XBRL Notes

Sherif ElGamal

About me

Sherif Mahmoud ElGamal
selgamal@outlook.com

- CPA, CMA, CIA, CFE, EA
- Individual XBRL Certification
- Areas of expertise
  - Audit and advisory
  - Data analytics
  - Business Intelligence

Contents

1. Outcomes
2. Introduction
3. XBRL – What is it and how it works
4. XBRL Taxonomy Development

1 Outcomes

1. Outcomes

- Basic understanding the ecosystem of XBRL reporting, structured data and supporting technologies
- Basic understanding of XBRL and its components.
- Familiarity with core terminology.
- Basic understanding of XBRL Taxonomy and instance document.
- Basic understanding of the process of development of an XBRL taxonomy.

2 Introduction

2.1 Few Notes

- Terminology → English
- XBRL US *“Taxonomy Development Handbook” (“TDH”)*

> *In 2016, I was asked by XBRL US: "can we make a guide to help people and organizations build XBRL taxonomies?" This handbook is that guide, and it is the result of the efforts of many people and organizations as part of the XBRL US Domain Steering Committee over the past few years.*
> *--TDH Preface page 1*

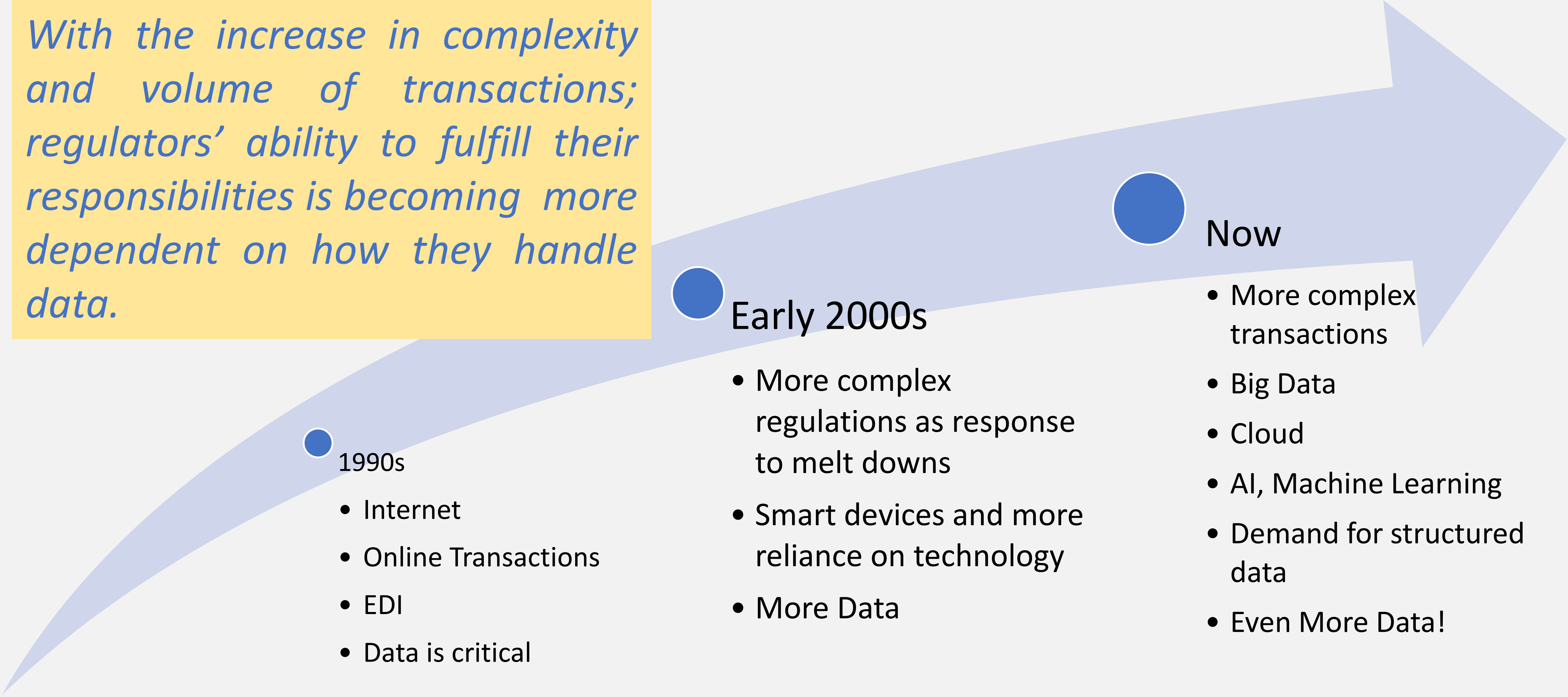
- Arelle Free and open-source software

> *Arelle is a project to provide the XBRL community with an easy-to-use open source platform for XBRL.*
> *-- Arelle website*

- Supporting material → https://git.io/J0dmE

2.2 Regulators and Data

With the increase in complexity and volume of transactions; regulators' ability to fulfill their responsibilities is becoming more dependent on how they handle data.



1990s

- Internet
- Online Transactions
- EDI
- Data is critical

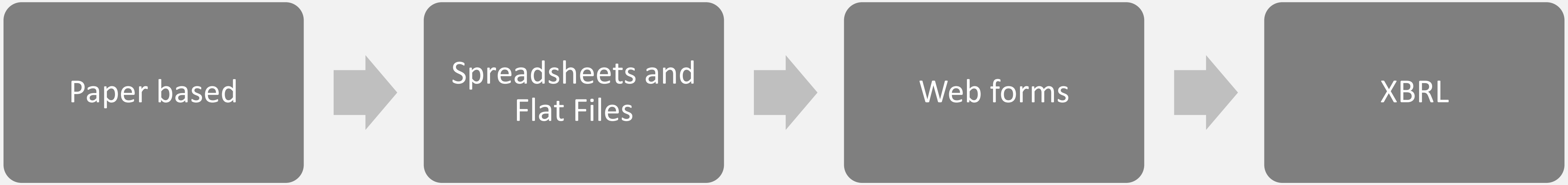
Early 2000s

- More complex regulations as response to melt downs
- Smart devices and more reliance on technology
- More Data

Now

- More complex transactions
- Big Data
- Cloud
- AI, Machine Learning
- Demand for structured data
- Even More Data!

2.3 Data Collection and Processing


Paper based

Spreadsheets and Flat Files

Web forms

XBRL

XBRL

eXtensible Business Reporting Language

Tagging Structured Financial Reporting Digital Financial Reporting Electronic Filing (E-Filing)

XBRL provides a language in which reporting terms can be authoritatively defined. Those terms can then be used to uniquely represent the contents of financial statements or other kinds of compliance, performance and business reports. XBRL lets reporting information move between organizations rapidly, accurately and digitally.

-- *XBRL.org*

2.4 What does XBRL provide?

Machine-Readable

- Automation
- Validation
- Analytics
- Time and cost savings

Interoperability

- XBRL is system independent

Unambiguous, reusable definitions

- Standardized reporting
- Flexibility
- Supports different views
- Common set of rule, common language.

Multidimensional

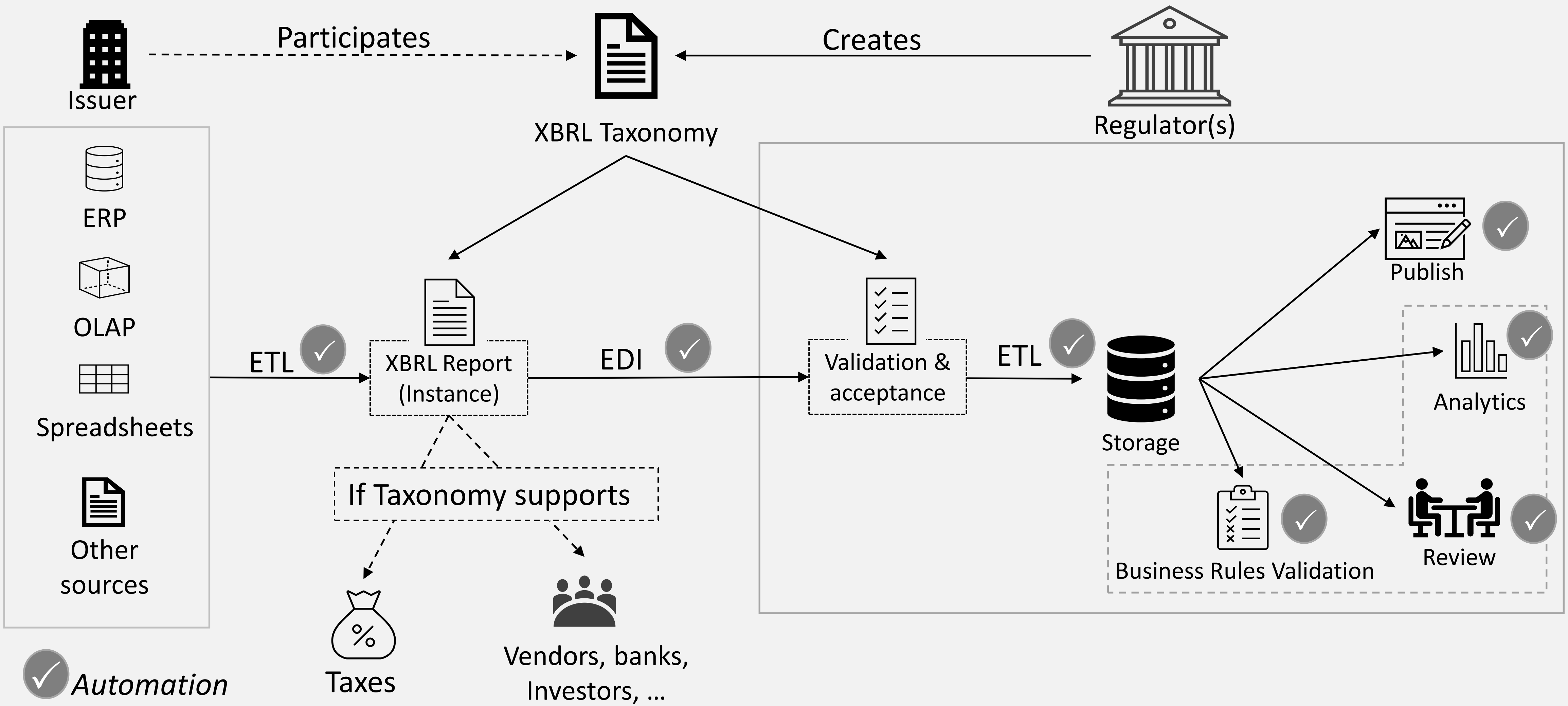
- Expresses complex information adequately

- Separates data content from form, same reported dataset can be filtered or viewed from different perspectives.
- Stable *structure* while maintaining *flexibility* of data content
  - ➢ Changes to “XBRL Reporting Taxonomy” can be easily implemented without any changes to the reporting system.

Short definitions

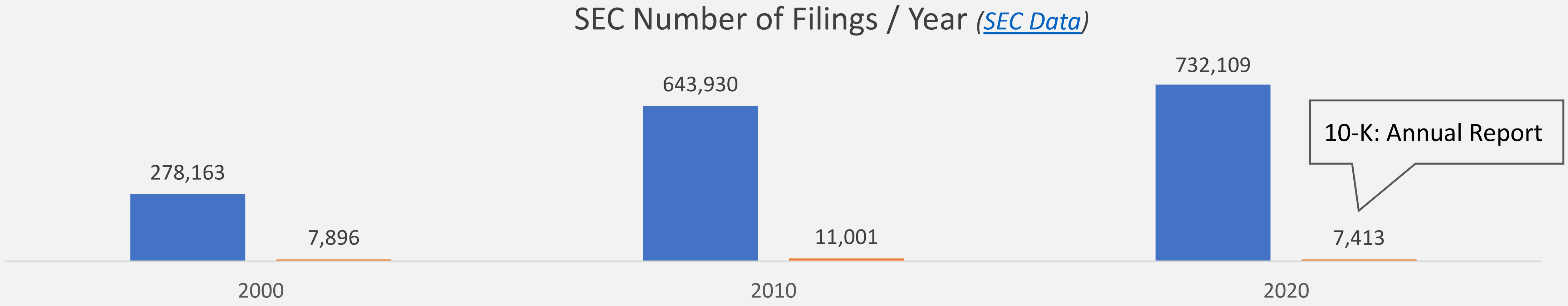
- **XBRL Taxonomy:** A collection of vocabulary (concepts), definitions and relationships relating to a specific domain.
- **XBRL Instance:** XBRL report created based on XBRL Taxonomy
- **XBRL Fact:** Individual piece of information in XBRL report (instance)

2.5 XBRL Reporting Overview



2.6 SEC – Improving reporting process

US Securities and Exchange Commission (SEC): Government agency responsible for regulating securities markets



SEC Number of Filings / Year (*SEC Data*)

10-K: Annual Report

Number of values reported in selected 10-K form for 2020 (Balances, Narrative disclosures, …):

FB	1,061	AAPL	1,388	AMZN	1,445	GOOGL	1,621
MSFT	1,913	BAC	7,853	JPM	8,970	AIG	11,418

2.6 SEC – Improving reporting process (cont.)

In 2009 SEC adopted the “Interactive Data to Improve Financial Reporting” rule to phase in the use of XBRL for SEC submissions made by its registrants.

Some of the results:

- Easier adoption to new taxonomies and reporting requirements by issuers.
- Data aggregators reported reduction of processing times from 20-30 mins to 1-2 mins per submission.

Source TDH Section 10.4.3 page 141

2.7 Banking in the United States

Approximately 8,200 bank institutions in the US are required to report quarterly financial information to the Federal Financial Institutions Examination Council (FFIEC) on standardized forms called "Call Reports". In 2005 FFIEC implemented XBRL for the call reports.

Some of the results:

- 95% of initial filings met validation requirements vs 66% in legacy systems.
- 100% mathematical accuracy vs 30%.
- Data received within one day of the end of the reporting period vs several weeks.
- Analysts could handle 550 to 600 banks, versus 450 to 500.

Source FFIEC publication Improved Business Process Through XBRL

2.8 Business to Government Reporting

Australia, Netherlands, among other countries developed programs based on the XBRL called Standard Business Reporting (SBR) that harmonizes the definitions used in reporting. Reports in SBR are machine-readable, consistent, clearly defined, and agreed upon by all members of the reporting supply chain.

Some of the results:

- Businesses spend less time collecting data, filling in forms.
- SBR reduces cost and fosters greater efficiency for government agencies.
- Australian Taxation Authority report for 2017-2018 reported annual (recurring) savings were estimated at $1.45 billion AUD from SBR

Source TDH Section 10.2 page 137

2.9 Who is using XBRL?

Securities and Exchange Commission

FDIC

U.S. DEPARTMENT OF ENERGY

Companies House

HM Revenue & Customs

BANK OF ENGLAND

SBR Standard Business Reporting

An Australian Government Initiative

SBR STANDARD BUSINESS REPORTING

تداول Tadawul

وزارة التجارة والصناعة
Ministry of Commerce and Industry

هيئة الأوراق المالية والسلع
Securities & Commodities Authority
الإمارات العربية المتحدة

هيئة قطر للأسواق المالية
Qatar Financial Markets Authority

TOKYO STOCK EXCHANGE

EBA EUROPEAN BANKING AUTHORITY

eiopa European Insurance and Occupational Pensions Authority

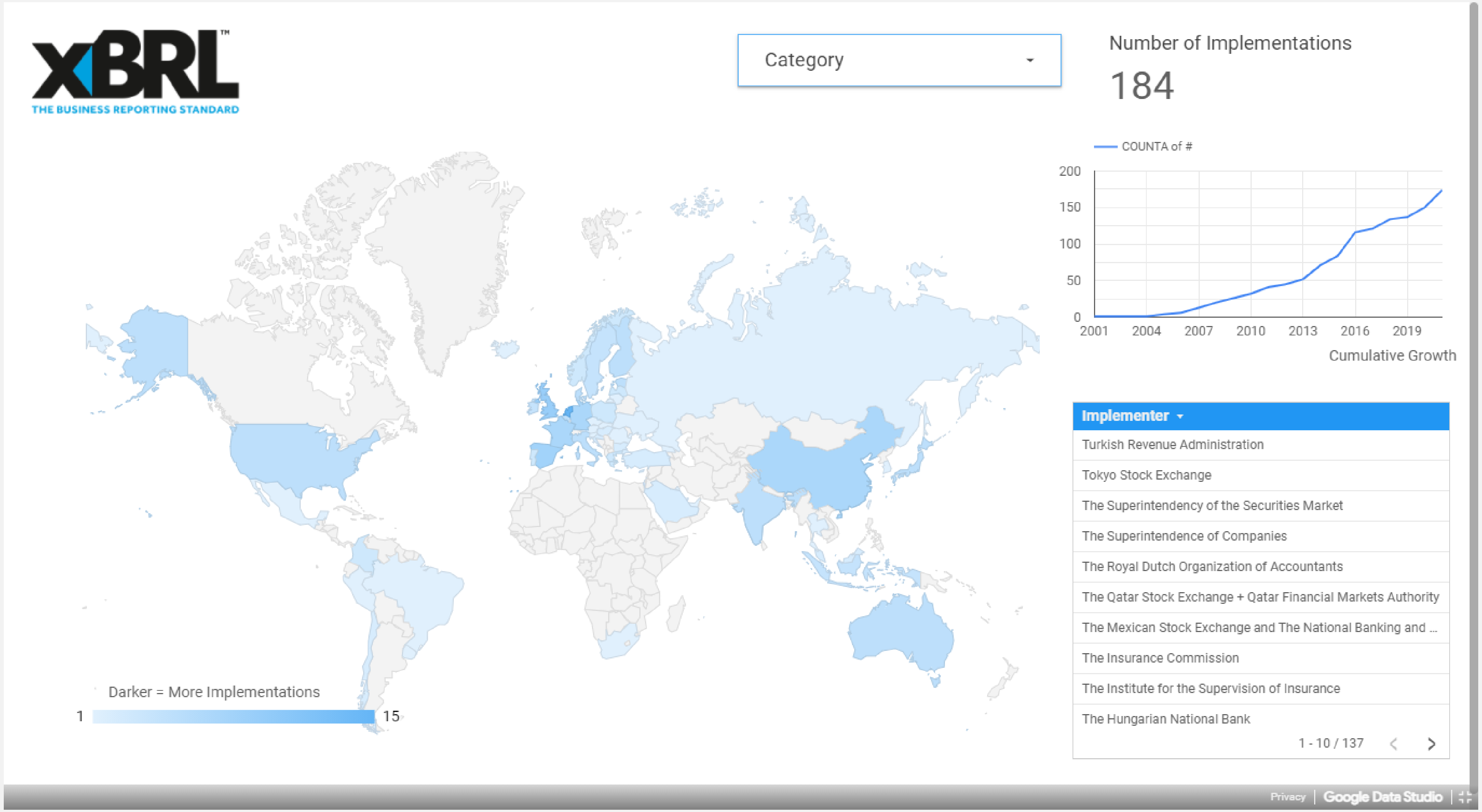
esma European Securities and Markets Authority

[Tope 10 countries with open data]

[Securities Regulators]

[Business Registrars]

2.10 XBRL Implementation Map



Number of Implementations

184

Implementer ▾
Turkish Revenue Administration
Tokyo Stock Exchange
The Superintendency of the Securities Market
The Superintendence of Companies
The Royal Dutch Organization of Accountants
The Qatar Stock Exchange + Qatar Financial Markets Authority
The Mexican Stock Exchange and The National Banking and ...
The Insurance Commission
The Institute for the Supervision of Insurance
The Hungarian National Bank

1 - 10 / 137

3 XBRL – How it works

3.1 What is XBRL

Specifications

- Based on XML
- Developed and maintained by XBRL international
- Stable since 2003
- No Licensing required

Data Model

- XBRL provides tools that enables the definition of taxonomy data model and reports

Communication Language

- XBRL enables exchange of structured financial data between systems (“Transport Model”)

- XBRL enabled software may have licensing fees.
- *“A Transport Model serves as an organizational structure when moving data from a source to a consumer”* — TDH section 2.1.2 page 10

3.2 XML and Markup Languages

3.2.1 Back in time

In ancient Egyptian writings, royal names were encapsulated in an oval shape called “Cartouche”.

The ancient Egyptians choose this method of marking or “tagging” to convey additional information about the written text. Markup languages do the same but, in a machine-readable format.

Markup languages in general do the same thing, the content of a file or a document is tagged in a way that makes it machine readable, the tags tell the computer what to do with the content.

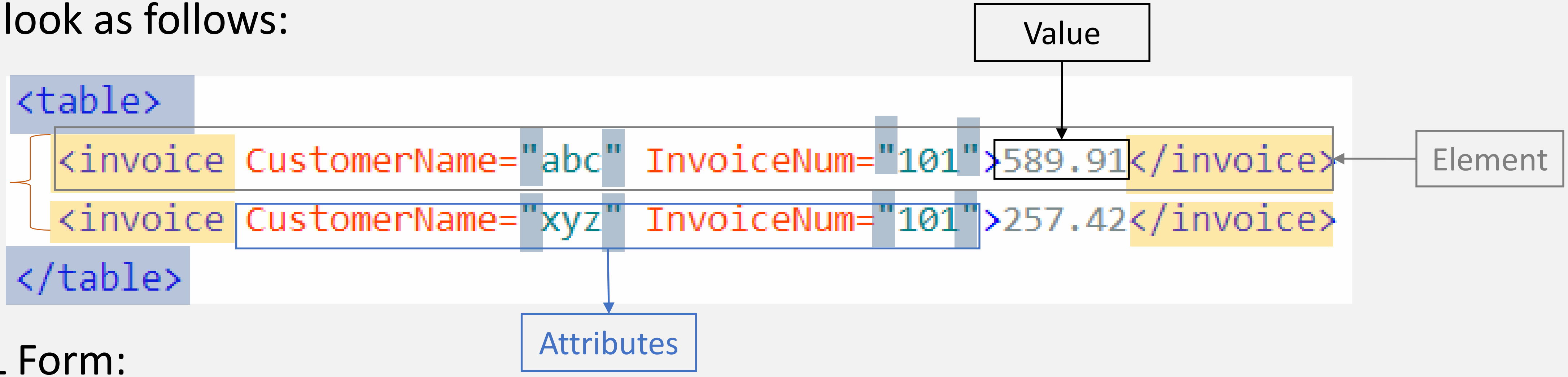
XML (eXtensible Markup Language) is a simple, text-based, system independent, format for representing structured information, it is one of the most widely-used formats for sharing structured information between programs, people, between computers and people, both locally and across networks.

Image by Osama Shukir Muhammed Amin FRCP (Glasg)

3.2.2 XML Basics

XML is system independent, and all operating systems and programing languages have tools to work with XML.

If we want to represent a table including information about customer invoices, it may look as follows:



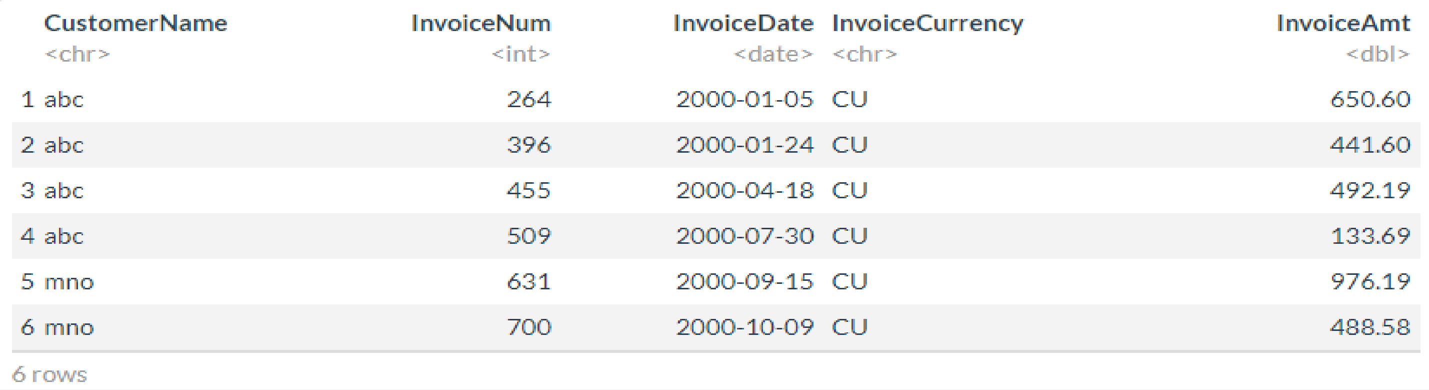
```
<table>
  <invoice CustomerName="abc" InvoiceNum="101">589.91</invoice>
  <invoice CustomerName="xyz" InvoiceNum="101">257.42</invoice>
</table>
```

XML Form:

- One root element
- Opening and closing tag
- Properly nested
- Attributes must be quoted

3.2.3 Storing Data in XML

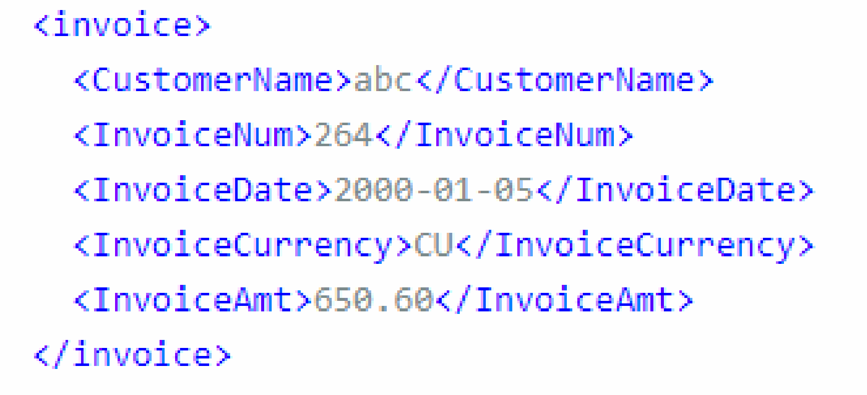
Assuming we have a table containing data about customers invoices and we want to model and store it in XML



	CustomerName <chr>	InvoiceNum <int>	InvoiceDate <date>	InvoiceCurrency <chr>	InvoiceAmt <dbl>
1	abc	264	2000-01-05	CU	650.60
2	abc	396	2000-01-24	CU	441.60
3	abc	455	2000-04-18	CU	492.19
4	abc	509	2000-07-30	CU	133.69
5	mno	631	2000-09-15	CU	976.19
6	mno	700	2000-10-09	CU	488.58

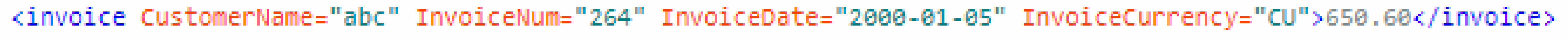
6 rows

We can store each piece of information as a value



```
<invoice>
  <CustomerName>abc</CustomerName>
  <InvoiceNum>264</InvoiceNum>
  <InvoiceDate>2000-01-05</InvoiceDate>
  <InvoiceCurrency>CU</InvoiceCurrency>
  <InvoiceAmt>650.60</InvoiceAmt>
</invoice>
```

It may be better to utilize XML attributes and focus the value on the invoice amount amount



```
<invoice CustomerName="abc" InvoiceNum="264" InvoiceDate="2000-01-05" InvoiceCurrency="CU">650.60</invoice>
```

Full example

3.2.4 Schema, Namespaces and Validation

XML is used to store structured information, **Schema** and **Namespaces** are the XML mechanisms to provide a design for this structure.

XML Schema is a component of XML (W3C recommendation) used to describe and validate elements in an XML document. Schema can be described as the blueprint of vocabulary and attributes, and what data is stored in an XML file and how. **Schema contains definitions of elements and types and derivations** allowed to be used in the XML file.

Namespaces is a component of XML (W3C recommendation) used to group and provide uniquely named elements in an XML document. XML document may contain elements from multiple vocabularies (schema), namespaces help in uniquely identifying elements from different vocabularies having identical names.

XML validation is the process of checking if an XML document is well formed, and valid against an XML schema

Supporting materials and validation example

3.2.5 XLink and XPointer

XLink and XPointer are XML components that can be used to link XML entities with each other within XML or to external resources.

XLink

“XML Linking Language (XLink) Version 1.1, which allows elements to be inserted into XML documents in order to create and describe links between resources. It uses XML syntax to create structures that can describe links similar to the simple unidirectional hyperlinks of today’s HTML, as well as more sophisticated links.”

— *W3.org XLink recommendation*

XPointer is a W3C recommendation is a construct that allows for locating specific fragment within XML.

Simple Links: Creates a unidirectional hyperlink from one element to another through a URI.

Extended Links: Provide for multiple resources at the source or destination to be connected via multiple arcs. An arc contains information about the origin, destination, and the behavior of a link between two resources.

Supporting materials

3.2.5 Conclusion

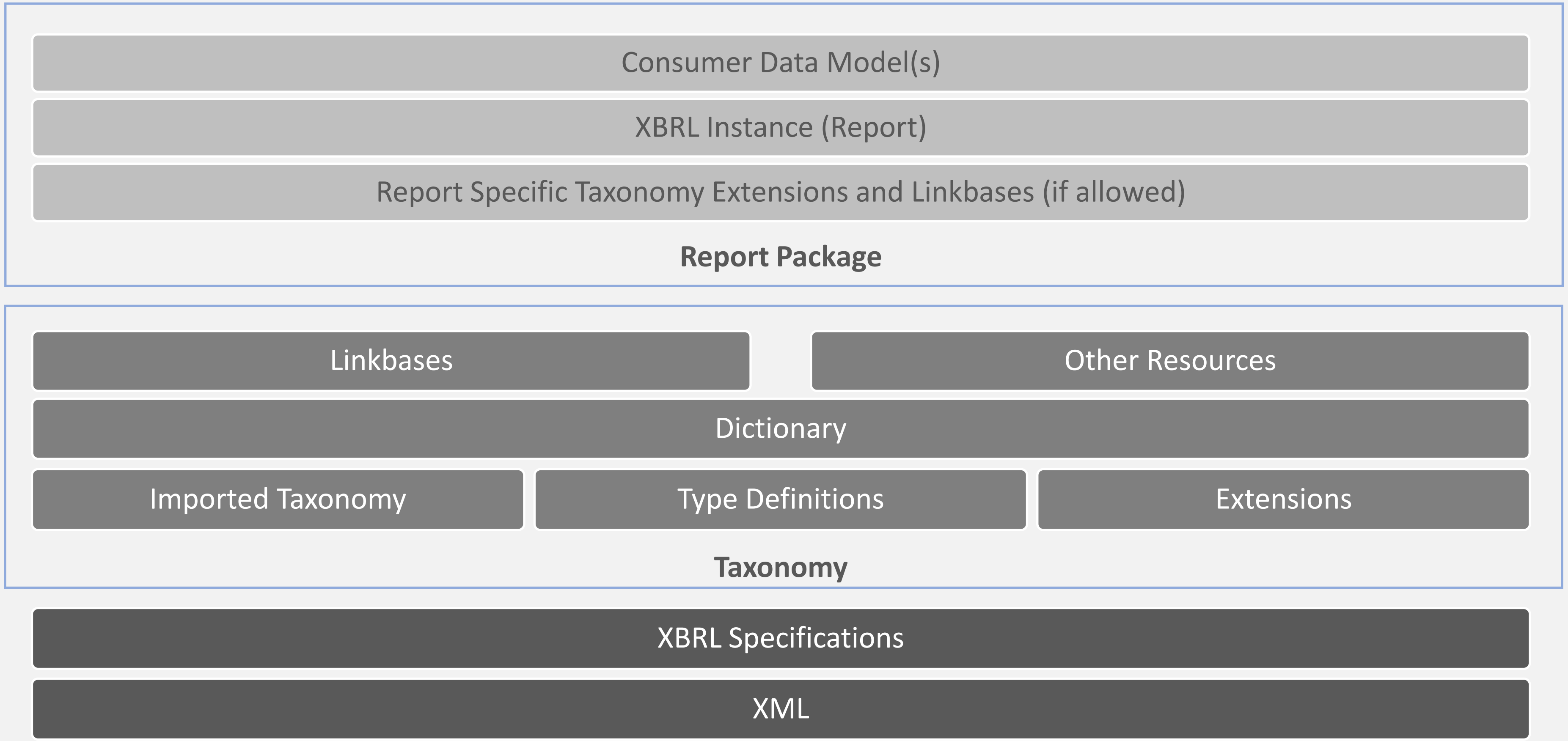
XML provides for:

- Flexibility in data modeling and data types definitions
- Mechanisms for creating vocabularies
- Mechanisms for validation
- Mechanisms for linking internal and external components

XBRL benefits from all the above tools that XML provides to achieve its objectives of being the standard language for machine-readable financial reports

3.3 XBRL

3.3.1 XBRL Components



Consumer Data Model(s)

XBRL Instance (Report)

Report Specific Taxonomy Extensions and Linkbases (if allowed)

Report Package

Linkbases

Other Resources

Dictionary

Imported Taxonomy

Type Definitions

Extensions

Taxonomy

XBRL Specifications

XML

Supporting materials

3.3.2 XBRL Specifications

XBRL Specifications are the set of technical standards and rules that makes up XBRL summarized as follows:

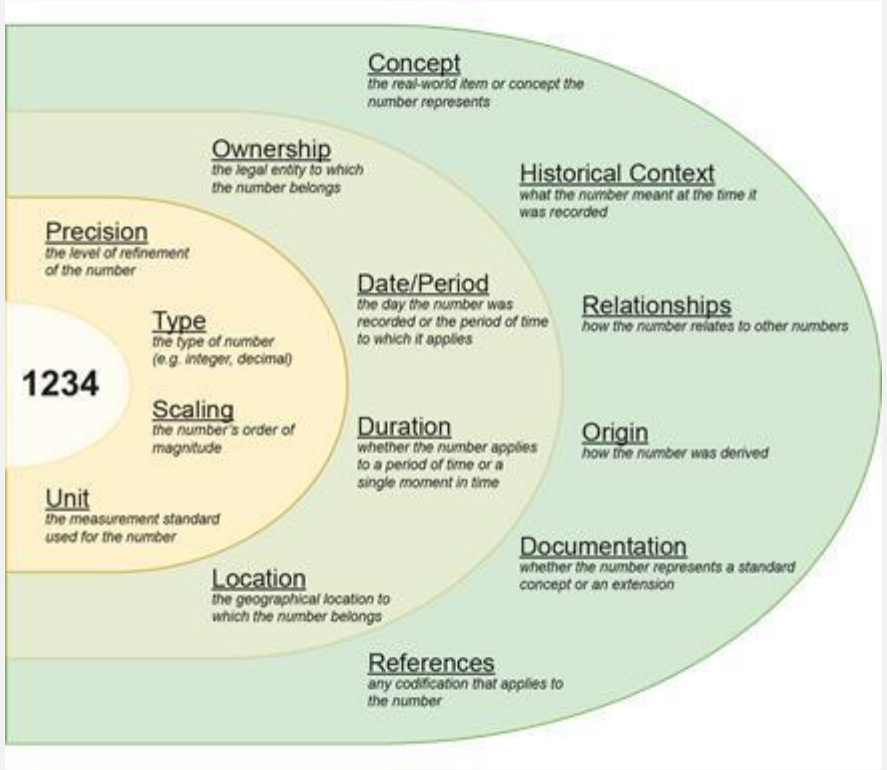
- XBRL
- Dimensions
- Extensible Enumerations
- Formula
- Generic Links
- Generic Preferred Label
- Global Ledger
- Infrastructure
- Inline XBRL
- Registries
- Table Linkbase
- Taxonomy & Report Packages
- Versioning

Supporting materials

3.3.2 XBRL Representation of Data

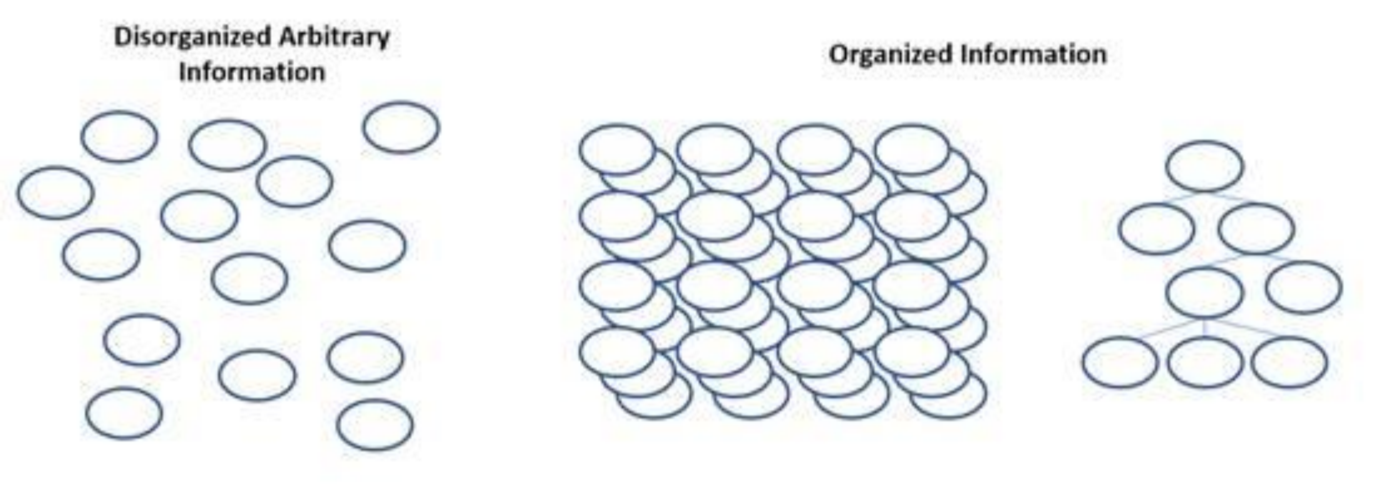
XBRL Provides a Platform to Give Data Meaning

TDH describes XBRL as a platform to give data meaning [Section 1.3.1]. A piece of data does not have a meaning without a context or means to associate it with other data points.



Concept
the real-world item or concept the number represents
Ownership
the legal entity to which the number belongs
Historical Context
what the number meant at the time it was recorded
Precision
the level of refinement of the number
Date/Period
the day the number was recorded or the period of time to which it applies
Relationships
how the number relates to other numbers
Type
the type of number (e.g. integer, decimal)
1234
Scaling
the number's order of magnitude
Duration
whether the number applies to a period of time or a single moment in time
Origin
how the number was derived
Unit
the measurement standard used for the number
Documentation
whether the number represents a standard concept or an extension
Location
the geographical location to which the number belongs
References
any codification that applies to the number

Possible types of information that could accompany a simple numeric data point

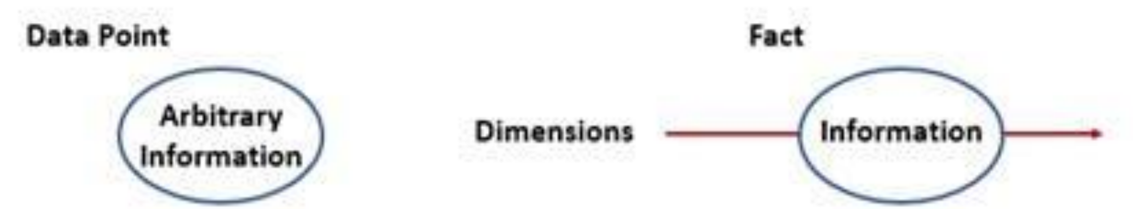


Disorganized versus organized data

3.3.2 XBRL Representation of Data (Cont.)

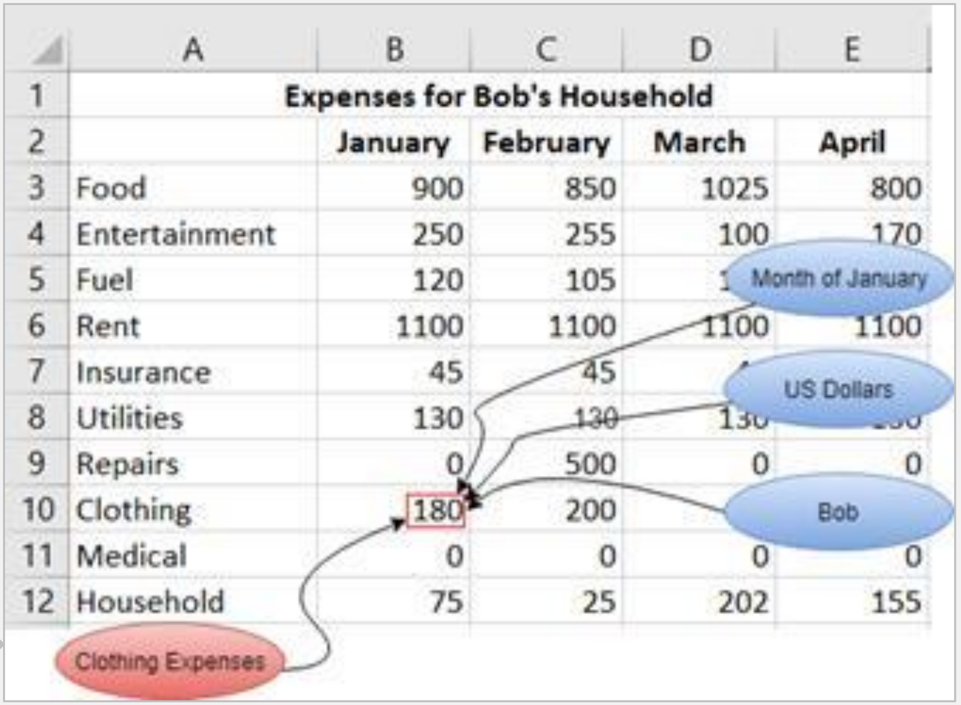
Expenses Example

TDH provides and example showing monthly expenses of an individual named "Bob" [Section 2.2] That explains the that XBRL "Fact" is the intersection of dimensions and data.



A data point versus an XBRL fact

The expense example with an illustrated XBRL fact

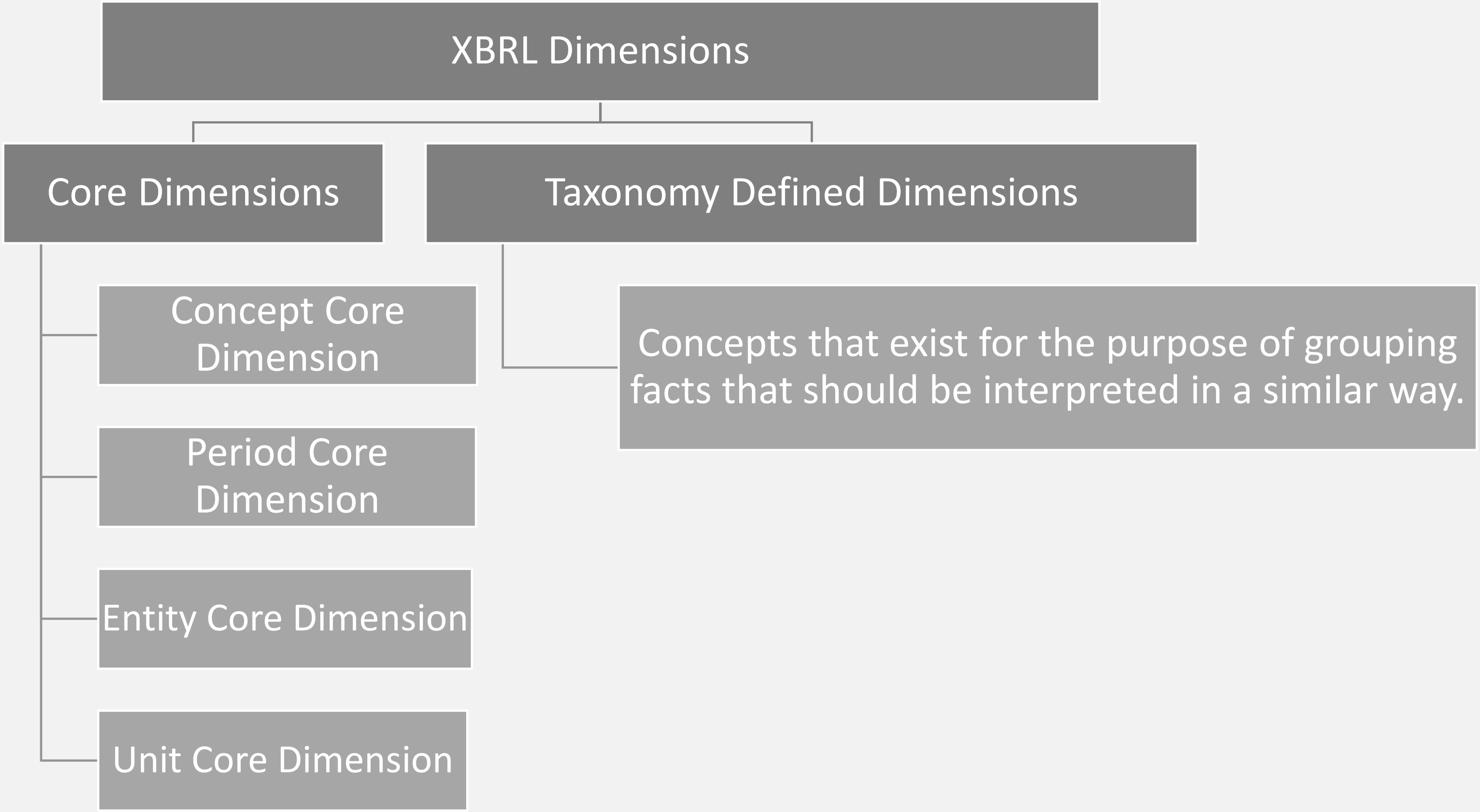


	A	B	C	D	E
1	Expenses for Bob's Household				
2		January	February	March	April
3	Food	900	850	1025	800
4	Entertainment	250	255	100	170
5	Fuel	120	105	[illegible]	[illegible]
6	Rent	1100	1100	1100	1100
7	Insurance	45	45	[illegible]	[illegible]
8	Utilities	130	130	130	[illegible]
9	Repairs	0	500	0	0
10	Clothing	180	200	[illegible]	[illegible]
11	Medical	0	0	0	0
12	Household	75	25	202	155

One of the basic concepts of XBRL design is that it identifies data points by multiple dimensions that gives enough context to the data point to be meaningful

3.3.3 XBRL Dimensions

The TDH Classifies Dimensions as follows:



XBRL Dimensions

Core Dimensions

Taxonomy Defined Dimensions

Concept Core Dimension

Period Core Dimension

Entity Core Dimension

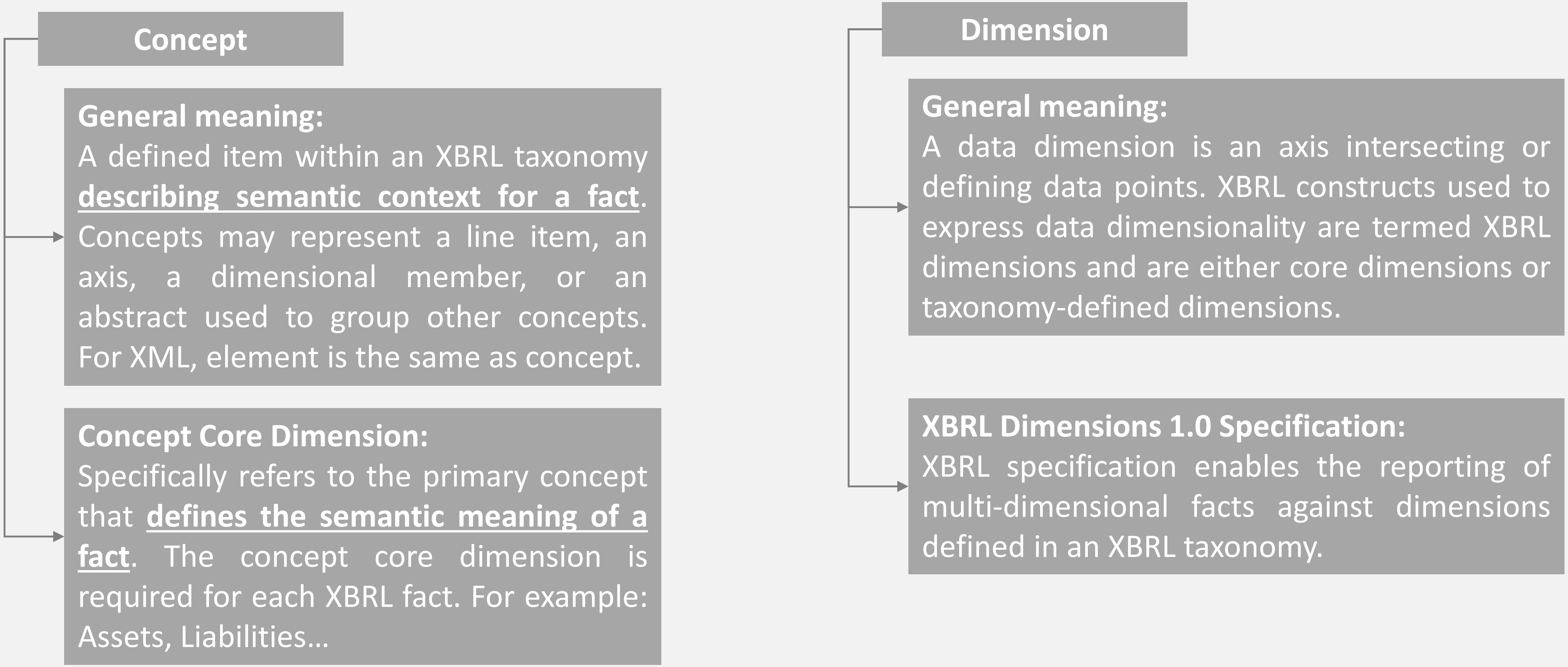
Unit Core Dimension

Concepts that exist for the purpose of grouping facts that should be interpreted in a similar way.

Supporting materials

3.3.4 Terms Concept and Dimension

Terms Concept and Dimension in XBRL may have different meaning depending on the context they are used in; this can be summarized as follows:



Concept

General meaning:
A defined item within an XBRL taxonomy **describing semantic context for a fact**. Concepts may represent a line item, an axis, a dimensional member, or an abstract used to group other concepts. For XML, element is the same as concept.

Concept Core Dimension:
Specifically refers to the primary concept that **defines the semantic meaning of a fact**. The concept core dimension is required for each XBRL fact. For example: Assets, Liabilities…

Dimension

General meaning:
A data dimension is an axis intersecting or defining data points. XBRL constructs used to express data dimensionality are termed XBRL dimensions and are either core dimensions or taxonomy-defined dimensions.

XBRL Dimensions 1.0 Specification:
XBRL specification enables the reporting of multi-dimensional facts against dimensions defined in an XBRL taxonomy.

3.3.5 Core Concept Dimension

Concept core dimension is the building block for XBRL taxonomy, a concept needs to be created for every reportable element. XBRL Concepts can be either in the ***"Item" or "Tuple" substitution group***

Substitution group: *is a feature of XML schema that allows defining elements that can replace another element in documents generated from that schema (W3School Example).*

Item: *XBRL technical term for a type of element, an Item represents a single fact or business measurement.*

Tuple: *XBRL technical term for a type of element. Tuple type is used for expressing an XBRL fact such that the fact is comprised of two or more data point pairs. Tuples are rarely used in XBRL taxonomies.*

3.3.6 XBRL Data Types

XBRL Concept is either in item or tuple substitution groups, a data type must be given to the concept that determines what type of data can be stored in this element (for example, numbers, dates, strings, … etc.), XBRL uses XML standard types in addition to other derived types, TDH table 2-3 page 28 shows the most common types used in XBRL:

dataType	description
stringItemType	Represents character strings in XML.
booleanItemType	Represents the values of two-valued logic (true, false).
decimalItemType	Represents a subset of real numbers, which can be represented by decimal numerals.
dateTimeItemType	Represents instants of time, optionally marked with a time zone offset.
integerItemType	Represents the standard mathematical concept of integer numbers by fixing the fractional digits of decimal to be 0 and prohibiting the trailing decimal point.
monetaryItemType	Represents a decimal with the added constraint of a currency unit.
qNameItemType	Represents a qualified XML name.

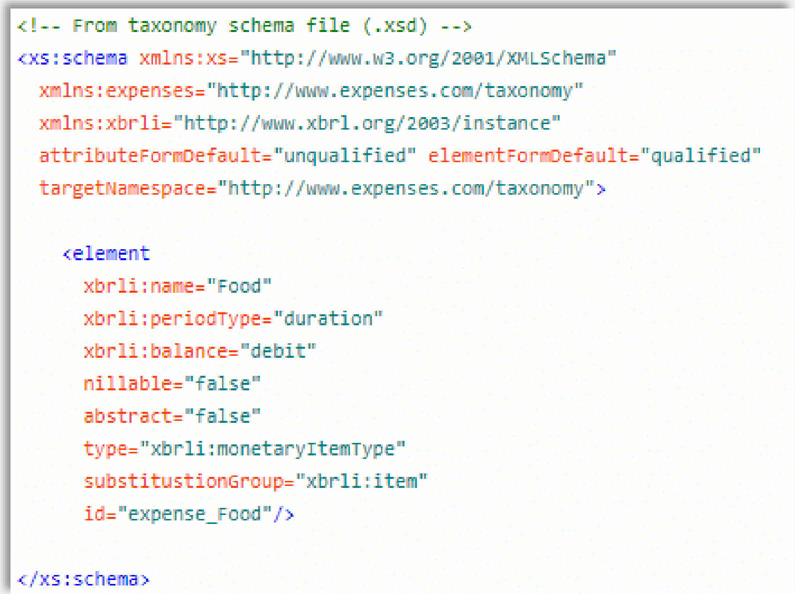
This is just a sample of the most used data types, XBRL Specifications [Section 5.1.1.3] lists more data types, also other XBRL data types are defined in XBRL Data Types Registry.

3.3.7 Creating Concept Core Dimension

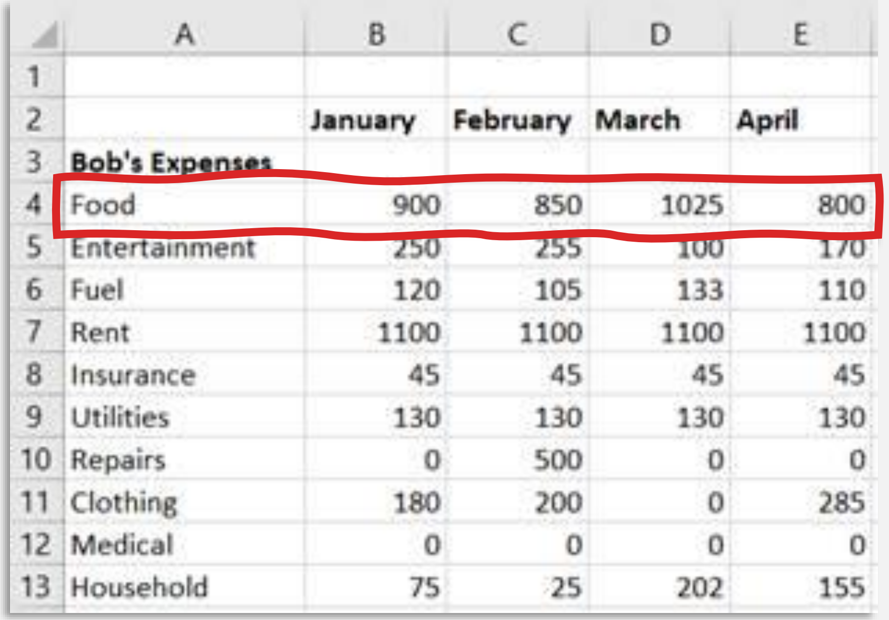
Assuming we want to create a concept for the **“Food”** Expenses

The Concept should have the following characteristics:

- Has a debit balance.
- Its value Cannot be null (absent value), it can have a value of 0 though.
- It is a monetary item, meaning that it needs to have a numeric value and a unit.



```
<!-- From taxonomy schema file (.xsd) -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:expenses="http://www.expenses.com/taxonomy"
  xmlns:xbrli="http://www.xbrl.org/2003/instance"
  attributeFormDefault="unqualified" elementFormDefault="qualified"
  targetNamespace="http://www.expenses.com/taxonomy">

    <element
      xbrli:name="Food"
      xbrli:periodType="duration"
      xbrli:balance="debit"
      nillable="false"
      abstract="false"
      type="xbrli:monetaryItemType"
      substitustionGroup="xbrli:item"
      id="expense_Food"/>

</xs:schema>
```



	A	B	C	D	E
1					
2		January	February	March	April
3	Bob's Expenses				
4	Food	900	850	1025	800
5	Entertainment	250	255	100	170
6	Fuel	120	105	133	110
7	Rent	1100	1100	1100	1100
8	Insurance	45	45	45	45
9	Utilities	130	130	130	130
10	Repairs	0	500	0	0
11	Clothing	180	200	0	285
12	Medical	0	0	0	0
13	Household	75	25	202	155

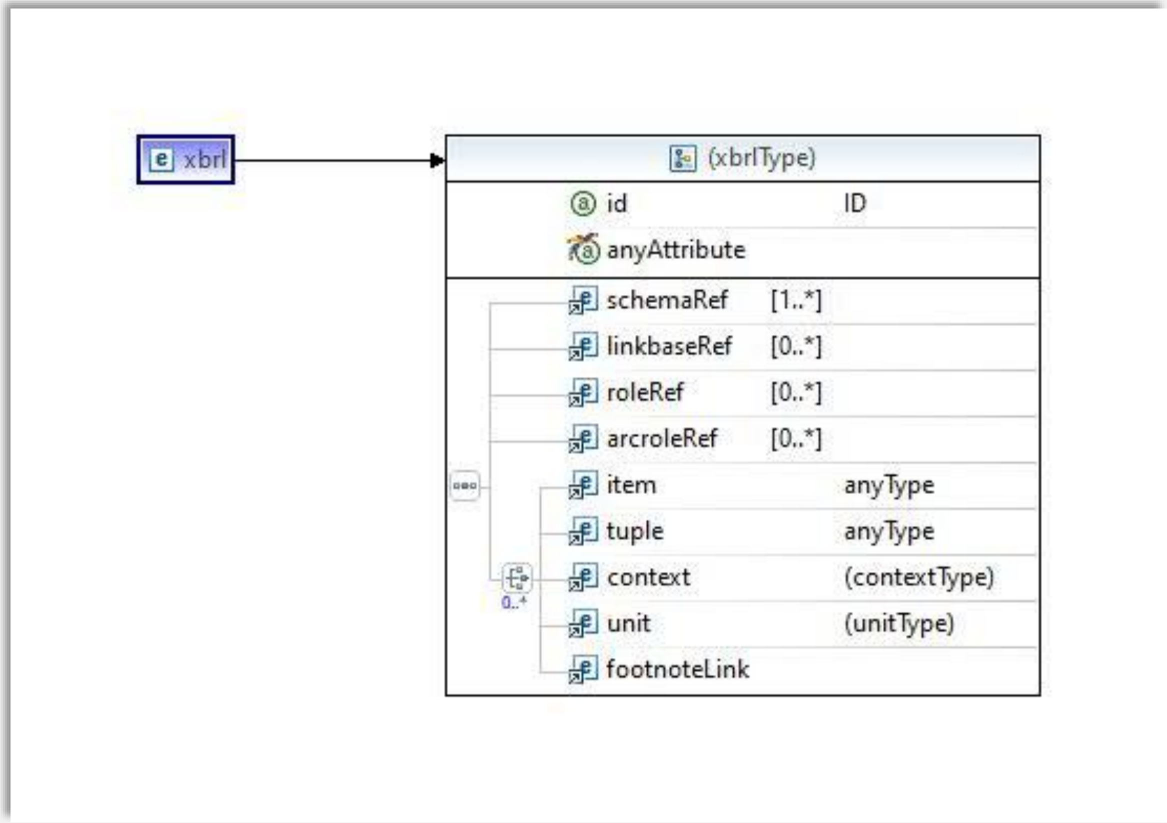
simple expense example

See supporting materials

3.3.8 XBRL Instance Document

Now that we created a concept we can create a report, in XBRL the report is called an Instance meaning it is an instance of the taxonomy.

Instance document is the actual report containing the facts and information for the report, it is constructed using XBRL constructs. The root element of XBRL instance document is <xbrl>, and it is defined in XBRL specification in the namespace {http://www.xbrl.org/2003/instance}, XBRL element is defined as follows:



See supporting materials

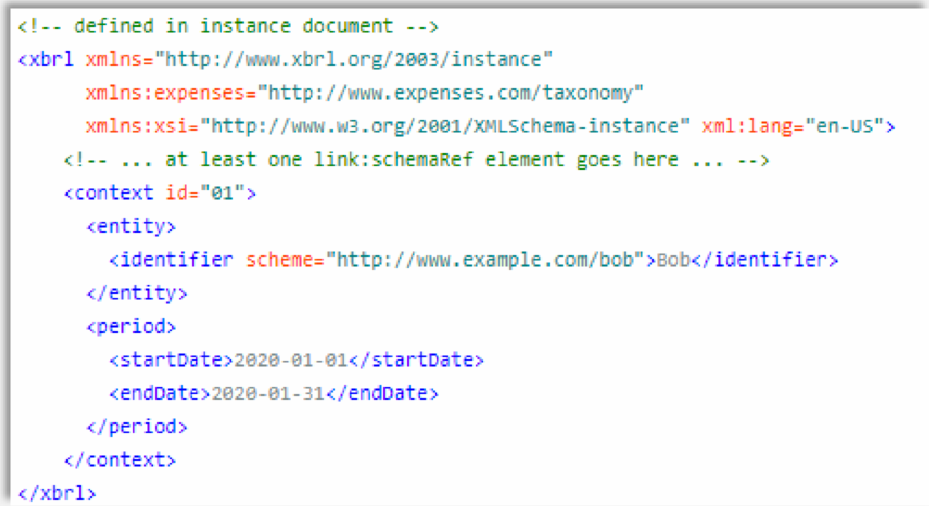
3.3.10 Creating Context

Assuming we want to report that Bob's Food expenses for January 2020 was $900. These are 3 pieces of additional information to the expense amount:

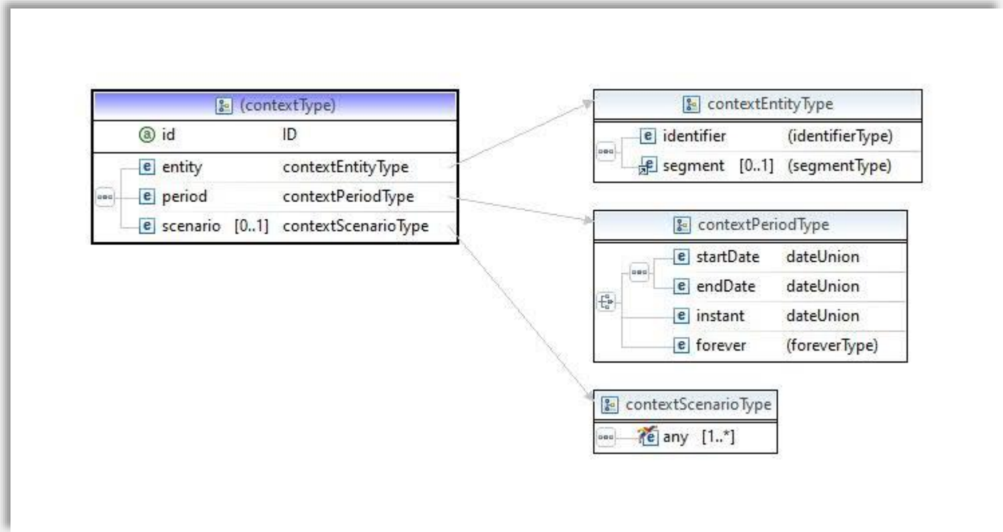
- Food the concept core dimension.
- Bob the owner of the expense.
- January 2020 the period core dimension.

We already defined the Food concept; the owner and the period core dimensions are added through the context.

Context is an XBRL element used in XBRL instance document (report) and referenced by facts in the XBRL report. It contains information about period, entity, and other taxonomy defined dimension relating to this context. See supporting materials

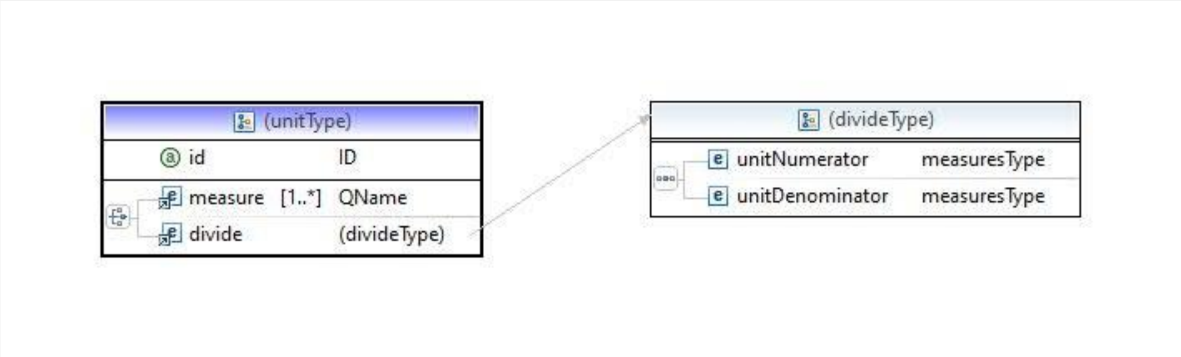


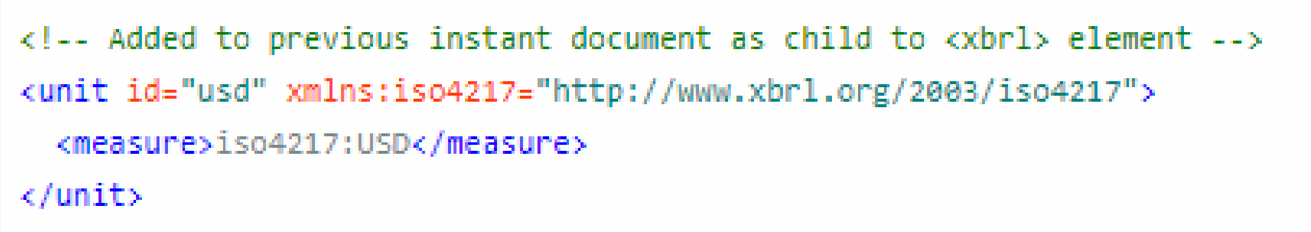
```
<!-- defined in instance document -->
<xbrl xmlns="http://www.xbrl.org/2003/instance"
      xmlns:expenses="http://www.expenses.com/taxonomy"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xml:lang="en-US">
    <!-- ... at least one link:schemaRef element goes here ... -->
    <context id="01">
      <entity>
        <identifier scheme="http://www.example.com/bob">Bob</identifier>
      </entity>
      <period>
        <startDate>2020-01-01</startDate>
        <endDate>2020-01-31</endDate>
      </period>
    </context>
</xbrl>
```



3.3.11 Creating Unit Core Dimension

XBRL requires that numeric facts has a reference to a unit [see XBRL specs 4.6.2]. And since our concept in monetary type which is numeric type, then we need to create a unit in our instance, in addition to the context before we can create a fact.



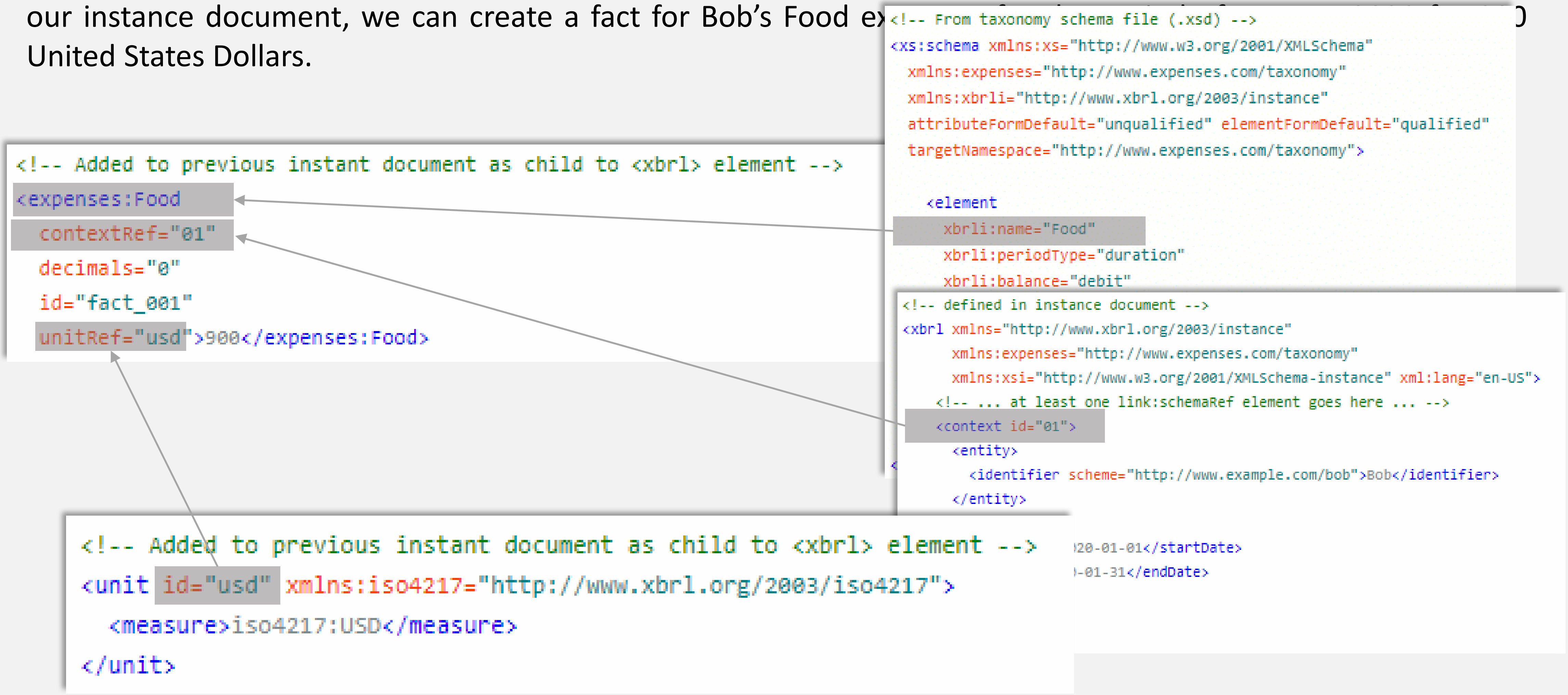


```
<!-- Added to previous instant document as child to <xbrl> element -->
<unit id="usd" xmlns:iso4217="http://www.xbrl.org/2003/iso4217">
  <measure>iso4217:USD</measure>
</unit>
```

Namespace iso4217 is references so we can use the “USD” unit defined in that namespace.

3.3.12 Creating Instance Fact

Now that we have Food concept in our XBRL Taxonomy and have a context with id=01 and a unit of id="usd" in our instance document, we can create a fact for Bob's Food ex 0 United States Dollars.



```
<!-- Added to previous instant document as child to <xbrl> element -->
<expenses:Food
  contextRef="01"
  decimals="0"
  id="fact_001"
  unitRef="usd">900</expenses:Food>
```

```
<!-- From taxonomy schema file (.xsd) -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:expenses="http://www.expenses.com/taxonomy"
  xmlns:xbrli="http://www.xbrl.org/2003/instance"
  attributeFormDefault="unqualified" elementFormDefault="qualified"
  targetNamespace="http://www.expenses.com/taxonomy">

    <element
      xbrli:name="Food"
      xbrli:periodType="duration"
      xbrli:balance="debit"
```

```
<!-- defined in instance document -->
<xbrl xmlns="http://www.xbrl.org/2003/instance"
      xmlns:expenses="http://www.expenses.com/taxonomy"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xml:lang="en-US">
    <!-- ... at least one link:schemaRef element goes here ... -->
    <context id="01">
      <entity>
        <identifier scheme="http://www.example.com/bob">Bob</identifier>
      </entity>
20-01-01</startDate>
-01-31</endDate>
```

```
<!-- Added to previous instant document as child to <xbrl> element -->
<unit id="usd" xmlns:iso4217="http://www.xbrl.org/2003/iso4217">
  <measure>iso4217:USD</measure>
</unit>
```

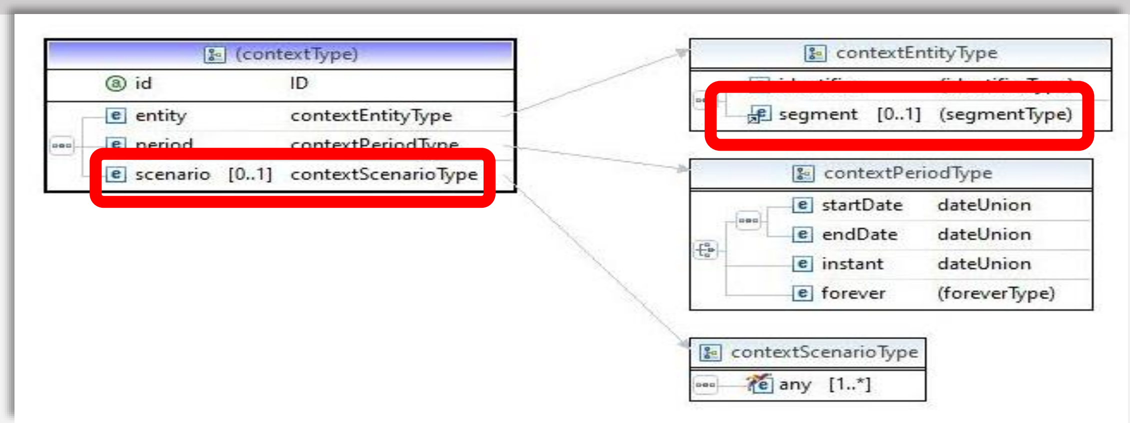
3.3.13 Segment and Scenario Elements

XBRL element context has 2 additional features that can provide dimensionality to a fact, <segment> and <scenario>.

<segment>: is defined in XBRL specifications as "an optional container for additional mark-up that the preparer of an XBRL Instance SHOULD use to identify the business segment more completely in cases where the Entity identifier is insufficient."

In addition, **<segment>** element is a child element of the <entity> element (which in turn is a child element of the <context> element) used to link with taxonomy defined dimension using *XBRL Dimensions Specifications*.

<scenario>: XBRL specifications describes this element as "Business facts can be reported as actual, budgeted, restated, pro forma, etc. For internal reporting purposes, there can be an even greater variety of additional metadata that preparers want to associate with items. The optional element allows additional valid mark-up (see note above regarding segment) to be included for this purpose."

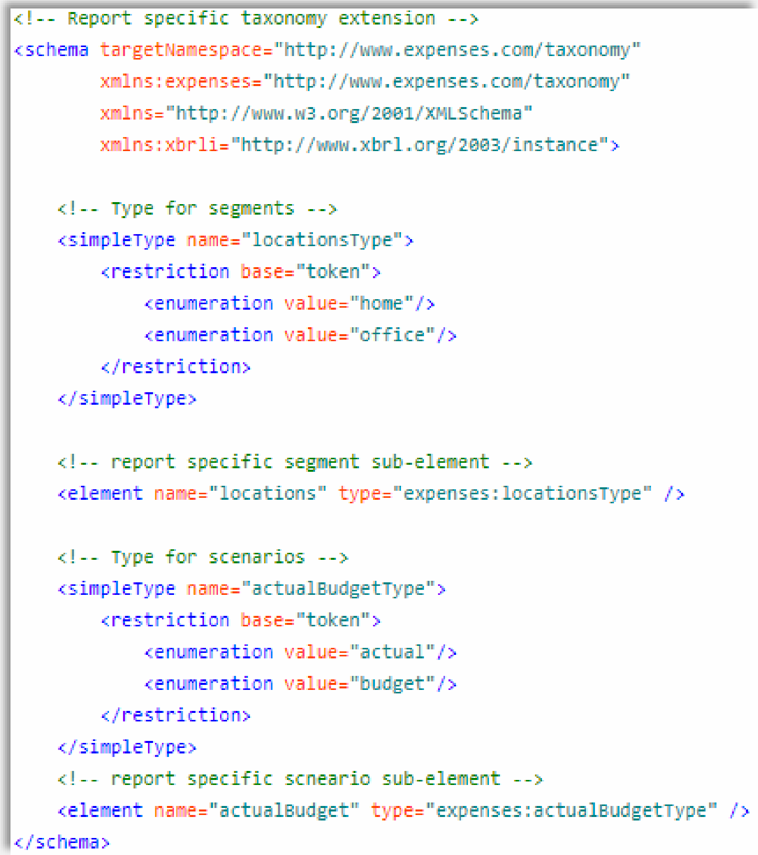


3.3.13 Segment and Scenario Elements (Cont.)

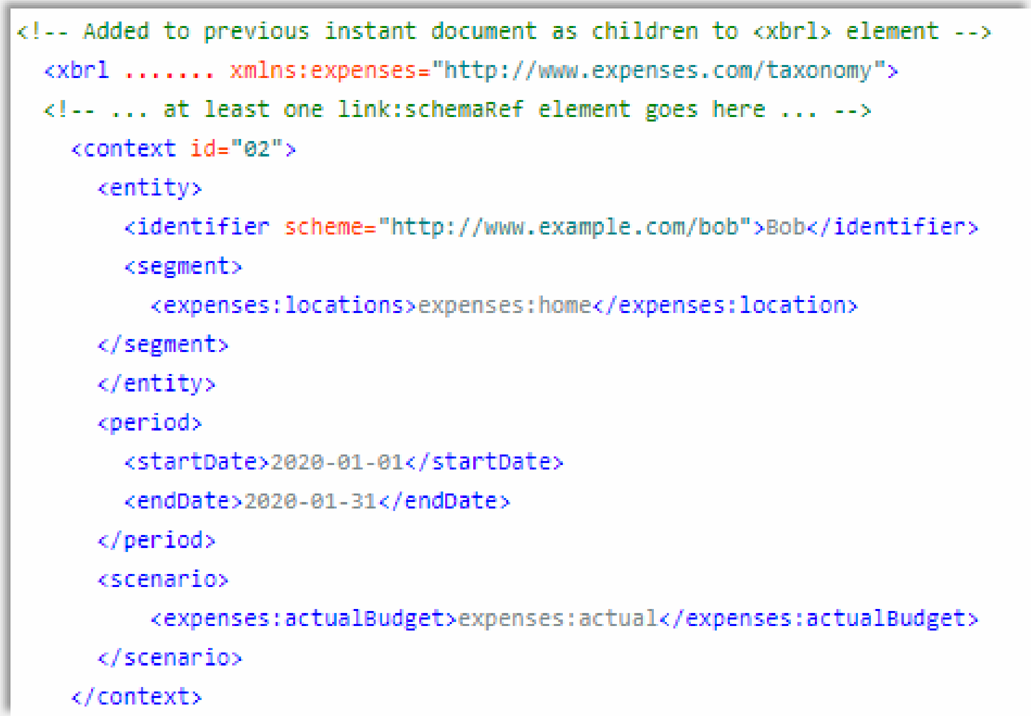
In the monthly expense report example, we make 2 assumptions:

- Bob has 2 locations to track expenses for; home and office represented as segments
- Bob tracks budget and actuals represented as scenarios

To be able to include these dimensions in our report we need **first to create an extension taxonomy**, then include these dimensions in the report as follows:



```
<!-- Report specific taxonomy extension -->
<schema targetNamespace="http://www.expenses.com/taxonomy"
        xmlns:expenses="http://www.expenses.com/taxonomy"
        xmlns="http://www.w3.org/2001/XMLSchema"
        xmlns:xbrli="http://www.xbrl.org/2003/instance">

    <!-- Type for segments -->
    <simpleType name="locationsType">
        <restriction base="token">
            <enumeration value="home"/>
            <enumeration value="office"/>
        </restriction>
    </simpleType>

    <!-- report specific segment sub-element -->
    <element name="locations" type="expenses:locationsType" />

    <!-- Type for scenarios -->
    <simpleType name="actualBudgetType">
        <restriction base="token">
            <enumeration value="actual"/>
            <enumeration value="budget"/>
        </restriction>
    </simpleType>
    <!-- report specific scneario sub-element -->
    <element name="actualBudget" type="expenses:actualBudgetType" />
</schema>
```



```
<!-- Added to previous instant document as children to <xbrl> element -->
  <xbrl ....... xmlns:expenses="http://www.expenses.com/taxonomy">
  <!-- ... at least one link:schemaRef element goes here ... -->
    <context id="02">
      <entity>
        <identifier scheme="http://www.example.com/bob">Bob</identifier>
        <segment>
          <expenses:locations>expenses:home</expenses:location>
      </segment>
      </entity>
      <period>
        <startDate>2020-01-01</startDate>
        <endDate>2020-01-31</endDate>
      </period>
      <scenario>
          <expenses:actualBudget>expenses:actual</expenses:actualBudget>
      </scenario>
    </context>
```

3.3.14 Taxonomy Defined Dimensions

> A taxonomy-defined dimension is a grouping of concepts that is used to add organizational structure to facts. These dimensional concepts should not be directly associated with a data point but rather are employed to indicate additional contextual information beyond the simple semantic identifier or what is provided through any of the other core dimensions.
> — TDH section 2.2.8 page 24

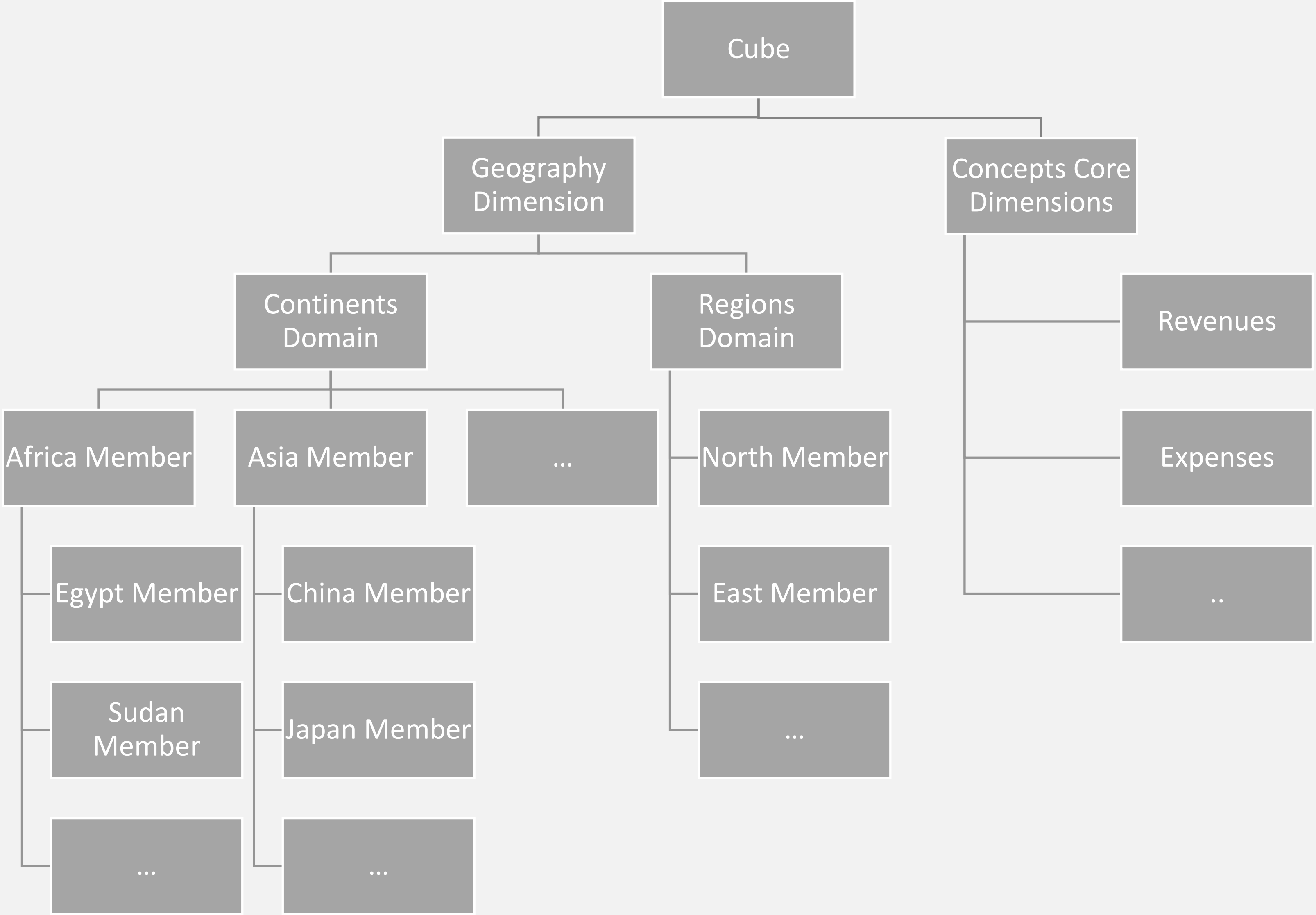
Taxonomy defined dimension are the focus of *XBRL Dimensions Specifications*.

XBRL Dimensions terminology

- **Dimension**: A qualifying characteristic that is used to uniquely define a data point (other than core dimensions) for example a “Geography Dimension”.
- **Domain**: A set of related values. Examples domains of a “Geography Dimension” would be “Countries”, and “Continents”.
- **Domain member**: An element representing one of the possibilities within a domain, example, Egypt, Sudan,…
- **Cube**: A cube is defined by combining a set of dimensions with a set of concepts. Cubes are often referred to as “hypercubes”, as unlike a physical, 3-dimensional cube, a hypercube may have any number of dimensions.

All the above constructs are defined as concepts but with special values for the @type and @substitutionGroup attributes, these values are defined in XBRL Dimensions specifications. (See illustration in next page)

See simplified example for Dimensions in supporting materials



Cube

Geography Dimension

Concepts Core Dimensions

Continents Domain

Regions Domain

Revenues

Africa Member

Asia Member

…

North Member

Expenses

Egypt Member

China Member

East Member

..

Sudan Member

Japan Member

…

…

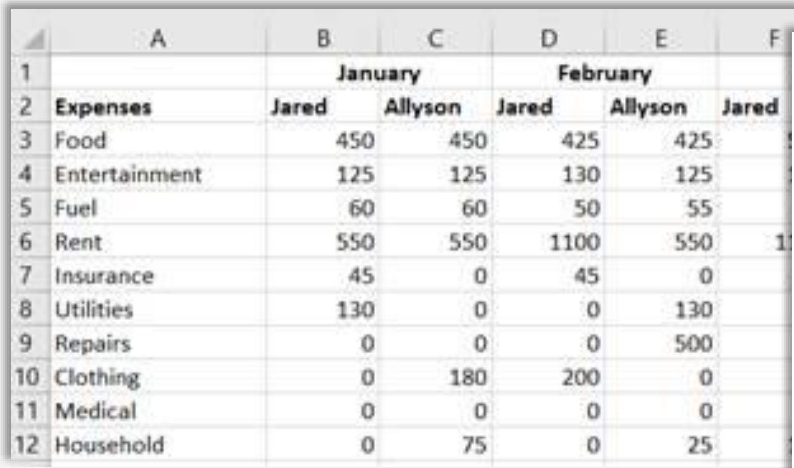
…

3.3.14 Taxonomy Defined Dimensions (cont.)

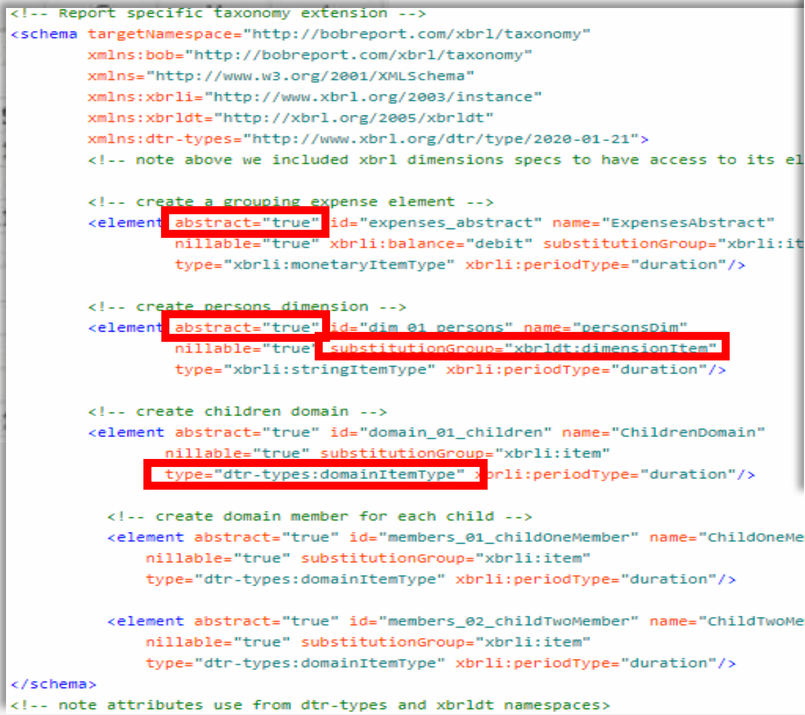
Assume we want to report monthly expenses split by Bob’s children, with each month split into 2 columns for each of Bob’s children.

we need to do the following:

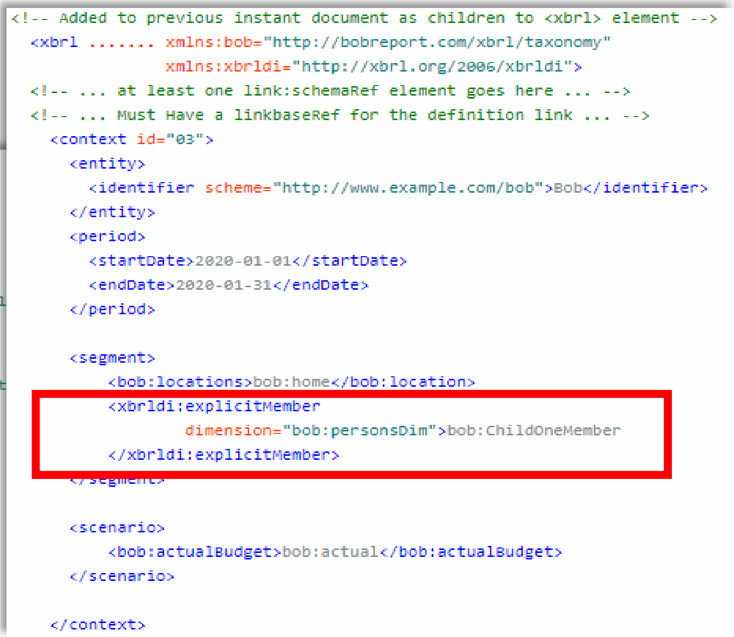
- Create a grouping concept or header called expenses to group all the expenses together,
- Create “persons” dimension,
- Create a domain for “bobChildrenDomain”
- Create a domain member for each child referenced in the report.



	A	B	C	D	E	F
1		January		February		
2	Expenses	Jared	Allyson	Jared	Allyson	Jared
3	Food	450	450	425	425	
4	Entertainment	125	125	130	125	
5	Fuel	60	60	50	55	
6	Rent	550	550	1100	550	
7	Insurance	45	0	45	0	
8	Utilities	130	0	0	130	
9	Repairs	0	0	0	500	
10	Clothing	0	180	200	0	
11	Medical	0	0	0	0	
12	Household	0	75	0	25	



```
<!-- Report specific taxonomy extension -->
<schema targetNamespace="http://bobreport.com/xbrl/taxonomy"
        xmlns:bob="http://bobreport.com/xbrl/taxonomy"
        xmlns="http://www.w3.org/2001/XMLSchema"
        xmlns:xbrli="http://www.xbrl.org/2003/instance"
        xmlns:xbrldt="http://xbrl.org/2005/xbrldt"
        xmlns:dtr-types="http://www.xbrl.org/dtr/type/2020-01-21">
        <!-- note above we included xbrl dimensions specs to have access to its el

        <!-- create a grouping expense element -->
        <element abstract="true" id="expenses_abstract" name="ExpensesAbstract"
             nillable="true" xbrli:balance="debit" substitutionGroup="xbrli:it
             type="xbrli:monetaryItemType" xbrli:periodType="duration"/>

        <!-- create persons dimension -->
        <element abstract="true" id="dim_01_persons" name="personsDim"
             nillable="true" substitutionGroup="xbrldt:dimensionItem"
             type="xbrli:stringItemType" xbrli:periodType="duration"/>

        <!-- create children domain -->
        <element abstract="true" id="domain_01_children" name="ChildrenDomain"
            nillable="true" substitutionGroup="xbrli:item"
            type="dtr-types:domainItemType" xbrli:periodType="duration"/>

          <!-- create domain member for each child -->
          <element abstract="true" id="members_01_childOneMember" name="ChildOneMember"
              nillable="true" substitutionGroup="xbrli:item"
              type="dtr-types:domainItemType" xbrli:periodType="duration"/>

          <element abstract="true" id="members_02_childTwoMember" name="ChildTwoMember"
              nillable="true" substitutionGroup="xbrli:item"
              type="dtr-types:domainItemType" xbrli:periodType="duration"/>
</schema>
<!-- note attributes use from dtr-types and xbrldt namespaces>
```



```
<!-- Added to previous instant document as children to <xbrl> element -->
  <xbrl ....... xmlns:bob="http://bobreport.com/xbrl/taxonomy"
                xmlns:xbrldi="http://xbrl.org/2006/xbrldi">
  <!-- ... at least one link:schemaRef element goes here ... -->
  <!-- ... Must Have a linkbaseRef for the definition link ... -->
    <context id="03">
      <entity>
        <identifier scheme="http://www.example.com/bob">Bob</identifier>
      </entity>
      <period>
        <startDate>2020-01-01</startDate>
        <endDate>2020-01-31</endDate>
      </period>

      <segment>
          <bob:locations>bob:home</bob:location>
          <xbrldi:explicitMember
                 dimension="bob:personsDim">bob:ChildOneMember
          </xbrldi:explicitMember>
      </segment>

      <scenario>
          <bob:actualBudget>bob:actual</bob:actualBudget>
      </scenario>

    </context>
```

3.3.15 Link Types and XBRL

XBRL linkbases (based on XML XLink) is mechanism to create relationships between XBRL elements and other internal or external resources to create a meaningful self-describing data structure.

Simple Links: A simple link in XLink creates a unidirectional hyperlink from one element (**source**) to another internal or external elements or resources (**destination**) through a URI (this is similar to HTML hyperlinking).

Source
Destination

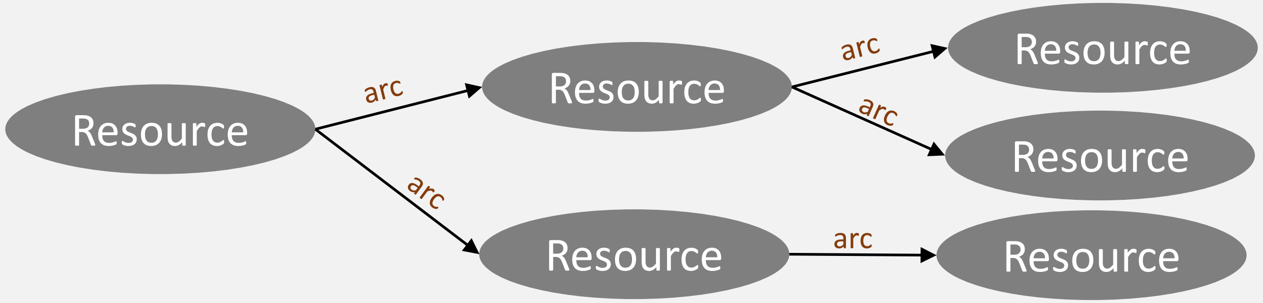
SchemaRef

linkbaseRef

linkbaseRef

…

Extended Links: A mechanism that allows multiple resources at the source or destination to be connected **via multiple arcs**. An **arc** contains information about the source, destination, and the behavior of a link. The source and the destination are defined by labels. Through one or more arcs, extended links achieve complex connections among multiple resources. Like simple links, extended links can define relationships between elements within the same namespace or across different namespaces.



presentationLink

calcultionLink

definitionLink

labelLink

…

3.3.16 XBRL Linkbases

The Term “**Linkbase**” in XBRL refers to a collection of XBRL Links in specific structure.

Most Common Types of XBRL Linkbases (There are more not mentioned here):

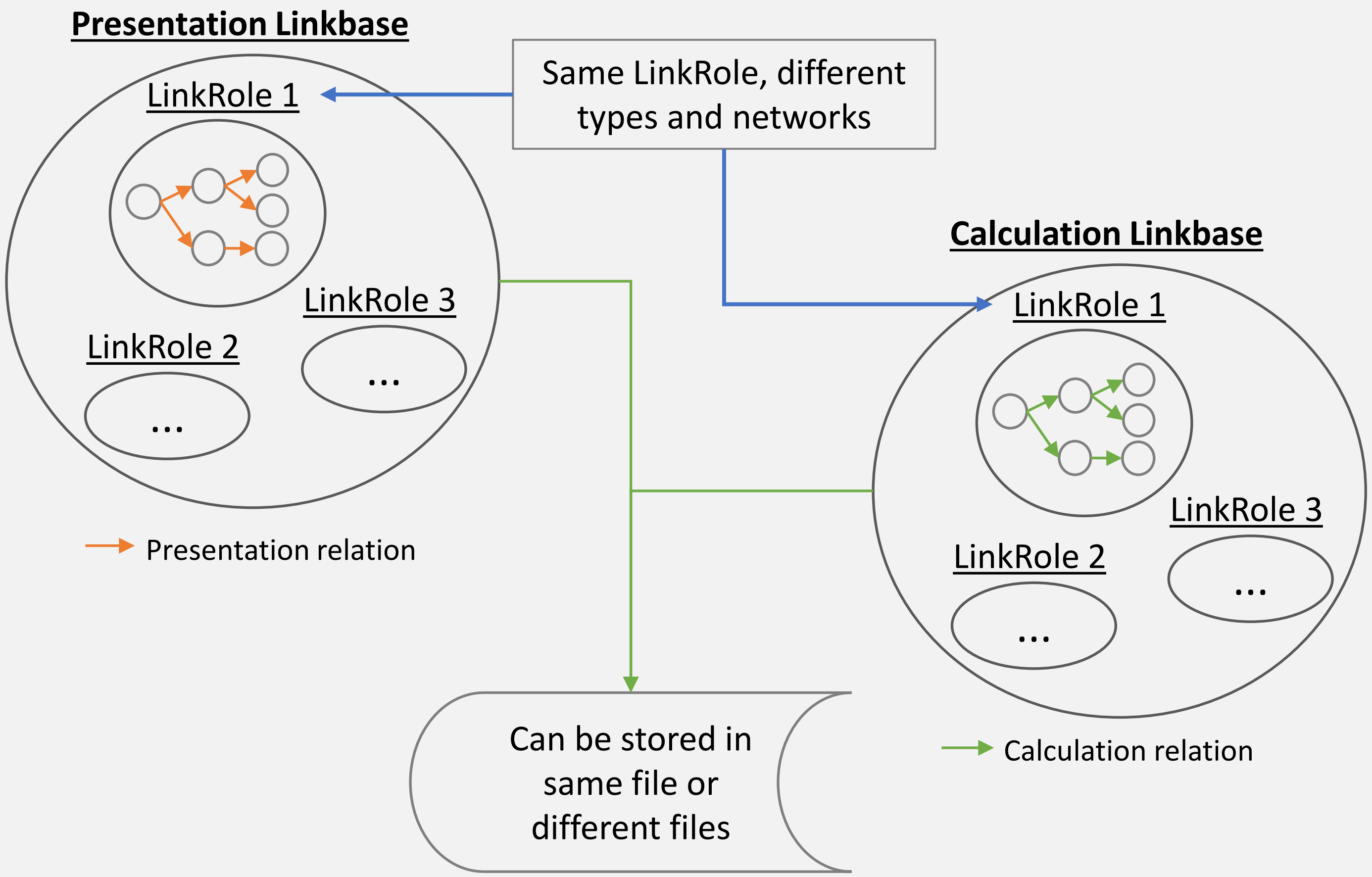
- **Presentation**: An extended link providing for the organization of taxonomy elements into a hierarchical structure.
- **Calculation**: An extended link providing relationships between concepts in a taxonomy for the purpose of describing and validating simple totals and subtotals.
- **Label**: An extended link providing a relationship between concept and human readable description of a taxonomy component in one or more languages.
- **Definition**: An extended providing for relationships that arranges pairs of concepts in a specific semantic relationship. These relationships may be above and beyond calculation or presentation relationships.

See supporting materials and TDH Section A.2 page 155

XBRL is typically comprised of one or more schema files (defined using XSD), and one or more linkbase files (defined using XLink). The linkbase files contain the concept relationships that help define the structure of the taxonomy. Taxonomies at a minimum should have presentation and definition linkbases; without these, the taxonomy lacks important relational structure. Calculation linkbases also provide useful mathematical relationship information as necessary.
-- TDH Section A.2 page 155

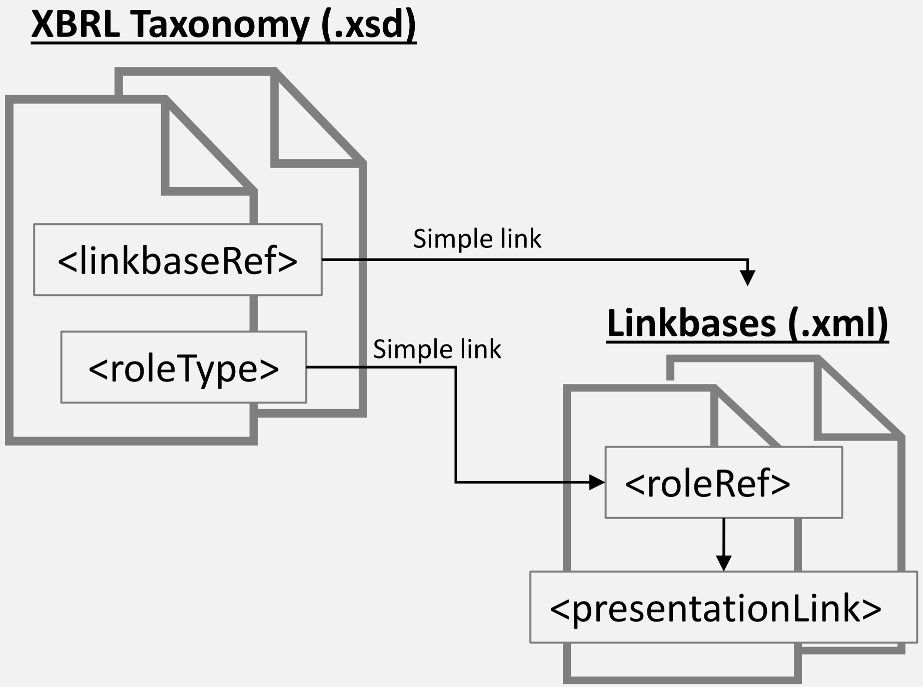
3.3.17 XBRL Linkbases Overview

XBRL links are organized by type (presentation, Definitions, …), then with in each type, links are further partitioned into “Networks of Relations” using “Extended Link Roles” (LinkRole for short). Link bases can be visualized as follows:



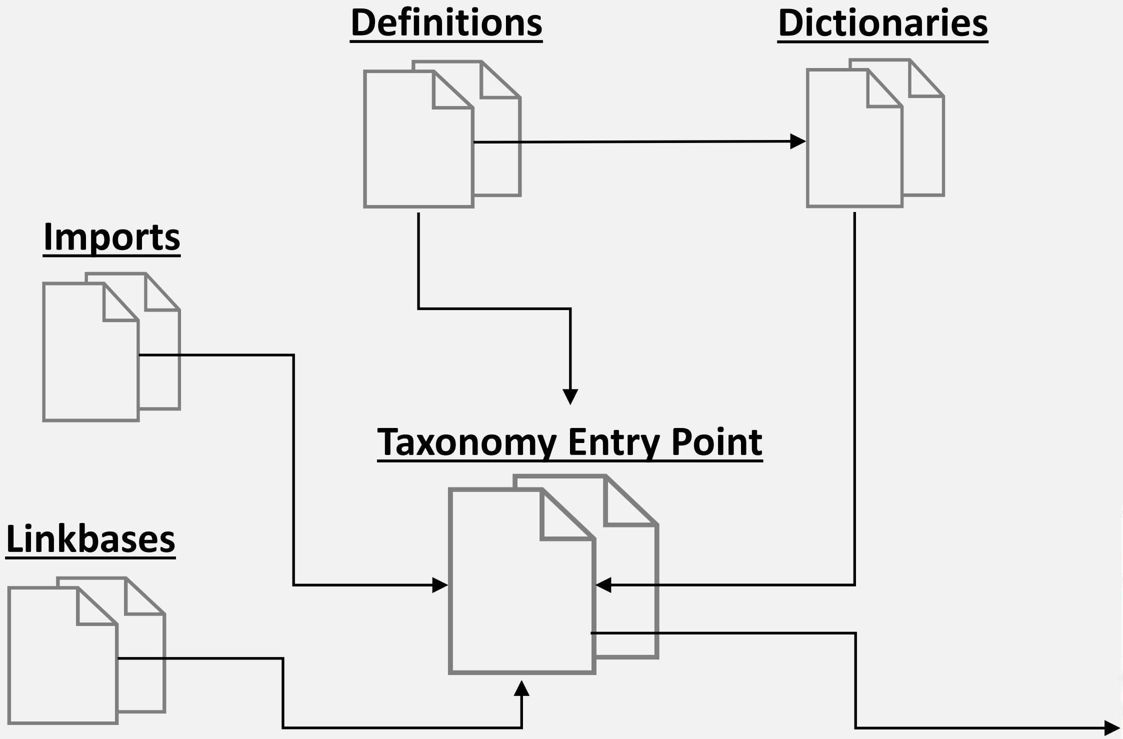
3.3.18 Linkbases XBRL References

- Linkbases are ".xml" files, referenced in XBRL taxonomy using element "<linkbaseRef>".
- "Extended Link Roles" are declared in XBRL taxonomy and referenced from linkbase files to be used to group links (Networks).



3.4 Example Income Statement

3.4.1 Overview

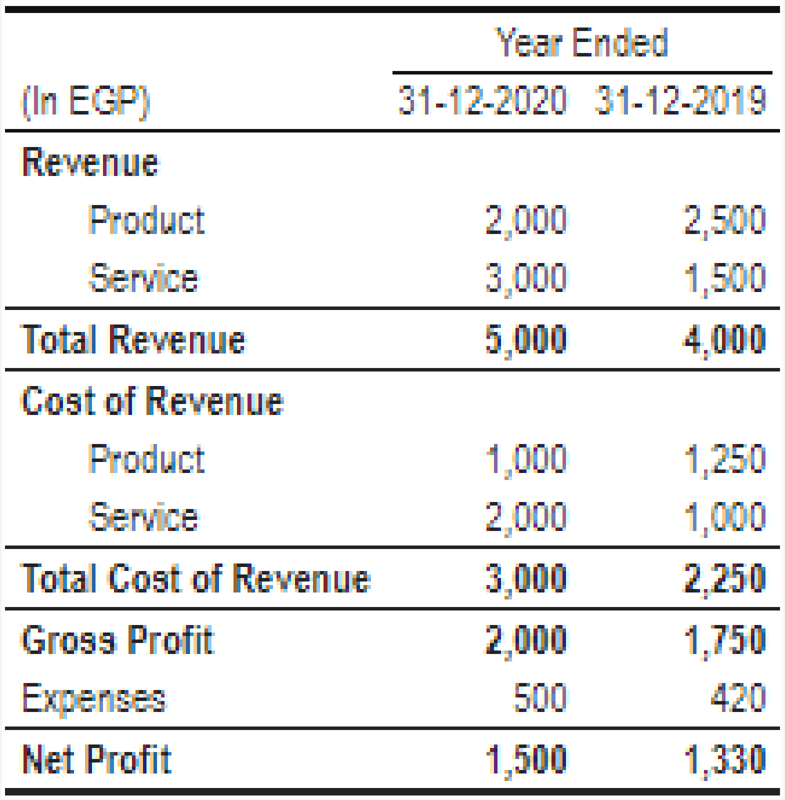


(In EGP)	Year Ended 31-12-2020	31-12-2019
Revenue		
Product	2,000	2,500
[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]
[illegible]		
[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]

3.4.2 Income Statement Example

In this section we go through the taxonomy and linkbases necessary to present a simple income statement in XBRL.

Description	Reference	File
Taxonomy (Schema File)	Link	is.xsd
Presentation Linkbase	link	is_pre.xml
Calculation Linkbase	link	is_cal.xml
Label Linkbases	link	is_lab_en.xml is_lab_ar.xml
Definition Linkbase	link	is_def.xml
XBRL Instance	link	instance.xml



	Year Ended	
(In EGP)	31-12-2020	31-12-2019
Revenue		
Product	2,000	2,500
Service	3,000	1,500
Total Revenue	5,000	4,000
Cost of Revenue		
Product	1,000	1,250
Service	2,000	1,000
Total Cost of Revenue	3,000	2,250
Gross Profit	2,000	1,750
Expenses	500	420
Net Profit	1,500	1,330

Required Format

3.4.3 Validation

In general, **validation** is the process of ensuring compliance with specific criteria and/or set of rules.

XBRL has two layers of validation:

- **Basic Validation**: XBRL Specifications requires that XBRL Instance, XBRL Linkbases and XBRL Taxonomy Schema must comply with the specifications, and this compliance is ensured through Validation process [XBRL specifications section 3.4]. This includes (but not limited to) syntax, data type and concept relationships-based validations.
- **Business Rules Validation:** Checks for compliance with rules specific to the domain and taxonomy; those will differ depending on the taxonomy authors' objectives and rules. This makes it necessary to implement custom validations that checks for compliance with specific business rules.

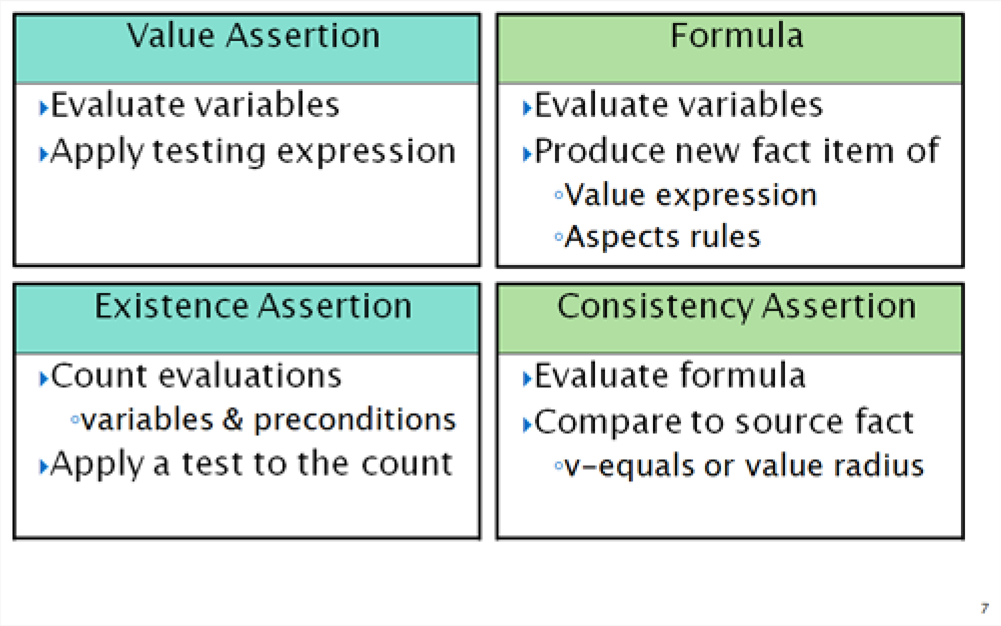
See Supporting materials

3.4.4 Validation – XBRL Formula

> XBRL formulas provide a standardized method for defining validation rules for XBRL reports that go beyond what is provided through calculations and other concept relationships.
> — TDH section 6.2.1 page 88

XBRL Formula is defined in XBRL Formula Specifications. XBRL Formula is a linkbase that can be embedded in the taxonomy itself or incorporated as needed. See formula example.

XBRL Formula has 4 processing models as follows:



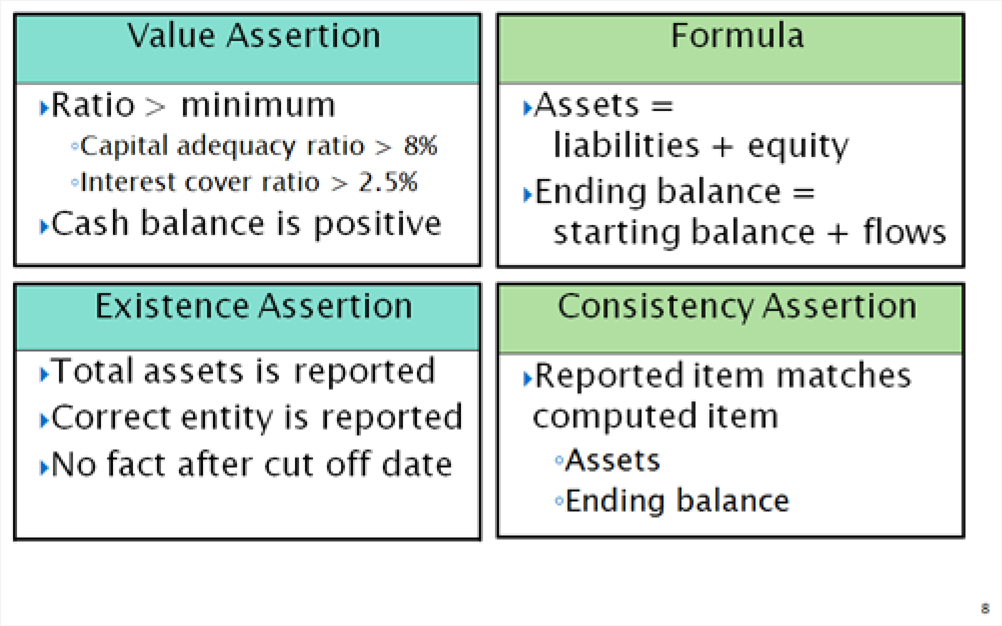
Value Assertion
- Evaluate variables
- Apply testing expression

Formula
- Evaluate variables
- Produce new fact item of
  - Value expression
  - Aspects rules

Existence Assertion
- Count evaluations
  - variables & preconditions
- Apply a test to the count

Consistency Assertion
- Evaluate formula
- Compare to source fact
  - v-equals or value radius

7



Value Assertion
- Ratio > minimum
  - Capital adequacy ratio > 8%
  - Interest cover ratio > 2.5%
- Cash balance is positive

Formula
- Assets = liabilities + equity
- Ending balance = starting balance + flows

Existence Assertion
- Total assets is reported
- Correct entity is reported
- No fact after cut off date

Consistency Assertion
- Reported item matches computed item
  - Assets
  - Ending balance

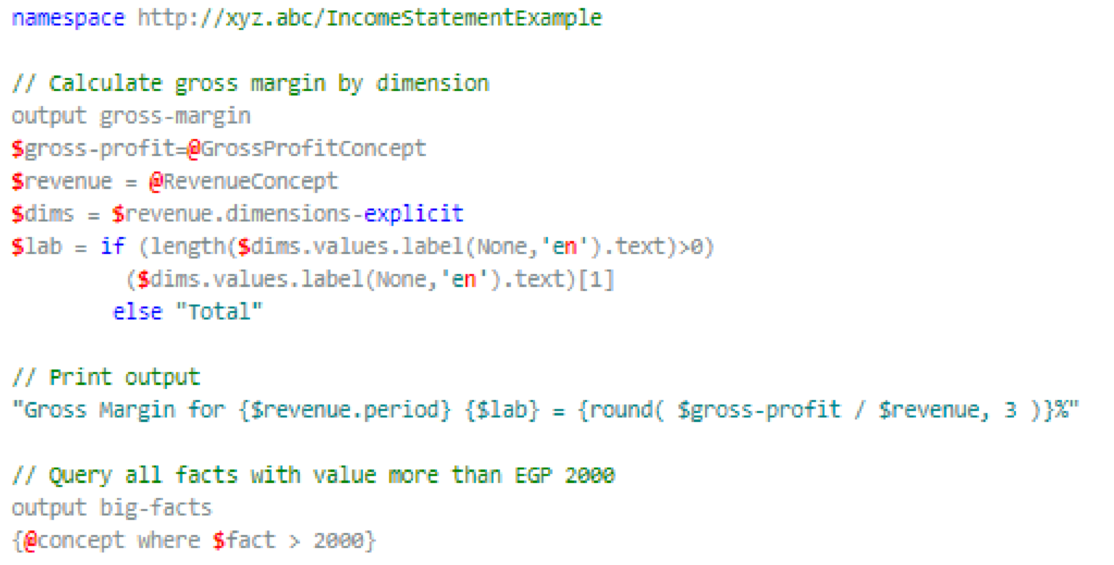
8

3.4.5 Validation – XULE

> Developed by XBRL US, XULE is an expression syntax that allows the querying of XBRL reports and taxonomies using a XULE processor. The primary purpose of XULE is to provide a user-friendly syntax to query and manipulate XBRL data. This can be helpful in a multitude of ways, including aiding consumers in quickly extracting specific facts from reports and supporting developers in querying XBRL taxonomies to render them as open API schemas or as iXBRL forms.
>
> — TDH section 6.2.2 page 99

XBRL US published XULE Language Syntax publicly available here

Sample XULE:



```
namespace http://xyz.abc/IncomeStatementExample

// Calculate gross margin by dimension
output gross-margin
$gross-profit=@GrossProfitConcept
$revenue = @RevenueConcept
$dims = $revenue.dimensions-explicit
$lab = if (length($dims.values.label(None,'en').text)>0)
         ($dims.values.label(None,'en').text)[1]
        else "Total"

// Print output
"Gross Margin for {$revenue.period} {$lab} = {round( $gross-profit / $revenue, 3 )}%"

// Query all facts with value more than EGP 2000
output big-facts
{@concept where $fact > 2000}
```

3.4.2 Table Linkbase

XBRL Table Linkbase specifications provides a mechanism for taxonomy authors to define a tabular layout of facts. The resulting tables can be used for both presentation and data entry.

> *Table linkbase enables the definition of tables with multiple axes. The components of these axes are not limited to individual items; instead, they can be defined in terms of a combination of dimensions, time period references, units, entities or any other property that can be used to identify the financial facts represented by taxonomies.*
> — XBRL Table Linkbase Overview

See supporting material

Example table linkbase

3.5 Taxonomy - Other Topics

3.5.1 Inline XBRL (iXBRL)

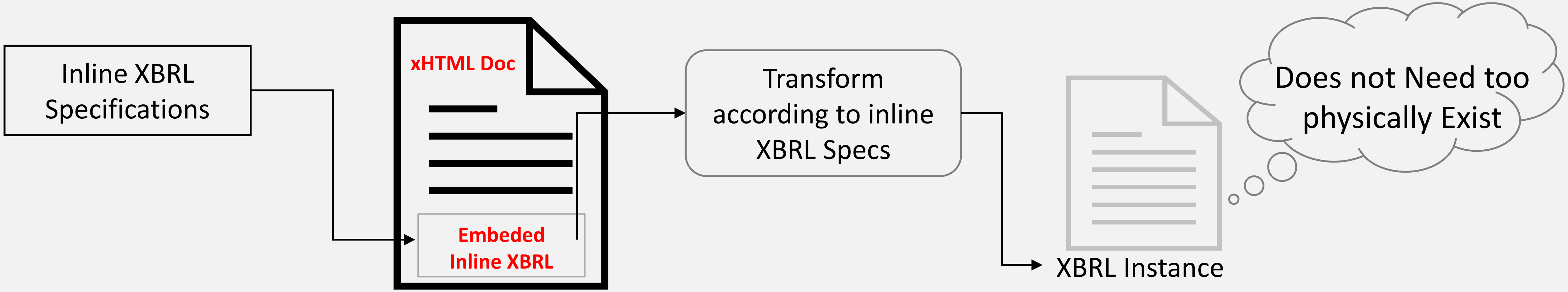
Inline XBRL specifications provide a mechanism for embedding XBRL tags in HTML documents (xHTML is required by the specifications). This allows the XBRL benefits of tagged data to be combined with a human-readable presentation of a report.

Inline XBRL specifications define XBRL elements that are used from within xHTML to form the metadata necessary to describe an XBRL instance document which is referred to as **Target Document.**

Target Document is defined as valid XBRL instance document represented by metadata in the Inline XBRL Document Set.

The target document need not physically exist, but the metadata must be sufficient to construct the target document when transformed according to the mapping rules prescribed in inline XBRL Specification.

See supporting materials   Microsoft Filing with usual form   Microsoft Filing with inline XBRL Viewer



Inline XBRL Specifications
xHTML Doc
Embeded Inline XBRL
Transform according to inline XBRL Specs
XBRL Instance
Does not Need too physically Exist

3.5.2 Taxonomy Entry Points

Taxonomy Entry Point is a collection of concept groups that have been joined for a specific use. For example, an entry point may consist of presentations, calculations, and definitions that are relevant to banking within a financial reporting taxonomy.
-- TDH Section 3.4.1 Page 30

Entry points help organize concepts by their use and can aid preparers in navigating a taxonomy and locating the sections that apply to their reporting needs.

Entry points are created through the actual physical structure (files), concepts and presentations are grouped into files and these files can be accessed via entry points using XBRL referencing and import constructs.

Example entry points from US GAAP Taxonomy Technical Guide Figure 14 page 16

IFRS Taxonomy Entry points

3.5.3 Taxonomy Style Guides

Style guides are one of the supporting documents that accompany an XBRL Taxonomy. Style guides helps in achieving consistency while creating, maintaining or extending the taxonomy.

Style guides set the rules for consistent language and naming conventions, styles and organization. For example, a style guide rule my set the whether to use camel case or pascal case, what characters are allowed or disallowed in labels, and so on.

Examples of style guides:

- IFRS Taxonomy Style Guide (The IFRS Taxonomy Architecture - Appendix A - page 33)
- XBRL US Style Guide

3.5.4 Open vs Closed Reporting System

Open Reporting System allows its XBRL taxonomy to be extended or customized. An open reporting system permits entity-specific reporting. Users can import additional taxonomies and can create their own concepts, datatypes and links to use in reports in conjunction with the taxonomy, usually this is subject to rules.

Closed Reporting System is reporting system that has a strictly defined structure whose taxonomy cannot be extended. Users cannot make any change to existing taxonomy; taxonomy is to be used as-is.

Open	Closed
Allows for greater flexibility in expressing users' specific dataset, but increases complexity and impairs comparability	Enhances Comparability, analysis, and interpretability while limiting the ability to express users' specific dataset.
Best suited for general purpose reports such as general-purpose GAAP based financial statements	Best suited for regulatory reporting with predefined structure.

NOTE: XML is extensible, and nothing can change that, restriction on extensibility must be implemented through the reporting system, validation rules and procession software.

3.5.5 Open Information Model (OIM)

Open Information Model (OIM) specifications as of August 2021 is still under development; this specifications describes methods of relaying XBRL information in a syntax-independent manner. It explores both JSON (JavaScript Object Notation) and CSV (Comma Separated Values) while revising some of the terminology to work in such environments.

Open information model allows for the transport of XBRL instance only, taxonomy and any extensions needs to be provided using XML.

Income Statement example XBRL instance in JSON using OIM

Income Statement example XBRL instance in CSV using OIM

3.5.6 Three Taxonomies

XBRL Taxonomies represents the foundation of a reporting system based on XBRL; there are several approaches in designing XBRL Taxonomies, and it is useful to be familiar with the approaches used by major taxonomies.

The table below compares 3 taxonomies based on Design approach, type of reporting and extensibility:

	IFRS Taxonomy (2020)	US GAAP Taxonomy (2021)	EBA Taxonomy (Framework 3.1)
Link	IFRS Taxonomy Resources	FASB 2021 Taxonomy	EBA Reporting Frame 3.1
Type of reporting	**GAAP Reporting**	**GAAP Reporting**	**Regulatory Reporting**
Design Approach	**Standard Based Approach**	**Domain Model**	**Data Point Model (DPM)**
Extensibility	**Unrestricted extensions**	**Restricted/Limited extensions**	**Limited extensions**

3.6 XBRL Infrastructure

3.6.1 Why XBRL Based Reporting Systems

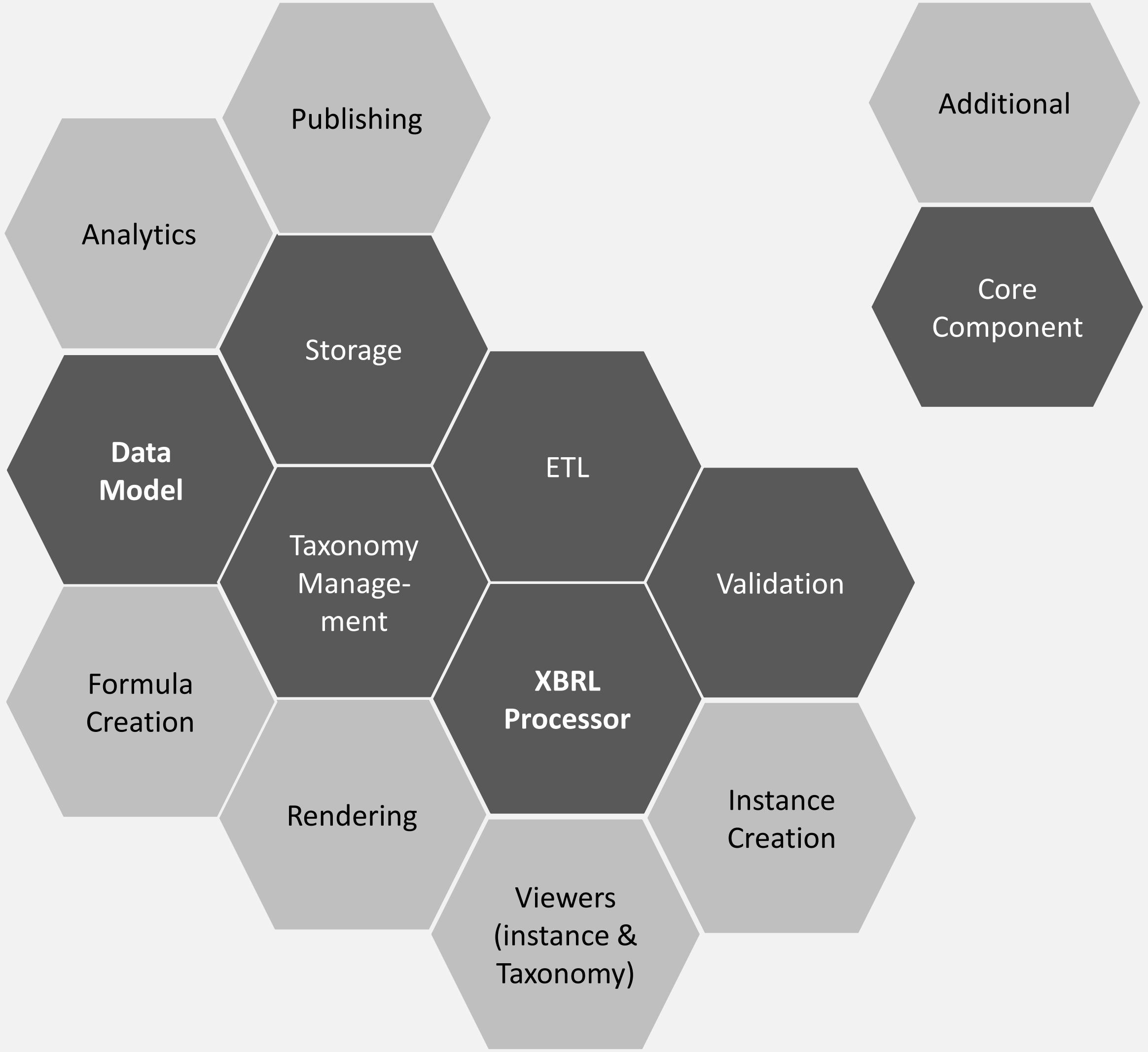
A reporting system may use XBRL as the report format because of its benefits:

- **End-to-end integrity**, with system-to-system connections, or alternatively form/template based assistance depending on the sophistication of the regulated entities, to **simplify and lower the costs** associated with compliance.
- Pre-defined public sets of business and accounting logic rules that must be passed prior to accepting filings, increasing the **quality and consistency of filed data**.
- Pre-defined **private sets of rules to help detect anomalies** early and allow resources to be assigned to high priority entities.
- **Multi-lingual** data input channels and mechanisms for multi-lingual republication of reports and forms, simplifying the accessibility of data where that is a goal.
- **filer-centric and analyst-centric** mechanisms for displaying reports.
- Deal with **complex multi-dimensional data**, including via presentation templates and forms, as well as well defined semantics.
- Natively **deal with accounting, Monetary and Boolean types** of information.
- **Store and transform the data** into existing or COTS business intelligence systems where that is a goal.
- Rapidly and accurately **republish** data for system-to-system ingestion and analysis.

See xbrl.org

3.6.2 XBRL reporting system components

- XBRL is just a part of the reporting system.
- To make a reporting system “XBRL enabled”, certain components are required
- **XBRL Processor (aka XBRL Engine)** is an essential component; it is a piece of software capable of parsing XBRL
- **Data Model** is an essential component, this is the model basis for the taxonomy and storage of data collected.
- Some of these components are relevant to Regulator only.



3.6.3 Approach for XBRL Report Creation

There are a wide variety of different approaches to XBRL document creation depending on the type of report, regulatory requirements, information contained in the report.

1. Forms based approach offered by regulator or third party.
2. In-house embedded production from existing software.
3. Outsourced production.
4. In-house production with “bolt on” tools and processes.
5. In-house embedded production with Disclosure Management or Regulatory Filing software and changed processes.

See xbrl.org

3.6.4 Tools and Software

XBRL enabled reporting systems relies on software to preform the XBRL functionalities, XBRL software vendors can obtain XBRL software certification from XBRL international to demonstrate conformance with XBRL specifications.

XBRL International lists certified software and available features on its website in the Tools and Services section, relevant features are as follows:

Relevant to Regulator and Issuer:

- Instance document validation
- Instance document viewer
- XBRL Processors and Software libraries
- XBRL collaboration tools
- Taxonomy Viewer

Relevant to Regulator:

- Data aggregation and publishing
- Quality assurance
- Taxonomy creation, maintenance and enhancement
- Taxonomy Validation
- Analytical tools

Relevant to issuer:

- Instance Document Creation
- Filing Preparation

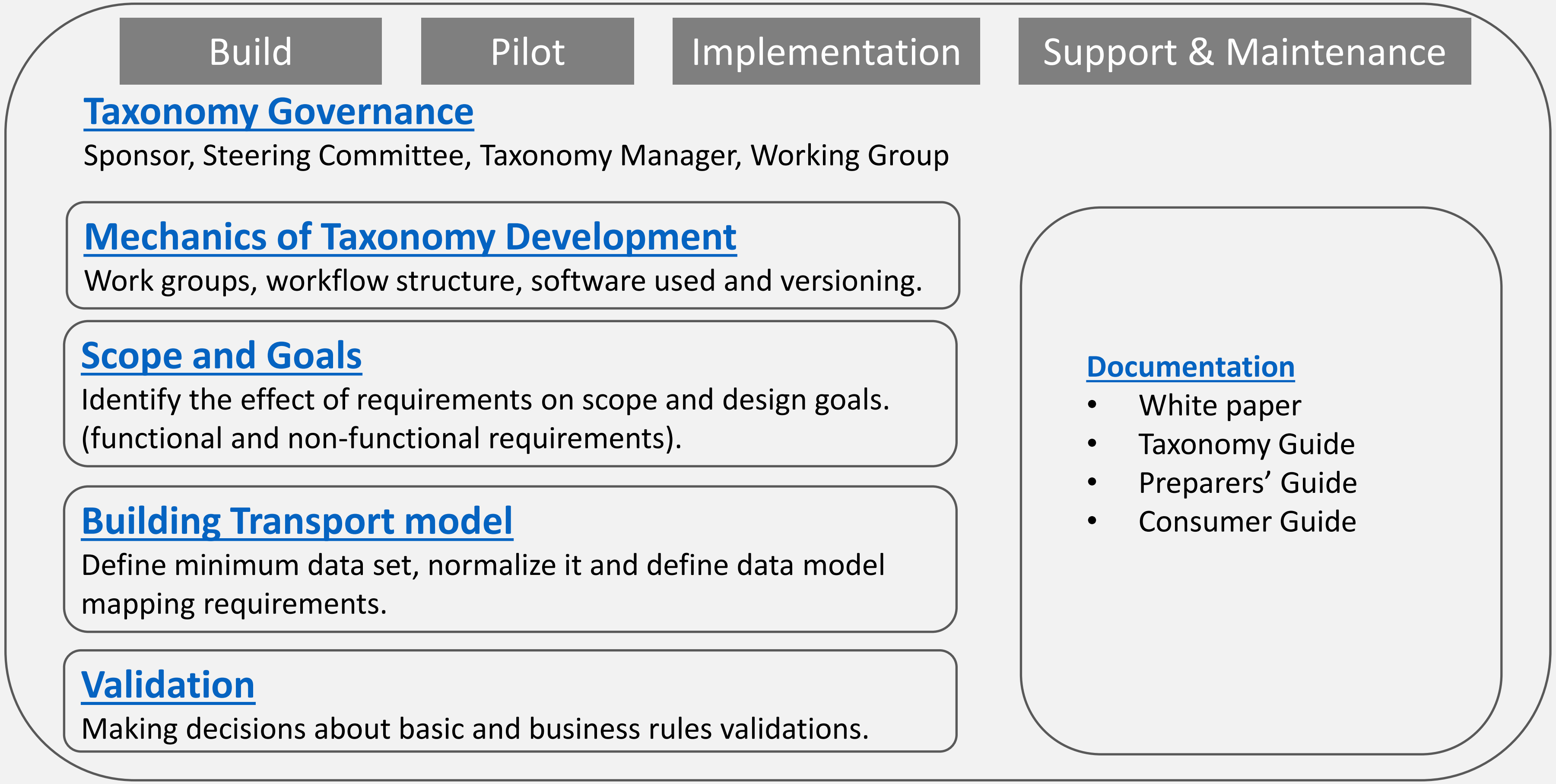
Features by vendors report

4 Taxonomy Development Project

4.1 Six Components

Taxonomy development might be a very a complicated task depending on the scope and size of the Taxonomy. Based on XBRL US experience, TDH starting from section 4 breaks down a Taxonomy Development Project into 6 main components and 4 phases as follows:

Taxonomy Development



Build

Pilot

Implementation

Support & Maintenance

Taxonomy Governance

Sponsor, Steering Committee, Taxonomy Manager, Working Group

Mechanics of Taxonomy Development

Work groups, workflow structure, software used and versioning.

Scope and Goals

Identify the effect of requirements on scope and design goals. (functional and non-functional requirements).

Building Transport model

Define minimum data set, normalize it and define data model mapping requirements.

Validation

Making decisions about basic and business rules validations.

Documentation

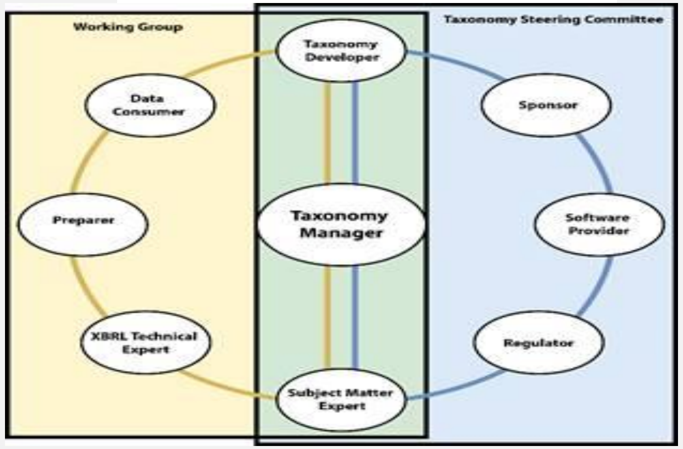
- White paper
- Taxonomy Guide
- Preparers’ Guide
- Consumer Guide

4.2 Taxonomy Governance

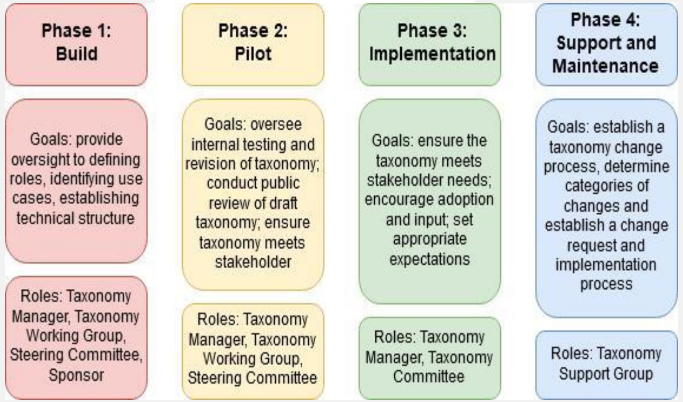
Governance Roles:

- **Sponsor:** Champions the development process and able to bring together stakeholders successfully, could be a regulatory agency or standards organization.
- **Working Group:** Includes representation of all stakeholders (regulator, preparers, developers, consumers…), this group perform the tasks to develop the taxonomy deliverables.
- **Steering Committee:** highest committee and is led by the sponsor, provides oversight, evaluates major milestones, reviews and approves deliverables, and serves as “tie breaker” on major decisions concerning the taxonomy.
- **Taxonomy Manager**: is the project manager, maintains detailed knowledge of the taxonomy and the project as a whole and provides day-to-day staff support for the taxonomy working group. Also interacts with feedback and reviews and comments, and reports to the steering committee and working group.

Governance Structure



Life Cycle



Phase 1: Build	Phase 2: Pilot	Phase 3: Implementation	Phase 4: Support and Maintenance
Goals: provide oversight to defining roles, identifying use cases, establishing technical structure	Goals: oversee internal testing and revision of taxonomy; conduct public review of draft taxonomy; ensure taxonomy meets stakeholder	Goals: ensure the taxonomy meets stakeholder needs; encourage adoption and input; set appropriate expectations	Goals: establish a taxonomy change process, determine categories of changes and establish a change request and implementation process
Roles: Taxonomy Manager, Taxonomy Working Group, Steering Committee, Sponsor	Roles: Taxonomy Manager, Taxonomy Working Group, Steering Committee	Roles: Taxonomy Manager, Taxonomy Committee	Roles: Taxonomy Support Group

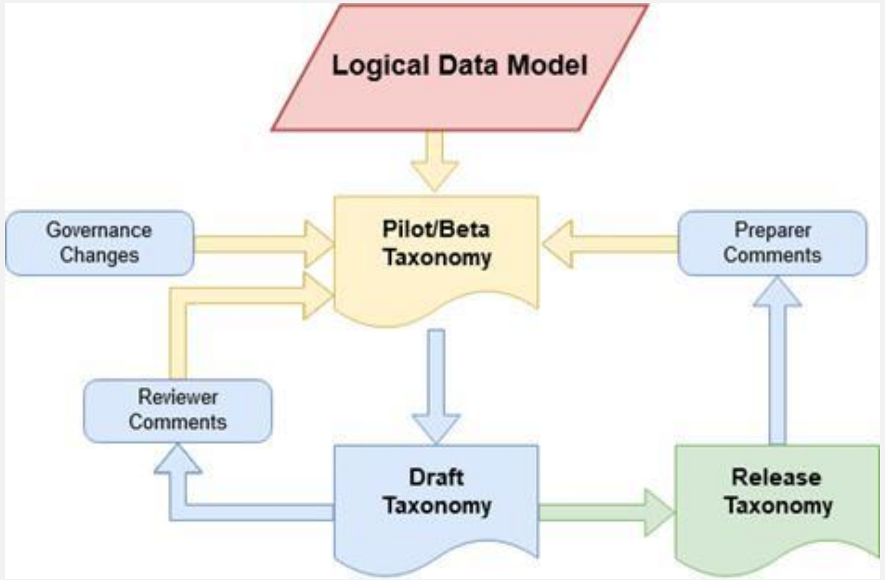
See Supporting materials, and TDH

4.3 Mechanics of Taxonomy Development

Workflow

Multiple groups performing different tasks are needed to create a taxonomy, and workflow should be organized, TDH gives examples of the groups as follows:

- Group responsible for creating data model and transforming it to taxonomy.
- Group responsible for overseeing incorporation of regulatory/governance rules and changes.
- Group responsible for reading reviewers' comments and making recommendations for modifications.



Determine the software to be used for generating the taxonomy

Taxonomy should undergo significant public review relative to the size and scope of the taxonomy, feedback and comments should be collected and analyzed.

See Supporting materials, and TDH

4.4 Scope and Goals

The goal of XBRL taxonomy is to facilitate the structured reporting of data from preparer to consumer. Project scope and goals should consider factors that enables preparers to produce the data and consumer should be able to use the data for its intended purpose.

Factors to consider:

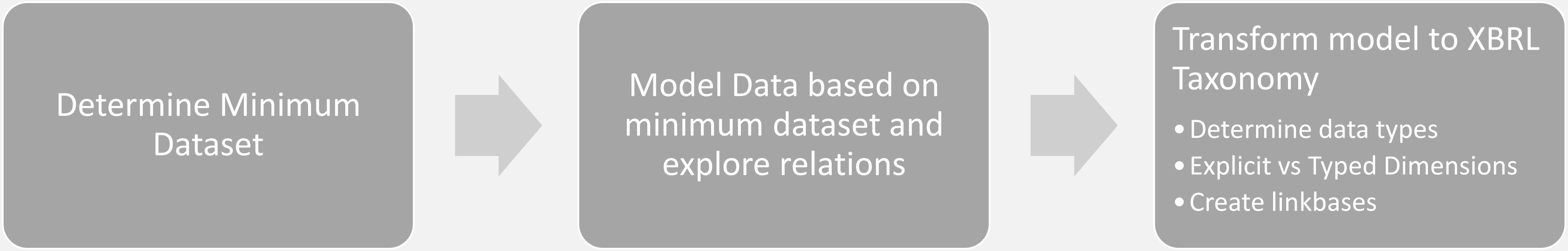
- Policy decision, such as extensibility.
- Functional requirements vs Non-functional requirement
- Understand use cases, how users interact with the systems to achieve their goals.
- Identify data to be transported, and systems that produce and consume the data.
- Stakeholders.
- Scope of the taxonomy (industry/sector wide or limited implementation).
- Resources required for the project.
- Support, maintenance requirements and change management
- Documentation and communication
- Balancing and prioritizing requirements from stakeholders and considering cost-benefits.
- Measures for success, such as accuracy and timeliness of data.

Functional requirements represents what a taxonomy is meant to do, while **Non-functional requirements** imposes constraints on system design. The focus in this stage is to identify the effect of requirements (functional and non-functional requirements) on the scope and design goals.

See TDH

4.5 Building Transport model

To build the taxonomy (Transport model), current datasets and dimensionality should be described. Functional requirements and non-functional requirements should be mapped to the data.



Determine Minimum Dataset

Model Data based on minimum dataset and explore relations

Transform model to XBRL Taxonomy

- Determine data types
- Explicit vs Typed Dimensions
- Create linkbases

Minimum dataset is the dataset free of redundant or extraneous information while representing all the necessary data. Current and legacy system may be a good source for determining minimum dataset(s).

Extensibility decision should be reflected in the taxonomy design and in determining allowable methods that users can extend a taxonomy, and how extensibility affects Comparability.

See Supporting materials, and TDH

4.6 Validation

As shown previously Validation ensures robust and accurate data. There are two levels of validation in XBRL:

- Basic Validation: Ensures reports are syntax valid, valid data types used and valid relationships used.
- Regulatory/Industry Requirements: Ensures that business rules are applied, methods used include software validation, XBRL Formula Validation, XULE validation and Data Quality Committees (issues and maintain data quality rules).

See TDH

4.7 Documentation

Taxonomy is a powerful tool, and it can only fulfill its purpose if users know how to use it. Taxonomy documentation is extremely important, it communicates the goals of the taxonomy and means to achieve those goals to all stakeholders.

Documentation include:

- **Taxonomy White Paper:** can be considered as an announcement of the taxonomy, with explanation of its purpose and justification for its development.
- **Taxonomy guide:** explains the taxonomy itself and logic behind it.
- **Preparer's guide:** provides preparers with information about the taxonomy's concepts and structures as needed to build XBRL reports.
- **Data Consumer Guide:** provides information and common use cases for data consumers. TDH provide detailed examples of documentation in section 8 page 111.

See TDH

See Taxonomy Creation Checklist