CONVERTING XBRL INTO SDMX: A PRACTICAL APPROACH

ANTONIO OLLEROS,
CEO
MEANINGFUL DATA

Frankfurt, 19 June 2019

INTRODUCTION

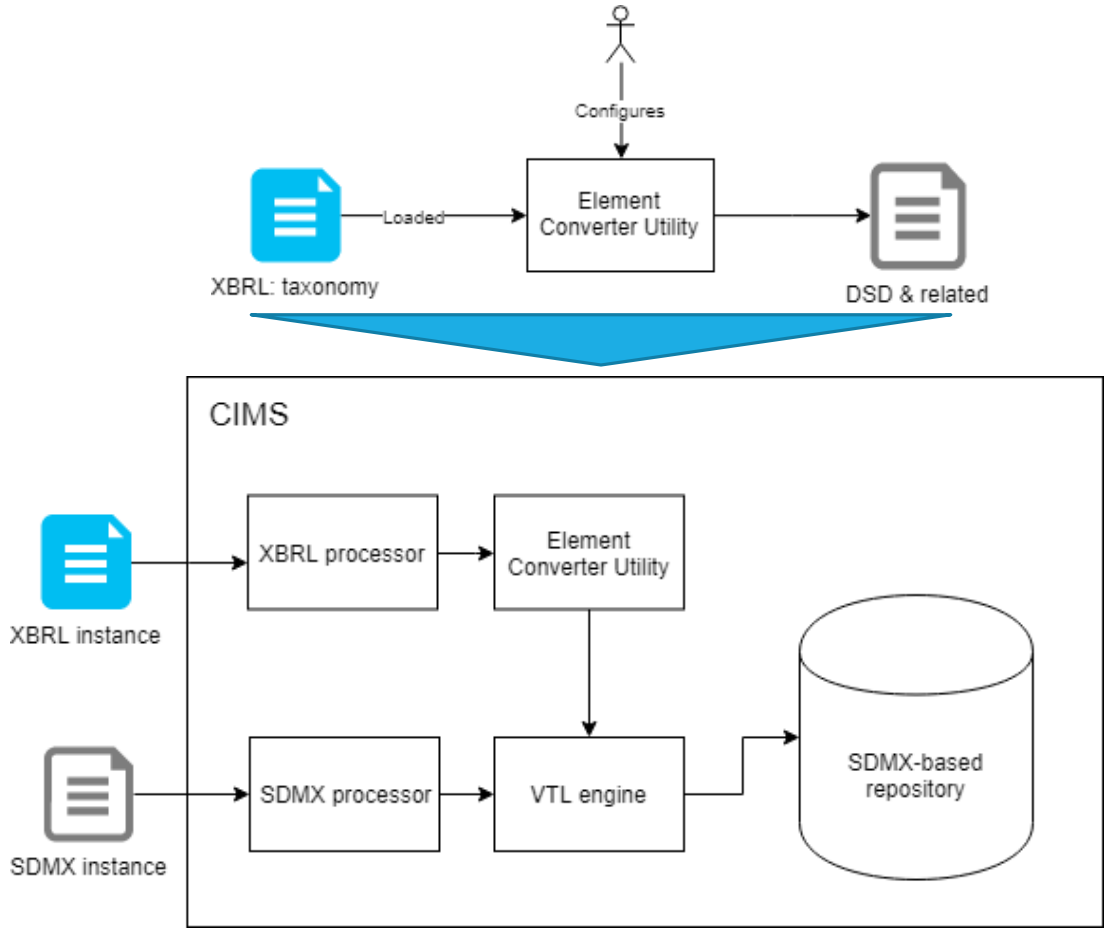
- This presentation shows the methodology used for an XBRL to SDMX converter to be built for the Reserve Bank of India (RBI)
- RBI has been successfully using XBRL since 2008. Some characteristics of RBI's system
  - 156 reports included
  - 10.000+ financial institutions
  - 200.000+ filings every year
  - System to system submission of filings
- RBI is starting the implementation of a new reporting system (CIMS)
  - Main reasons
    - Dealing with microdata
    - Growth of data requirements
    - Improving scalability
  - Using global standards, including XBRL, SDMX and VTL
  - Bid to implement the new system won by TCS, in a joint proposal with IRIS Business Services and meaningfulData

CIMS USE CASE

THE USE CASE

- RBI's new system will have an SDMX-based metadata-driven repository
- All reports to be defined using SDMX
- Flexibility for filers to use XBRL for the currently existing reports (temporary)
- Need to convert the XBRL input into SDMX, in order to store all the information in the repository
  - XBRL to SDMX conversion
  - Codes mappings
  - Other customisation
- For validation and transformation of data, Validation and Transformation Language (VTL) has been selected
  - XBRL validations are not automatically translated

HIGH-LEVEL DIAGRAM



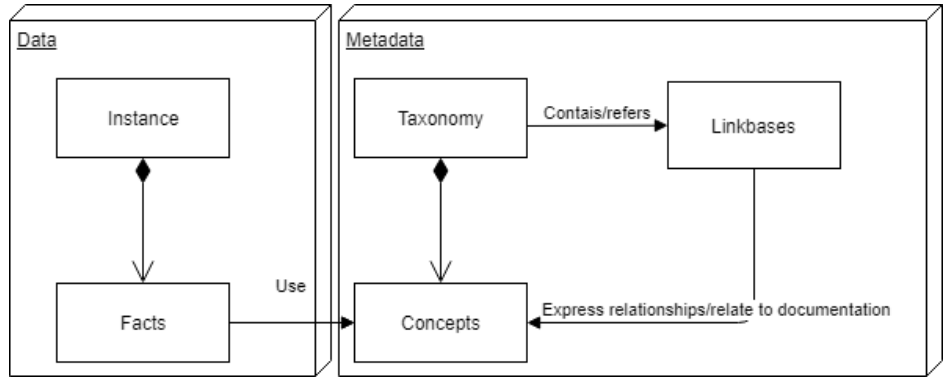
- As preliminary step , XBRL taxonomies are translated into SDMX DSDs and related artefacts
  - Includes parametrization of the mappings, like transcoding
- The system can take as input XBRL or SDMX files for a certain report
- If XBRL is sent:
  - XBRL instance is processed with the XBRL processor
  - If the file is correct (including basic validations), the file is sent to the ECU to convert it into an SDMX instance
  - The file goes through the VTL engine for final validations and transformations, if required
- If SDMX is sent:
  - SDMX instance is processed with the SDMX processor
  - If the file is formally correct, it is sent to the VTL engine for validations and transformations, if required

CONVERSION METHODOLOGY

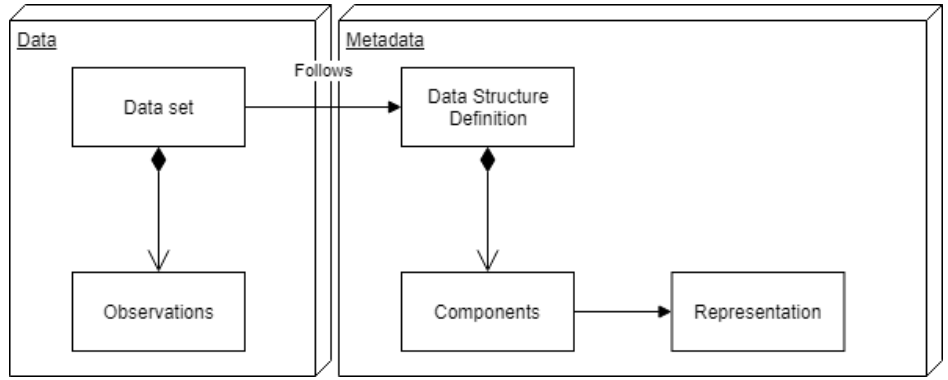
MOVING FROM A FACT-CENTRIC FORMAT TO A DATASET-CENTRIC FORMAT

- The basic building blocks of XBRL are facts, instance documents, concepts and taxonomies

  (taken from XBRL essentials from XBRL.org)



- In SDMX, the basic building block, as regards metadata, is the Data Structure Definition, which defines a structure that the datasets, containing observations, need to follow.



SEEING XBRL INSTANCES AS DATASETS

- An instance is a collection of facts. A fact has:
  - A concept
  - A context
    - Entity
    - Segment/Scenario: Collection of pairs dimension-member
    - Period
  - Decimals
  - Unit of reference
  - Observation

```
<tax1:Loans
    contextRef="20180331_ES_EUR"
    unitRef="EUR"
    decimals="0">157235
</tax1:Loans>
<tax1:Securities
    contextRef="20180331_ES_EUR"
    unitRef="EUR"
    decimals="0">217653
</tax1:Securities>
```

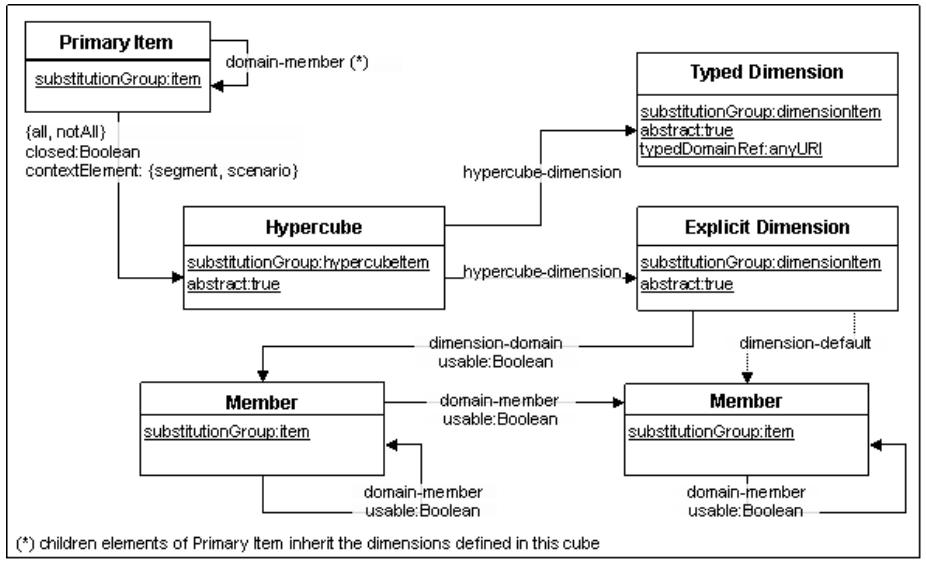
```
<xbrli:context id="20180331_ES_EUR">
 <xbrli:entity>
  <xbrli:identifier scheme="http://acme.org">018</xbrli:identifier>
  <xbrli:segment>
   <xbrldi:explicitMember dimension="tax1:locationOfActivities">tax1:ES</xbrldi:explicitMember>
   <xbrldi:explicitMember dimension="tax2:currencyOfInstrument">tax1:EUR</xbrldi:explicitMember>
  </xbrli:segment>
 </xbrli:entity>
 <xbrli:period>
  <xbrli:instant>2018-03-31</xbrli:instant>
 </xbrli:period>
</xbrli:context>
```

- And the collection of facts can be seen as a dataset

entity	period	primaryItem	locationOfActivities	currencyOfInstrument	observation	unitRef	decimals
018	2018-03-31	Loans	ES	EUR	157235	EUR	0
018	2018-03-31	Securities	ES	EUR	217653	EUR	0

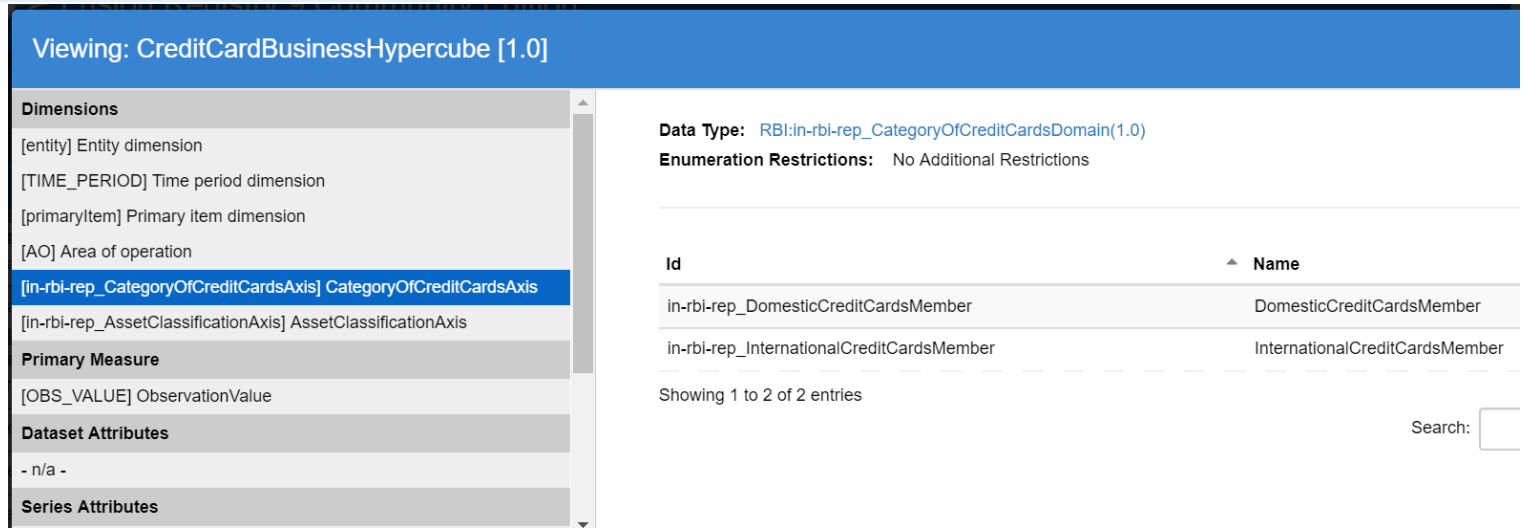
CONVERTING XBRL TAXONOMIES

- But how can we know from the taxonomy what is going to be the structure of the dataset?
- Hypercubes in an XBRL taxonomy define a kind of data structure, which can be assimilated to a DSD
- Primary items are considered another dimension, with a generic name
- Hypercubes define only some of the components of the DSD, other dimensions and attributes are implicit:
  - Entity
  - Period
  - Unit
  - Decimals



CONVERTING XBRL TAXONOMIES

Dimension Relationships	Arcrole
[2100] Credit Card Business	
Statement showing details of Credit card business carried by banks or through its subsidiai	
Credit card business [table]	all
Region of business [axis]	hypercube-dimension
Category of Credit cards [axis]	hypercube-dimension
Category of Credit cards [domain]	dimension-domain
Domestic Credit cards [member]	domain-member
International Credit cards [member]	domain-member
Asset classification [axis]	hypercube-dimension
Details of Credit card business [line items]	domain-member



CONVERTING INSTANCES

- In XBRL, instance files are a list of facts, with no assignment to any hypercube
- In SDMX, an instance file contains one to many datasets, each dataset having one to many observations
- So the key question is: How can we assign the facts to a hypercube?
  - Select the hypercubes containing the primary item of the fact
  - From the selection, select the hypercubes that use at least the same dimensions as the fact shows in the scenario/segment
  - From the selection, select the hypercubes for which the allowed domain for the dimensions fits the values of the context of the fact
  - If a fact uses a subset of dimensions of a hypercube, the missing dimensions need to be added with a default value

CONVERTING INSTANCES

```
<in-rbi-rep:AggregateNumberOfCardHoldersForCreditCardBusiness
    contextRef="fromto_20181001_20181231_DomesticCreditCardsMember_DomesticMember"
    unitRef="PURE"
    decimals="INF" >34</in-rbi-rep:AggregateNumberOfCardHoldersForCreditCardBusiness>
```

```
<xbrli:context id="fromto_20181001_20181231_DomesticCreditCardsMember_DomesticMember">
    <xbrli:entity>
        <xbrli:identifier scheme="http://www.rbi.gov.in/000/2010-12-31">
            041</xbrli:identifier>
        <xbrli:segment>
            <xbrldi:explicitMember
                dimension='in-rbi-rep:CategoryOfCreditCardsAxis'>
                in-rbi-rep:DomesticCreditCardsMember</xbrldi:explicitMember>
            <xbrldi:explicitMember
            dimension='in-rbi-rep:RegionOfBusinessAxis'>
            in-rbi-rep:DomesticMember</xbrldi:explicitMember>
        </xbrli:segment>
    </xbrli:entity>
    <xbrli:period>
        <xbrli:startDate>2018-10-01</xbrli:startDate>
        <xbrli:endDate>2018-12-31</xbrli:endDate>
    </xbrli:period>
</xbrli:context>
```

```
<message:DataSet action="Replace"
    structureRef="in-rbi-rep_CreditCardBusinessHypercube"
    validFromDate="2019-06-16T19:00:43">
    <generic:Series>
        <generic:SeriesKey>
            <generic:Value id="entity" value="041"/>
            <generic:Value id="in-rbi-rep_CategoryOfCreditCardsAxis"
                value="in-rbi-rep_DomesticCreditCardsMember"/>
            <generic:Value id="AO" value="D"/>
            <generic:Value id="primaryItem"
                value="in-rbi-rep_AggregateNumberOfCardHoldersForCreditCardBusiness"/>
        </generic:SeriesKey>
        <generic:Attributes>
            <generic:Value id="startDate" value="2018-10-01"/>
            <generic:Value id="decimals" value="INF"/>
            <generic:Value id="unitRef" value="xbrli:pure"/>
        </generic:Attributes>
        <generic:Obs>
            <generic:ObsValue id="OBS_VALUE" value="34"/>
            <generic:ObsDimension id="TIME_PERIOD" value="2018-12-31"/>
        </generic:Obs>
    </generic:Series>
```

CONCLUSION

- XBRL is a fact-centric format, while SDMX is a dataset-centric format
- The main challenge is how to translate XBRL structures to an SDMX dimensional data structure
- Hypercubes play a key role in the conversion, because they represent data structures
- There is no such a thing as a perfect one-fits-all mapping between XBRL and SDMX!! ➔ Importance of a use case