

Improving XBRL for Data Modelling

How can XBRL be improved to help Taxonomy authors for large reporting systems?

A recent article identified the potential for conflict between the modelling approaches of XBRL and the Data Point Methodology (DPM). However, no discussion would be complete without reviewing why the European Banking Authority (EBA) and European Insurance and Occupational Pensions Authority (EIOPA) selected to use DPM and then translate to an XBRL taxonomy, rather than starting with XBRL in the first place.

Was it a lack of features in XBRL, was it a lack of understanding, or just simply a lack of tools? This article argues there are still some features missing, many of these are on XBRL International’s Open Information Model (OIM) work plan. but some like ‘versioning’ are yet to be addressed.

The other key insight is that there appears to be no incentive for software vendors to build the type of data modelling tools that large XBRL Taxonomy authors need.

Reviewing the key requirements of large reporting frameworks, like those for EBA CRD and EIOPA Solvency, this paper assesses the status of XBRL specifications to meet them and to enable XBRL to provide master data management capabilities for reporting frameworks.

The Benefits and Issues of XBRL

The XBRL specifications are designed to support a diverse set of business information reporting applications across the world. There are now over two hundred major XBRL reporting frameworks built around this open standard, there is a large community of experts and a growing range of software vendors.

One of the strengths of the XBRL standards, apart from the common core, is their independence from each other, allowing XBRL taxonomy designers to choose the specifications they wish to use. However, its weakness is that the development as independent specifications means that there is little interoperation between them.

This is particularly noticeable for the developers of large reporting frameworks. To help understand what the specific issues are, it is worth reviewing the feedback from the EBA during its recent presentations on DPM 2.0, aka the ‘DPM Refit’. The EBA presented a case for deepening the use of DPM versus a more standardised XBRL approach. The slide below is an example of how they compare their DPM with XBRL.

Data dictionary features	XBRL	DPM
▪ Formal definition of data requirements	✓	✓
▪ Glossary of business terms	✓	✓
▪ Dimensional data definitions	✓	✓
▪ Templates rendering	✓	✓
▪ Explicit metamodel	✗	✓
▪ Invariant data point identifiers	✗	✓
▪ Historisation of concepts and relationships	✗	✓
▪ Metadata exploration with standard query languages	✗	✓
▪ Verifiable global model consistency	✗	✓
▪ Support frameworks integration	✗	✓

We believe that many of the EBA’s observations are biased because of their decision to base their internal data storage system on DPM. We would counter on behalf of the XBRL community that:

-) The XBRL Taxonomy produced by the DPM tools provides no obvious semantic guide to definitions, being made up of few high-level concepts and broken down by numerous dimensions, e.g., there is globally one concept for ‘assets’ whereas the IFRS taxonomy has many types of assets that are subsets of the wider concept. The EBA’s Credit Risk Directive (CRD) taxonomy is a so-called ‘highly dimensional model’. Good for computers, but poor for helping the reader’s understanding, which is important in transmitting requirements in heterogeneous reporting systems.
-) The DPM process also generates numerous, low-level rules to check the quality of the data, rather than a high-level semantic rule, such as ‘all totals should add up’. Checking the consistency of such a large volume of low-level DPM rules (around 8,000) in any automated way is somewhat doubtful.
-) Many of the differences cited in the EBA’s slide are related to the specific implementation, so issues such as integration, and invariant identifiers exist only in the eye of the beholder.

The issue of ‘versioning’ however is real. XBRL includes a superficial version and best practice guidelines for documenting differences between versions, but no more. Against this, the EBA claim that DPM supports ‘historisation’ of concepts is based upon its proprietary implementation of DPM. If XBRL is to provide some form of ‘master data management’ for large reporting frameworks, then versioning is a critical feature. However, it is worth noting that data exchange has different versioning requirements to those of data analysis systems, as discussed in more detail later, but the requirement is important for all XBRL systems.

The EBA is also planning to adopt XBRL’s new Open Information Model (OIM) and in particular the xBRL-CSV format for submissions to reduce the file size of reports. However, once again rather than focus on improving the XBRL design and performance potential it has selected to use a CSV structure which explicitly uses the ‘DPM-ID’ - a construct of the EBA data storage system that is semantically ‘vacant’ and provides no assistance to optimise XBRL processing. To learn more about this approach and why we believe that it is a bad idea, please refer to the earlier article [‘DPM and XBRL’](#).

So today, if the EBA were starting afresh, the big question would be ‘Would they still use DPM to define the model of the data to be collected or would they use XBRL’?

Our view is that the EBA would still find a lack of ‘taxonomy development’ tools with which to build a ‘good’ semantic model that is easy to maintain. We believe that the OIM initiative is a massive step in the right direction, but the independent XBRL specifications still hinder the process. The next sections review the details of OIM and where they are heading, plus some recommendations for improving the XBRL modelling capabilities.

For now, DPM works for the EBA and EIOPA as a useful mechanism for their internal systems (... they could use XBRL better, but that is for later). The real issue that the EBA reveals for the XBRL community, is that defining large-scale reporting frameworks in XBRL is a largely manual and complex process. Other large frameworks, like the IFRS taxonomy, experience similar issues. So, what needs to be improved?

OIM and Future xBRL standards

The Open Information Model (or “OIM”) is XBRL International’s strategic effort to simplify and modernise important aspects of the XBRL Standard by defining a model that represents the meaning of the standard, without reference to a specific syntax, i.e., it removes the dependence upon XML. OIM defines multiple and interchangeable formats, which can be added to over time.

-) xBRL-CSV – condenses the data into a highly compact tabular form to enable the collection of large quantities of data.
-) xBRL-JSON - provides XBRL data in a format that is simpler to process and present.
-) xBRL-XML – continues to support a wide range of reporting requirements.

The skills and effort to develop rules to validate the data against (XBRL Formulas) have proven another area of concern for Taxonomy authors. The XBRL Standards Board (XSB) has recently provided a pathway forward for XBRL formulas in an OIM world:

-) Starting with Formula 2.0, which will remove the XPath syntax and formalise the specification of XF, or Text-based Formula which provides the same functionality as XBRL Formula but is quicker to write and easier to read.
-) Eventually, the plan is to develop a new specification which will encompass rules for both the XBRL instance and taxonomy, based upon the new OIM Taxonomy specifications. It also means a name change to 'XBRL Rules 3.0' to recognise the significance.

Confusingly for some, in XBRL there is another way to check simple relationships of the supplied data using the Calculation specification. Ideally, this should provide a simpler mechanism to define the major 'quality checks' found in financial reporting models, e.g., roll-ups, roll forward, and aggregations. The Calculation specification is being updated and the plan for version 2.0 includes dimensional aggregation capabilities. XBRL Formula would then only be used for more complex rules as well as structural validations.

OIM will keep this flexibility for taxonomy authors, but still leaves questions such as 'Should you use XBRL Formulas or Calculations':

-) The EBA develops XBRL Formula from its DPM notation, which is defined by business users as part of their spreadsheet templates but does not describe the inherent relationships in the Tables or use Calculations to add up basic hierarchies.
-) The IFRS taxonomy on which the European Securities Management Authority's (ESMA) ESEF is based, uses both Calculations and Formulas, but does not use Tables. In 'Open Reporting' frameworks like ESEF the issuer develops their own Table structures. Poor modelling of these, means calculations are often missed or only partially included in the Taxonomy, resulting in numerous data quality issues.

So, does OIM go far enough? We strongly believe that there needs to be some additional thought given by the XBRL community as to how hypercubes, Tables, Calculations, and Formulas can work together to help deliver better XBRL models.

Managing Large Reporting Frameworks

If XBRL is to be used to model large-scale collection systems, then we need to go back to some of the underlying issues raised by the EBA, EIOPA and the European NCAs, which not only implement the collection of EBA and EIOPA reports from thousands of European banks and insurance firms, but also extend them for local reporting requirements.

It is a large area to cover, so it is best to start by breaking the issues down into smaller sub-areas:

-) Taxonomy development and maintenance
-) Versioning
-) Large files
-) Numerous and of complex XBRL formulas

Taxonomy Development and Maintenance

Data modelling is arguably the most impactful decision for a data reporting team. It determines your architecture and the path that the project will follow. Modelling large, complex data sets has always presented designers with decisions and issues.

Large XBRL taxonomies (data dictionaries) can reference other XBRL Taxonomies (eXtensible) as building blocks, and can be separated into numerous entry points, each containing multiple Table definitions or ELRs, that make it easier to model the individual parts. This helps but does not go far enough to really help designers develop a 'good', performant semantic model and manage this over

time, nor does it incentivise developers to build the kind of tools that would support designers in this process, such as.

- J The XBRL Dimensions specification is used to define hypercubes, while Table Linkbases can use the dimensions and can be linked to hypercubes, however, each Table Linkbase must be independently specified, i.e., coded. More code equals more development and more maintenance. Why is there no ability to produce a Table Linkbase specification directly from the reporting hypercube? This would encourage taxonomy designers to think carefully about the hypercube structure and the Tables.
- J The Table Linkbase specification defines a tabular presentation layer for rendering. However, it does not provide any simple 'Tabular arithmetic', such as row totals, column totals or subtotals. This idea of 'dimensional aggregation' has been proposed before and has resurfaced in Calculations 2.0. The designer could use Formula code today, as shown by the EBA, but if the process is automated and part of the underlying model, then it reduces code and taxonomy designers will be more structured in their Table designs.
- J Designers are clearly interested in xBRL-CSV and xBRL-JSON formats. Adding some simple ideas to help build and manage definitions over time, reduces code and puts the focus on the model:
 - o Method for the direct generation of xBRL-CSV from table and hypercube definitions.
 - o Bidirectional linking of Table Linkbase and xBRL-CSV definitions.
 - o Direct rendering of xBRL-CSV data into Table Linkbase defined tables.

We believe that the addition of such simple features ensures that the XBRL specifications link together in a more consistent and supportive way – 'reunifies' the individual specification modules. You could call this 'master reporting management', which suggests a more structured and methodical way to develop a taxonomy, rather than using a 'hotchpotch' of different tools.

Versioning

Over time all reporting frameworks will develop and change as the elements, architecture, rules, and specifications used, need to be updated. From an XBRL perspective, two areas have been the focus of the XBRL Best Practices Task Force:

- J How to communicate the changes between taxonomy versions – more details at <https://www.xbrl.org/guidance/communicate-taxonomy-changes/>
- J How to manage taxonomy versioning – more details at <https://www.xbrl.org/guidance/communicate-taxonomy-changes/>

For most XBRL projects which are about the exchange of business information, these are sufficient. Although there is no technical specification to enable software to automatically update systems from the old to the new Taxonomy version.

The EBA's view of 'Versioning' is much deeper. They want to review when a 'concept' (data point) was first referenced, modified, and when it was deprecated, plus they want to capture who made the change and why. So, their vision is much closer to 'Master Data Management' where metadata on the model is collected so that the model itself can be reviewed.

Note, the EBA confuses things when they state that '(XBRL) cannot handle the evolution of a datapoint between releases, making them unsuitable for time series analyses. This, we believe, confuses collection systems with analysis systems. Analysis systems require a different approach as the source data comes in many different pipelines, needs to be transformed, and needs to be stored in a specific way to make it efficient for analysis, such as Time Series. In collection systems, the issue is how to make it easier for submitters to know what data to collect, how to test it is valid, and how

to make the transfer process efficient. These two aims and objectives can conflict, which is why most organisations split the two systems.

The benefits to XBRL of a more detailed taxonomy model and element versioning are:

-) Providing a standardised method for XBRL providers to update associated materials with a newer Taxonomy version would significantly help and reduce costs for vendors and hence, users.
-) Understanding how data definitions and rules have changed over time provides important background information.
-) for analytics and operational decision-making.

How important this is for most XBRL projects is questionable but for larger complex reporting frameworks it would clearly help their management. One caveat is that adding 'versioning' to XBRL is a large task and something that would need a real-life 'use case' as guidance.

Validation performance of large datasets

Concerns about the processing time of large reports have always been present, it is just the size of what is defined as a 'large' data file has risen exponentially. Any performance test will depend upon the environment that it runs in, i.e., give a program more memory and more CPU performance and it should run faster. So, the question should be rephrased to 'Is it running efficiently?' so that it scales.

When you analyse performance in large reporting frameworks, such as the EBA CRD and EIOPA Solvency, the issues mostly appear with record-based datasets, expressed as 'Open' tables. An Open Table is one where there is an unlimited number of rows, columns, or sheets. Record format or transactional data is often organised in a row per record, i.e., multiple related facts are grouped in a row. On other Tables, which contain relatively few, aggregate data points, performance has always been good for most XBRL processors.

The xBRL-CSV specification was specifically developed to handle the issues resulting from large record-based datasets. Firstly, it compresses the data, so report files are smaller and easier to transmit. Secondly, if the CSV structure follows the Table layout, i.e., according to their 'record format', then the data could be read in as rows which provides a 'natural' grouping of associated data, significantly improving the performance of XBRL formulas on large open tables.

This provides a huge performance improvement over xBRL-XML, where such Tables are expressed as a single fact per row and an XBRL processor must 'regroup' the individual rows forcing processors like UBPpartner's XPE employs an 'optimiser' to work out how best to group and filter the data for a given formula.

Note, when you combine large datasets with numerous low-level data quality checks, as created using the EBA's DPM, then you do see processing times increase. Unfortunately, the EBA's proposed approach for the collection of CRD data in xBRL-CSV does not help, as it has decided for the first time to fully introduce DPM notation directly into the XBRL model by selecting the following fixed xBRL-CSV format:

DPM_ID, Value, Unit

This is like the xBRL-XML model of one fact per line and then adding an extra lookup by using the semantically vacant 'DPM-ID' as the key in the CSV file. An indirection that from a local NCAs and submitters' point of view, offers no advantages. Instead, it constricts validation performance and makes conversion between other formats more difficult.

Incorporating semantic capability, such as Tabular Arithmetic into the underlying model helps XBRL processors understand the structure of the data that they are working with and then 'optimisers' can be formalised to improve performance based on the data and its structure.

The above should also be linked to the XBRL Filing Indicator specification, which also provides a mechanism to help partition large data sets into smaller logical sections. These logical sections can then be linked to both tables and formula sets. Being able to identify the appropriate subsections of the data and their associated taxonomy constructs allows XBRL processors:

-) Reduction of scope for formula and calculations, that currently target the complete data set.
-) Opportunity to break down processing into smaller operations, which use a subset of the model and the data and have the potential to be processed independently.

XBRL Formula Improvements

The capability to embed validation rules in a taxonomy is one of the most powerful features of XBRL for data exchange and the source of improved data quality. Today, as discussed, we have two methods – Calculations and XBRL Formula. The first is simple and easy to implement but limited, whereas XBRL Formulas provide so much more, but are hard to develop, as they are tied to XML. In addition, a designer of an ‘Open’ taxonomy, such as ESMA ESEF or for US GAAP, cannot currently write rules to check the extension Taxonomy built by the issuer.

In response to the last issue, XBRL US has built their own rules system (DQR) in a different technology, XULE. While this provides an immediate fix, it does not help standardisation across the XBRL community.

As highlighted above, the XSB recently announced that Formula 2.0, removing the XML dependency and formalising the use of XF (text-based formula). XBRL Rules 3.0 plans to make a clean break from the existing specifications and is expected to draw heavily on the experience gained by XBRL US. The above should help Taxonomy designers significantly and, allow for the easier definition of quality business rules with which to check the instance document and any extension Taxonomy.

In addition, XBRL Europe has recognised that the EBA’s and EIOPA’s DPM architecture has certain specific features with three models in one: datapoints, templates and semantic dimensions which need a ‘bridge’ to help move to the new XBRL features. It has set up a Task Force to review ‘XF-DPM’, which would both help translate between DPM rules and XBRL Formulas but also possibly improve the performance of the resulting XBRL Formulas. However, it would still suffer from defining ‘data quality checks’ at a datapoint level, so would produce many of these, rather than using the semantics embedded in a dimensional XBRL model.

Conclusions

XBRL continues to grow and support an ever-wider range of reporting frameworks. OIM is a crucial step in ensuring the future by supporting alternative formats, however, the XSB also needs to focus on recommendations that make it simpler and less resource-intensive to design and develop an XBRL Taxonomy that is consistent and performant.

When the EBA and EIOPA started on their XBRL journeys, XBRL provided a standards-based method of collecting the data they needed to supervise their market. However, XBRL did not have the features to allow them to model the data in XBRL, and they used DPM instead as their modelling tool. Now there is little incentive to change this setup and, in fact, the EBA’s DPM Refit proposal is to move the reporting framework more towards the DPM architecture.

The XBRL community needs to provide such an incentive. The XSB has delivered OIM and the XBRL-CSV format and made significant proposals on the updating of Formulas. However, this does not meet the need for XBRL to be used as the basis for a ‘master reporting management’ system. It also means that there is little incentive for vendors to provide the tools to help NCAs, in the same way, that the EBA and EIOPA are developing their own proprietary data management systems using DPM.

We believe that the OIM vision needs to be extended to:

-) Reunify the set of specifications.
-) Harmonisation of dimensions/tables/collection specifications.
-) Add Versioning capability.

Unfortunately, standards specification takes time to get consensus and then to develop the specifications. The XBRL community is based on volunteers contributing to this process, it is therefore important that work is relevant and prioritised if we are to see tangible benefits in a realistic timeframe. The authors would suggest that to start with extending the OIM roadmap so that users and developers have a clearer picture of future developments and give direction to authors like the EBA and EIOPA.

The authors are David Bell, Kapil Verma and Martin DeVille of UBPartner. Please send comments, corrections, and any alternative ideas to info@ubpartner.com.