type	Size
ModuleID	Long	4
TableVID	Long	4
Order	Long	4

OrdinateCategorisation

A pair of dimension and member describing one aspect of the categorisation of a particular position along an axis of a table

Field Name	Type	Size	Description
OrdinateID	Long	4	Artificial ID
DimensionID	Long	4	The dimension considered
MemberID	Long	4	The relevant value of that dimension

PeriodType

Field Name	Type	Size
PeriodTypeID	Long	4
PeriodTypeCode	Text	1
PeriodTypeLabel	Text	50

ReferencePeriodOffset

Ways in which a data point may be related to another data point such that for e.g. time series analysis purposes they may need to be considered as the same data series.

Field Name	Type	Size	Description
PeriodOffsetID	Long	4	Artificial ID
PeriodOffsetLabel	Text	255	Descriptive label (English)
PeriodType	Text	255	Nature of the time period linking the data points
OffsetValue	Integer	2	Degree of offset from the data point considered the "latest" / "current period" / "default" etc.

ReportingFramework

Overall reporting framework - high level, stable concept

Field Name	Type	Size	Description

FrameworkID	Long	4	Artificial ID
FrameworkCode	Text	255	Short code
FrameworkLabel	Text	255	Descriptive label (English)
ConceptID	Long	4	Reference to concept (change, owner and translation) information - to be implemented

Table

Most of the times a table will be the same as a business template, except when, for modelling reasons, the templates had to be normalised and split into two or more tables

Field Name	Type	Size	Description
TableID	Long	4	Artificial ID
FrameworkID	Long	4	Reporting framework this table belongs to
TableCode	Text	255	Short code
TableLabel	Text	255	Descriptive label (English)
Template	Text	255	Name/Code of template of (part of) which this table is a representation.
ConceptID	Long	4	Reference to concept (change, owner and translation) information - to be implemented

TableCell

Represents an individual intersection of row, column (and sheet) for a particular table.

Field Name	Type	Size	Description
CellID	Long	4	Artificial ID
TableVID	Long	4	(The version of a) table this cell is part of
IsShaded	Boolean	1	Is no data expected to be entered into this cell? - either because it is not required, or because this cell forms part of a heading, or the intersection of its row and column (and sheet) has no logical meaning.
IsRowKey	Boolean	1	Does this cell represent the artificial code/ID used to identify a row of data (in an open table/list)?
DataPointID	Long	4	Business information contained in this cell

TableGroup

Grouping (for information purposes only) of (specific versions of) tables within a taxonomy

Field Name	Type	Size	Description
TableGroupID	Long	4	Artificial ID
TaxonomyID	Long	4	Taxonomy to which this table group belongs
TableGroupCode	Text	255	Short code
TableGroupLabel	Text	255	Descriptive label (English)
Order	Long	4	Position of this table group within its tree (Tree structure information)
Level	Long	4	Level of this table group, lower level numbered groups contain higher numbered ones, lower levels are nearer the root (Tree structure information)
Path	Text	255	Path from the root of the grouping to this table group (Tree structure information)

ParentTableGroupID	Long	4	Parent of this table group, if any - i.e. the level immediately above (Tree structure information)
ConceptID	Long	4	Reference to concept (change, owner and translation) information - to be implemented

TableVersion

The specific description of a particular table from a reporting framework, within a taxonomy, valid during a particular time period. Several TableVersions may represent the evolution of a particular Table over time.

Field Name	Type	Size	Description
TableVID	Long	4	Artificial ID
TableID	Long	4	Table that this is a particular version of
TableCode	Text	255	Short code
TableGroupID	Long	4	Group of tables in which this (version of this) table is included
FromDate	Date	8	Date from which this version of this table is/was valid
ToDate	Date	8	Date until which this version of this table is/was valid

Taxonomy

A specific description of the classification of the tables and data points of a reporting framework, at a particular point/period in time

Field Name	Type	Size	Description
TaxonomyID	Long	4	Artificial ID
FrameworkID	Long	4	Reporting framework this taxonomy describes
TaxonomyCode	Text	255	Short code
TaxonomyLabel	Text	50	Descriptive label (English)
ConceptID	Long	4	Reference to concept (change, owner and translation) information - to be implemented
FromDate	Date	8	Date from which this taxonomy is/was valid
ToDate	Date	8	Date until which this taxonomy is/was valid