The Challenges of Processing Large Instances

Ashu Bhatnagar, XBRL International
Michal Piechocki, BR-AG

HOH-SHOH B (3rd fl) (125)
2:00 – 5:30 pm

Seminar agenda

- 2:00 – 2:45 pm: PLID1
  - Introduction to the Processing of Large Instance Documents (45 minutes)
- 2:45 – 3:30 pm: PLID2
  - Identifying the Challenge Panel Discussion (45 minutes)
- 4:00 – 4:45 pm: PLID3
  - Business and Technical Solutions Panel Discussion (45 minutes)
- 4:45 – 5:30 pm: PLID4
  - Next Steps Panel Discussion (45 minutes)

Problem origins

- Unrestricted data requirements in form of „open tables"
- Unknown number of rows or columns or rows and columns
- Non-explicit content of rows or columns
- Challenges in estimating volumes of data, number of records
- Well-known aspect of database modeling

Row 1	[column 1]	[column 2]	…	[column n]
Row 2				
Row 3				

	[column 1]	[column 2]	…	[column n]
[row 1]				
[row 2]				
…				
[row n]				

Column 1	Column 2	Column 3
[row 1]		
[row 2]		
…		
[row n]		

Problem origins

- Aggregated reporting vs disaggregated reporting
- Large volumes of data expected within one table
- Performance requirements for solutions
- Flexibility requirements for data and metadata standards
- Reporting requirements examples:

List of investments with their ratings, values etc.	List of open positions for derivatives with their ratings, values etc.	Detailed list of a specific class of assets, liabilities, equities	List of loans given by a financial institution

How XBRL addreses open tables

	XBRL specification: tuple	XBRL specification: open context	Dimension specification: typed dimension
Taxonomy	Example: Taxonomy defines columns as primary items. Tuple (complex type) references (and binds) primary items as columns in one table.	Example: taxonomy may (but not necessarily) define a simple or complex XML item (or type) that is used in the instance document open context.	Example: taxonomy defines the typed dimension container as a simple or complex XML item (or type) that is used in the instance document as part of dimensional context definition.
Report	Pieces of data from cells in the open table are reported as values of facts. Facts are bound by their placement within the tuple tag as according to the tuple definition from the taxonomy. The infinity factor is provided by infinite number of tuple (group of facts) instantiations. Tables open from row and column perspective are addressed through tuple nesting.	Pieces of data from cells in the open table are most commonly reported as values of facts. Facts are bound by the context ID that references multiple facts to a single context with specific open context component which may be any XML-valid construct. The infinity factor is provided by infinite number of contexts that are distinguished by open context components.	Pieces of data from cells in the open table can be reported either as values of facts or values of explicit or typed dimensions in contexts or in any combination. Facts are bound by the context ID that references multiple facts to a single context with specific typed dimension context component value. The infinity factor is provided by infinite number of contexts that are distinguished by typed dimension context component.

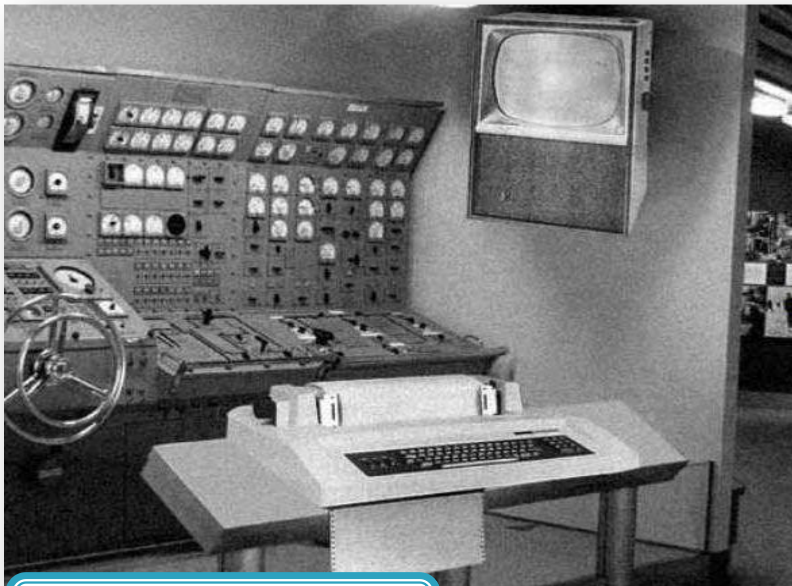
Solutions exist … but are they good enough for large reports?

Criteria for large XBRL report

- Multiple criteria:
  - Size of file
  - Memory requirements
  - Number of records
  - Number of dimensional contexts combinations
  - Number of unique contexts and number of facts
  - Processing time
- Exemplary estimation (market information):
  - Size: a ratio of 1 to 60 for XBRL file size to memory requirements of DOM model (100 MB instance document requires 6GB memory)
  - Size 2: a ratio of 1 to 10 for XML file size to memory requirements of DOM model (100 MB report = 1 GB memory)

Large report = report that requires large memory consumption

Before we begin…

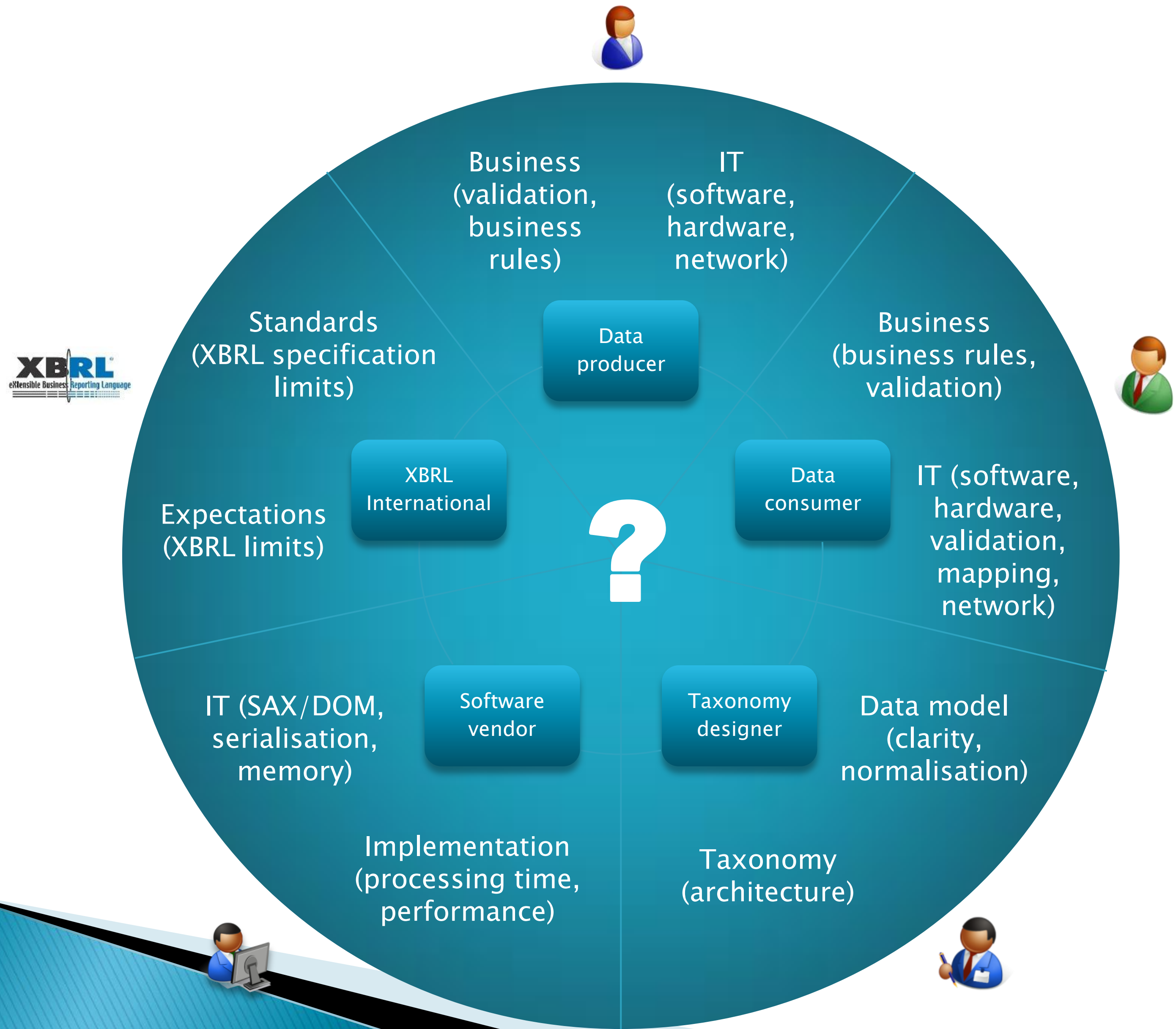


Validation issues

Transfer issues

Seriously?

Key perspectives of large reports issue



Data producer perspective

Knowledge requirements
(taxonomy architecture, software selection criteria)

Sourcing data
(mapping large data quantities to sophisticated taxonomy structures)

Quality assurance
(source data quality meeting the requirements of XBRL taxonomy)

- Business rules development and validation

Software & hardware requirements

- Performance for viewing and rendering
- Performance for specification validation
- Performance for business rules validation

Transfer
(submitting large XBRL instance documents)

Data consumer perspective

Knowledge requirements
(taxonomy change management, data analysis, taxonomy architecture design)

Mapping data
(mapping sophisticated large XBRL instances to DB / DWH / BI, identification of unique rows)

Quality assurance
(data quality meeting the requirements of XBRL taxonomy)

- Business rules development and validation

Software & hardware requirements

- Performance for viewing and rendering
- Performance for specification validation
- Performance for business rules validation

Transfer
(receiving large XBRL instance documents)

- Peak times and frequency
- Number of submissions

Taxonomy designer perspective

Knowledge requirements

- Impact on mapping (from and to)
- Impact on business rules
- Impact on software performance

Design approach

- Normalised vs non-normalised tables
- Explicit vs typed qualifiers
- Primary item vs dimension
- Hypercubes
- Data Point Model (DPM)

$$y=\frac{\sqrt{x}}{2}$$

XBRL Formula

- Designing filters
- XPath expressions (navigation)
- Advanced functions

Software & hardware impact

- Performance for viewing and rendering
- Performance for standard validation
- Performance for business rules validation

Software vendor perspective

XBRL standard

- Distribution of facts and contexts (e.g. dimensions)
- XPath processing of distributed information
- Processing of XML content in XBRL reports
- Lack of ordering of nodes in instance documents for streaming

Taxonomy design flexibility (practices)

- Mixed architectural designs (tuples nested and mixed with dimensions)
- Using non-standard functions (Formula) or design approaches

Reports creation flexibility (practices)

- Using open context components (XML)
- Using non-semantical components to carry information (contextRef)

Solution design

- Streaming events or having access to entire model (DOM vs SAX vs alternative models)
- Translating XBRL syntax to business logic and solution logic (common logical XBRL model, Infoset, AM)
- Internal serialisation of XBRL information

$$y = \frac{\sqrt{x}}{2}$$

Evaluating XBRL Formula

- XPath expressions allowing addressing of XML
- Syntactically-driven rules instead of semantically-driven ones

XBRL International perspective

Standard changes

- New version of base comprehensive specification
- Update (iso-morphic instance documents)
- Restrictions on Formula specification

Best practices development

- Taxonomy design
- Software design
- Abstract Model
- Guidance on handling large data sets

Communication

- XBRL standard limitations
- Maximum size of instance documents
- Best practices promotion

Ideas and solutions

1. Change XBRL specifications
   - Remove contexts as separate structure in XBRL 2.1
   - Identify XPath usage in Formula that can be mapped to other technologies
2. Introduce isomorphic instance documents
3. Order contexts appeaance in instance documents
4. Limit available design approaches for taxonomies
5. Indicate performance limits for XBRL
6. Promote optimised algorithms for XBRL software design (pure XBRL processor)
7. Promote Abstract Model 2.0
8. Promote partitioning of large instance documents
9. Consider streaming possibility

Speakers

Introduction to the Processing of Large Instance Documents
Ashu Bhatnagar
Michal Piechocki
Identifying the Challenge Panel Discussion
Yudi Sugalih
Michal Skopowski
Herm Fischer
Timo Schmuck
Business and Technical Solutions Panel Discussion
Paul Warren
Masatomo Goto
Julien Reber
Shweta Gupta
Herm Fischer
Bartosz Ochocki
Next Steps Panel Discussion
Makoto Koizumi
Paul Warren
Herm Fischer