TAXONOMY DESIGN - Thinking validations & analysis during the modeling phase

- Julien REBER
- Invoke
- Member of XBRL France

Agenda

- Data Point Modeling on dynamic tables
- Usage of data points vs Dimensions

Data Point Modeling on dynamic tables

- The DPM is a **structured representation of the data**, identifying all the business concepts, relations and validation rules.

Data Point Modeling on dynamic tables

- For a given reporting model, the **chosen modeling approach** has impacts on:

  - The **size** of the instance documents.
  - The **data capture** from the instance documents.
  - The **time required to filter the data** for a validation or an analysis.

Data Point Modeling on dynamic tables

- To illustrate our point, let's take the example of **Table T 03.01 - Intragroup liabilities**, LDR DPM 4.0.3, Single Resolution Board

Data Point Modeling on dynamic tables

- In this table, data points are defined using **metrics and functional dimensions** at **column level,** plus a **technical dimension** "eba_dim:IDL – Row ID" placed at the **row level** to dissociate the data as breakdown.

- The same table could have been designed using a different modeling approach by using a **one** (different) **metric per column** (without functional dimensions) and the **same technical dimension** "eba_dim:IDL – Row ID" at the **row level** as breakdown.

Data Point Modeling on dynamic tables

- What are the **consequences** of the two modeling approaches in term of instance **document size** ?

Impact on the size of the instance documents

- Here are the differences between the two approaches regarding the **XBRL** (XML) instance document composition and size:

	Current DPM approach	The least dimensional possible approach
Number of context for 1 row	8	1
Number of context for 1 000 rows	8 000	1 000
Number of context for 80 000 rows	640 000	80 000
Instance size for 1 row	8 KB	4 KB
Instance size for 1 000 rows	5 578 KB	1 903 KB
Instance size for 80 000 rows	452 453 KB	156 188 KB

Impact on the size of the instance documents

- With its "**concept" of aliases** for the contexts, the **OIM-CSV** drastically **reduces the size** of the instance documents.

Aliases in Json

Data in CSV

Impact on the size of the instance documents

- The benefit in term of size is the following:

	Current DPM approach	The least dimensional possible approach	OIM-CSV, csv data file
Instance size for 1 row	8 KB	4KB	**<1 KB**
Instance size for 1 000 rows	5 578 KB	1 903 KB	**250 KB**
Instance size for 80 000 rows	452 453 KB	156 188 KB	**4 250 KB**

Impact on the size of the instance documents

- Shown into a graphic view:
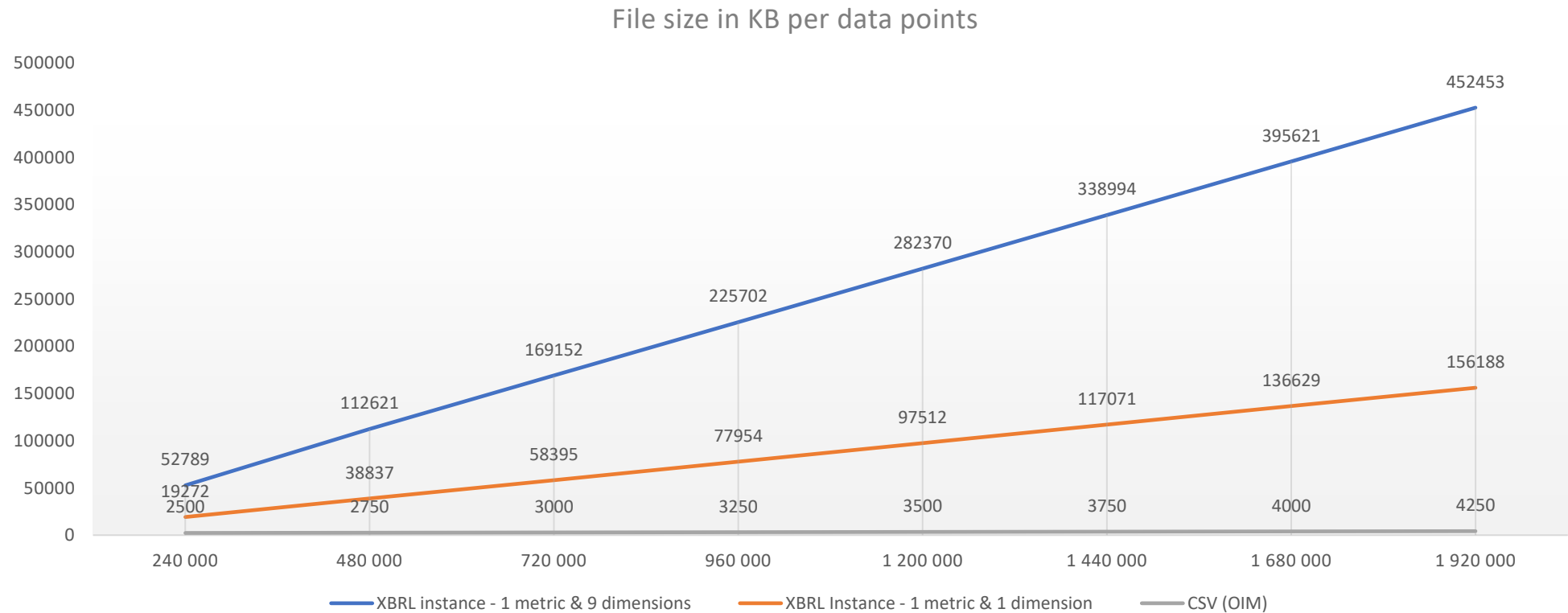
File size in KB per data points

52789	112621	169152	225702	282370	338994	395621	452453
19272	38837	58395	77954	97512	117071	136629	156188
2500	2750	3000	3250	3500	3750	4000	4250

X-axis: 240 000 | 480 000 | 720 000 | 960 000 | 1 200 000 | 1 440 000 | 1 680 000 | 1 920 000

Y-axis: 0 — 500000

— XBRL instance - 1 metric & 9 dimensions   — XBRL Instance - 1 metric & 1 dimension   — CSV (OIM)

Data Point Modeling on dynamic tables

- What are the **consequences** of the two modeling approaches in terms of the **data capture** ?

Impact on the data capture

- ## The benefit in terms of time regarding **data capture** is the following:

LDR, T0301, Loading data into memory

31%

Data Point Modeling on dynamic tables

- What are the consequences of the two modeling approaches in term of **data filtering** for **validation or analysis** ?

Impact on the filtering of the data

- To illustrate our point, let's take the example of the XBRL assertion **srb_v6511_m**, LDR DPM 4.0.3, Single Resolution Board :
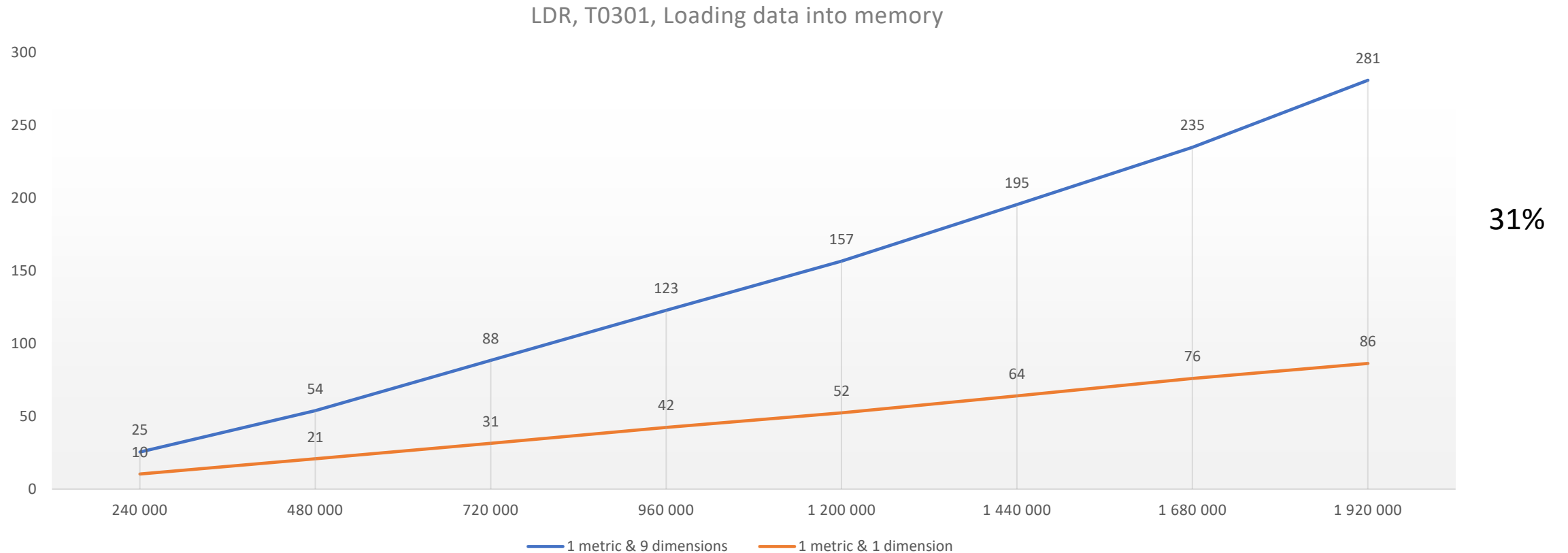 - Test: "iaf:numeric-greater-equal-than(iaf:sum(($a, $b)), $c)"

Impact on the filtering of the data

- On the **current modeling** approach, the assertion srb_v6511_m **filters the data** as follow:

	Global filters on assertion	Filters on variable $a	Filters on variable $b	Filters on variable $c
eba_dim:IDL	disctinct values			
eba_dim:ROL	eba_RP:x0			
eba_dim:MCJ	eba_MC:x0			
eba_dim:MRE	eba_BT:x0			
concept		eba_met:mi602	eba_met:mi603	eba_met:mi81
eba_dim:BAS		eba_BA:x8	eba_BA:x8	eba_BA:x11
eba_dim:MCY		eba_MC:x29	eba_MC:x29	eba_MC:x275
eba_dim:RPR		eba_RP:x34	eba_RP:x34	eba_RP:x0
eba_dim:MCK		eba_MC:x0	eba_MC:x0	eba_MC:x29
eba_dim:OFS		eba_OF:x0	eba_OF:x0	eba_OF:x10

Common filter

Variable specific filters

- The **contexts** are different per row

Impact on the filtering of the data

- On the **least dimensional approach**, the assertion srb_v6511_m **filters the data** as follow:

Common filter

	Global filters on assertion	Filters on variable $a	Filters on variable $b	Filters on variable $c
eba_dim:IDL	disctinct values			
concept	■■■■■■■■■	eba_met:mi602	eba_met:mi603	eba_met:mi81

Variable specific filters

- The **contexts** are the same per row

Impact on the filtering of the data

- The **difference between those 2 approaches** on the srb_v6511_m assertion for the **"same" data points** controlled is :

srb_v6511_m - formula evaluation - 1 metric & 9 dimensions hypercubes vs 1 metric & 1 dimension hypercubes

38%

NUMBER OF DATA POINTS

——— 1 metric & 9 dimensions   ——— 1 metric & 1 dimension
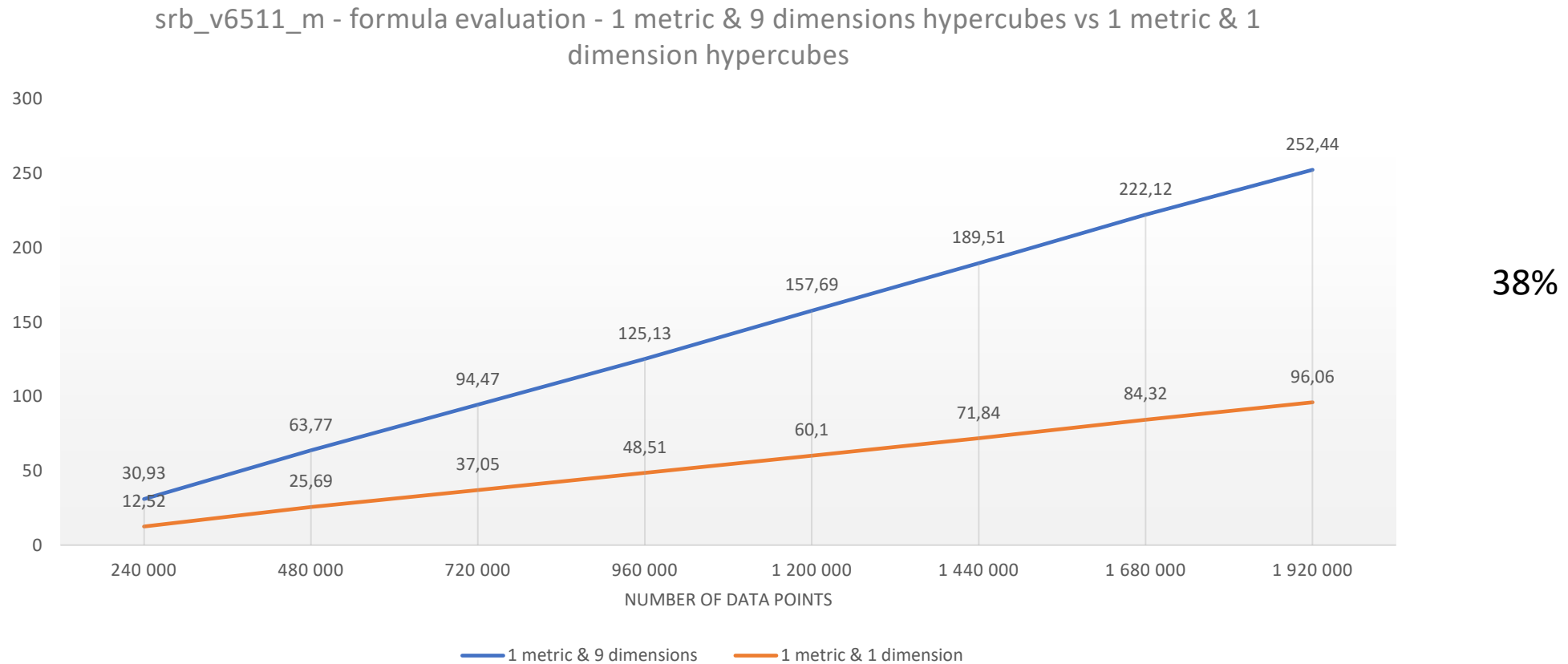
Impact on the filtering of the data

- The **OIM-CSV**, with its principle of **creating aliases** dissociating the data from its *"context"* information, has the same problem for **data filtering** when using the current modeling approach.

```
"dp418084": {
    "dimensions": {
        "concept": "eba_met:si599",
        "eba_dim:BAS": "eba_BA:x8",
        "eba_dim:MCY": "eba_MC:x29",
        "eba_dim:RPR": "eba_RP:x34"
    },
    "eba:documentation": {
        "logicalDatapointId": "418084",
        "templateId": "T 03.01",
        "SheetId": "n.a.",
        "rowId": "r999",
        "columnId": "c0040"
    }
},
```

```
"dp410400": {
    "dimensions": {
        "concept": "eba_met:si289",
        "eba_dim:MCK": "eba_MC:x29",
        "eba_dim:ROL": "eba_RP:x43",
        "eba_dim:RPR": "eba_RP:x34"
    },
    "eba:documentation": {
        "logicalDatapointId": "410400",
        "templateId": "T 03.01",
        "SheetId": "n.a.",
        "rowId": "r999",
        "columnId": "c0050"
    }
},
```

Agenda

- Data Point Modeling on dynamic tables
- Usage of data points vs Dimensions

Usage of data points vs Dimensions

- At the beginning of **Solvency II** DPM, **EIOPA** prepared 2 modeling approaches :
    - **Highly Dimensional DPM-based XBRL Taxonomies** : high quality of the model, explicit dependencies between concepts, ...
    - **Moderately Dimensional DPM-based XBRL Taxonomies** : concatenation of HD Metrics and those dimensions that are not necessary from table rendering perspective

- EIOPA asked the software vendors **which approach should be promoted:**
    - The **Moderately Dimensional approach has been chosen** by the great majority of the market, thinking it would be easier to manage fewer dimensions.

Usage of data points vs Dimensions

- The report « **S.06.02.01** - List of assets », DPM 2.4.0 EIOPA, is defined with **one metric at the column level** and **technical dimensions** at the **row level** as breakdowns

- From a **business validation and analysis** point of view this report contains **problems on filtering**

Usage of data points vs Dimensions

- Let's look at the assertion **s2md_BV784-5**:
  - Test: "*if (not(empty($a))) then iaf:numeric-equal($b, iaf:sum((iaf:numeric-multiply($a, $d), $e))) else (true())*"
  - Label: *BV784: The "Total Solvency II amount" in S.06.02 - List of assets should be equal to the product of "Par amount" and "Unit percentage of par amount Solvency II price" plus "Accrued interest". →table 1: S.06.02; Filter: not({c0290} like '##71' or {c0290} like '##9#' or {c0290} like '##09'); Expression: if {c0140} <> empty then {c0170} = {c0140} * {c0380} + {c0180}*

if c0290 <> '##71' or '##9#' or '##09' Then
c0170 = c0140 * c0380 + c0180

Usage of data points vs Dimensions

- The **complexity** of this assertion is linked to the **indirect filtering** of the variable **$f** (i.e. CIC codes must be different than '##71' or '##9#' or '##09') on the other variables of the assertion :

Select each data point C0290 different than ##71 or ##9# or ##09

Retrieve the **list** of values for the technical dimension **s2c_dim:UI** in their context

For each **value** in the list of dimension s2c_dim:UI

    Select each other data point : C0140, C0170, C0180 and C0380;

    Perform : if {c0140} <> empty then {c0170} = {c0140} * {c0380} + {c0180}.

Usage of data points vs Dimensions

- Thanks to a **modeling** using HD approach, i.e. considering the column **C0290 – CIC** as a **dimension** of the **other data points**, the **filtering will be direct**:

> **Select** the list of **context** with with **dimension** CIC different than **##71** or **##9#** or **##09**.

> **For each context** in the list
>> **Select** each data point : C0140, C0170, C0180 and C0380;
>>
>> **Perform** : if {c0140} <> empty then {c0170} = {c0140} * {c0380} + {c0180};

- In DPM 2.4.0, on **63 assertions** linked to the **report S.06.02**, **53 assertions** are using the **CIC** data points **as a filter** of the variables.

Usage of data points vs Dimensions

- In general we find the **same problem** on every modeling **using the MD approach**, for which **functional and/or technical dimensions** are used instead of using **"business" dimensions** to describe data points. Examples :
  - Solvency 2 – S0602: on **31 data points** at least **15** could have been business dimensions
  - LDR – T0301: on **24 data points** at least **10** could have been business dimensions

Usage of data points vs Dimensions

- How to **identify** the current data points which should have been **dimensions** ?

  - Basically each data point defined as "**enumerationItemType**" is a dimension;
  - Some data points like "**stringItemType**" (like CIC code) or "**dateItemType**" may be considered as dimensions.

Impact on the size of the instance documents

- There is a **minimal impact** in term of file size :

	Current DPM approach	The least dimensional possible approach	**With Enumerations as dimensions**	OIM-CSV, csv data file	**OIM-CSV, csv with enumeration as dimensions**
Instance size for 1 row	8 KB	4KB	**4 KB**	<1 KB	**<1 KB**
Instance size for 1 000 rows	5 578 KB	1 903 KB	**2 204 KB**	250 KB	**250 KB**
Instance size for 80 000 rows	452 453 KB	156 188 KB	**174 655 KB**	4 250 KB	**4 250 KB**

File size in KB per data points

— XBRL instance - 1 metric & 9 dimensions   — XBRL Instance - 1 metric & 1 dimension   — CSV (OIM)   — XBRL Instance - 1 metric & Enumerations as dimension
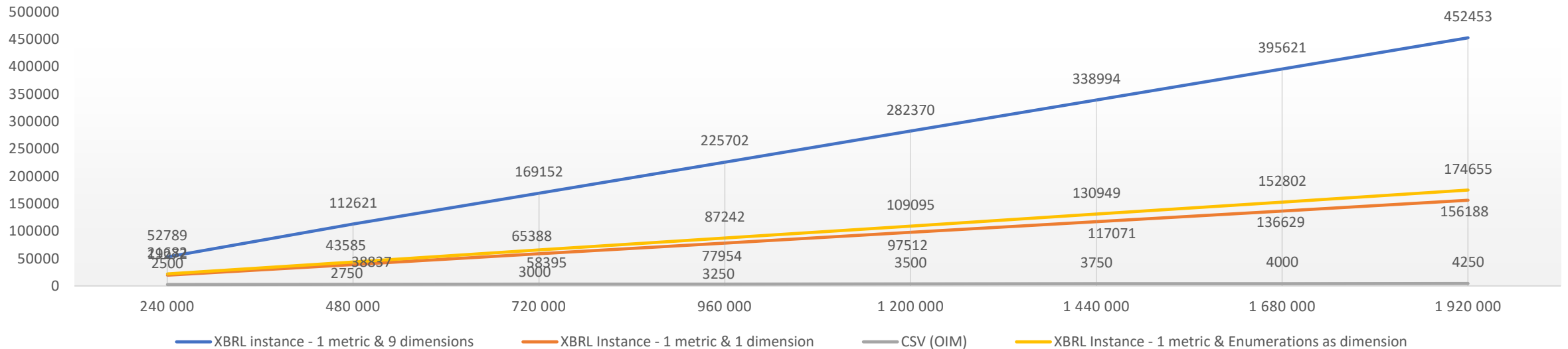
Impact on the filtering of the data

- There is **a minimal impact on the filtering** of the srb_v6511_m assertion because **all the data points share** the **same context** for a **row** :

srb_v6511_m - formula evaluation - 1 metric & 9 dimensions hypercubes vs 1 metric & 1 dimension hypercubes

Legend: 1 metric & 9 dimensions — 1 metric & 1 dimension — 1 metric & Enumerations as dimension

X-axis: NUMBER OF DATA POINTS

NUMBER OF DATA POINTS	1 metric & 9 dimensions	1 metric & 1 dimension	1 metric & Enumerations as dimension
240 000	30,93	11,52	14,62
480 000	63,77	28,8	25,69
720 000	94,47	37,05	41,74
960 000	125,13	48,51	54,06
1 200 000	157,69	60,1	65,97
1 440 000	189,51	71,84	79,1
1 680 000	222,12	84,32	90,84
1 920 000	252,44	96,06	101,46

Usage of data points vs Dimensions

- Thanks to the **OIM-CSV** we have a good opportunity to **reduce** the processing times regarding **data capture**.

- Even on XBRL (XML) instance, the **modeling approach** can already **enhance the performances and the usage of the resources** needed to perform the whole process.

- For the **validation and data analysis** we need to define **"business" dimensions**, share the **same context** for one row, in order to have a **direct data filtering** by **using dimensions** instead of data points.

Thank you for your attention !