



Universidad  
Carlos III de Madrid  
[www.uc3m.es](http://www.uc3m.es)

## *TESIS DOCTORAL*

# Marco para la definición multidimensional del modelo de datos, estructura, taxonomías y reglas de la especificación XBRL

AUTOR:

**Ignacio-J. Santos Forner**

DIRECTORA:

Dra. Elena Castro Galán

CODIRECTOR:

Dr. Manuel Velasco de Diego

DEPARTAMENTO DE INFORMÁTICA

Leganés, diciembre 2015





Universidad  
Carlos III de Madrid  
www.uc3m.es

## TESIS DOCTORAL

### Marco para la definición multidimensional del modelo de datos, estructura, taxonomías y reglas de la especificación XBRL

**Autor:** Ignacio-J. Santos Forner

**Directora:** Dra. Elena Castro Galán

**Codirector:** Dr. Manuel Velasco de Diego

Firma del Tribunal Calificador:

Firma

Presidente		
Vocal		
Secretario		

Calificación:

Leganes, de de 2015.



## Agradecimientos

---

En Regulación y Supervisión, una importante norma es *Basilea II* (recomendación para leyes bancarias y regulatorias). Ésta está basada en tres pilares, que en resumen son:

- Requerimientos mínimos de capital (obtener los datos de los supervisados).
- Supervisar y revisar los datos obtenidos de los requerimientos del capital (explotar estos datos).
- Disciplina de mercado, es decir, hacer estos datos obtenidos y supervisados públicos.

Si se hace valer el símil de los tres pilares de Basilea, en esta tesis hay cuatro personas fundamentales (dos de ellas representadas en el primer *pilar*), sin las cuales no podría haberse realizado. Estas personas son:

- Mis tutores, Manuel Velasco y Elena Castro. Mención especial a Elena Castro, que ha hecho un esfuerzo sobrehumano en entender el dominio de los informes económico semánticos, muy cercano al mundo de las finanzas, pero también haciendo un esfuerzo enorme para entenderme, pues no soy el típico alumno que acaba de terminar los estudios universitarios.
- Ignacio Boixo con *XBRL España* y el *Banco de España*, que nos han facilitado a Elena y a mí todo el apoyo en medios e infraestructura necesarios para desarrollar esta tesis.
- Mi secretaria, Aurora Forner, con una paciencia infinita, trabajando incansablemente pasando a Word el texto en español e inglés, realizando todas las figuras del documento en PowerPoint, clasificando documentos, correos, etc. Ha hecho un trabajo excelente y brillante en el ordenador. En este punto no quiero olvidar a mi hermano, Félix Santos, ni a mi padre, Ignacio Santos, que trabajó muchísimo por sus hijos.

Hay mucha más gente que me ha apoyado, y espero no olvidarme de nadie. Ejemplo de esta contribución son: Víctor Morilla del *Banco de España*, Moira Lorenzo de *Atos* y Santiago Carrasco de *Indra*. También me gustaría resaltar a María José Gallego,

compañera del Banco de España, por su enorme trabajo de edición y maquetación de esta tesis doctoral, que le ha costado alguna noche de trabajo.

Durante la elaboración de esta tesis doctoral se han llevado a cabo cuatro trabajos finales de carrera, por lo que sería impensable no agradecer el esfuerzo realizado en un tema tan técnico y especializado. Estos cuatro estudiantes han sido, por orden cronológico:

- Jorge Valencia Martínez, *Modelo de meta-metadatos XBRL* (Valencia, 2011) (accesible en <http://hdl.handle.net/10016/13157>).
- Yolanda León Román, por su trabajo fin de grado titulado: *Automatización de la transformación del modelo de datos de la especificación XBRL en Base de datos* (León, 2012) (accesible en <http://hdl.handle.net/10016/16843>).
- Camelia Cid Canelada, *Mapa de carreteras del Modelo de Datos Multidimensional (MDM), haciendo uso de herramientas ROLAP, de FINREP 2012 y Solvency II (01-07-2012) con validaciones, Prueba de Concepto. Trabajo en progreso* (Cid, 2013).
- Abel Nieto Cano, *Validación de los informes semánticos económicos financieros* (Santos y Nieto, 2014, 2015; Nieto, 2015).

Por último, aunque por ello no menos importante, quisiera antes de terminar esta sección dejar escrito, para que *no se lo lleve el viento*, mi más profundo agradecimiento a los doctores D<sup>a</sup> Dolores Cuadra y D. Harith Al-Jumaily, por su implicación y aporte, sin el cual no hubiera llegado hasta aquí. Gracias.

# Índice de Contenidos

---

Agradecimientos .....	i
Índice de Algoritmos.....	v
Índice de Códigos .....	vii
Índice de Definiciones .....	ix
Índice de Figuras.....	xi
Índice de Reglas .....	xv
Índice de Tablas.....	xvii
Resumen.....	1
CAPÍTULO 1. Introducción .....	7
1.1. Motivación.....	7
1.2. Objetivos.....	8
1.3. Estructura del documento .....	11
CAPÍTULO 2. El estado del arte.....	15
2.1. Introducción .....	15
2.2. Evolución .....	19
2.3. Estado actual y futuro .....	22
2.4. Diseño de Informes semánticos .....	24
2.5. Model Driven Architecture .....	25
CAPÍTULO 3. Modelo de Meta metadatos de la especificación XBRL .....	29
3.1. Introducción .....	29
3.2. Modelo y objetos XML.....	33
3.3. Modelo de meta-metadatos XBRL .....	35
3.4. Objetos XML y XBRL utilizados en la definición de los meta-metadatos	

CAPÍTULO 4. Fundamentación y Conceptos Básicos de la Propuesta .....	59
4.1. El Universo del discurso y propuesta .....	59
4.2. Modelo Independiente de Computación (CIM) .....	61
4.3. Modelo Independiente de la Plataforma (PIM): reglas y definiciones ...	64
4.4. Cuestiones Semánticas acerca del diseño de los Metadatos en el PIM	76
4.5. Análisis en el Modelo específico de la Plataforma (PSM).....	78
CAPÍTULO 5. Fundamentación y Conceptos Básicos de la propuesta de fórmulas	83
5.1. El Universo del Discurso .....	83
5.2. Modelo de datos y Cálculos en XBRL .....	85
5.3. Introducción a la especificación XBRL formula.....	86
5.4. Definiciones y semántica de la especificación de fórmulas XBRL .....	90
5.5. Cuestiones semánticas en el PIM de la especificación de fórmulas XBRL .....	101
5.6. Implementación y prueba de concepto (POC), en el PSM.....	104
CAPÍTULO 6. Validación de la propuesta.....	113
6.1. Introducción a la validación .....	113
6.2. Validación del desarrollo de los metadatos de los informes semánticos 113	
6.3. Validación de los cálculos utilizando la especificación XBRL de fórmulas 127	
CAPÍTULO 7. Conclusión y trabajo futuro .....	143
7.1. Conclusiones .....	143
7.2. Trabajos Futuros.....	145
Bibliografía .....	149
Acrónimos .....	161
Glosario.....	167

## Índice de Algoritmos

---

Algoritmo 4.1 Transformación desde el CIM al PIM.....	74
Algoritmo 4.2 Implementación del modelo de metadata .....	79
Algoritmo 5.1 Creación de plantillas SQL .....	108
Algoritmo 6.1 Extracción del modelo de metadatos .....	115



## Índice de Códigos

---

Código 5.1 Plantilla I (SQL I), resumen del código .....	110
Código 6.1 Creación de las tablas del modelo en estrella ( <i>Nieto, 2015</i> ).....	120
Código 6.2. Busca ciclos en la jerarquía de los conceptos ( <i>Nieto, 2015</i> ).....	121
Código 6.3. Crea las tablas de dimensiones y carga sus atributos de dimensión ( <i>Nieto, 2015</i> ).....	122
Código 6.4 SQL script que crea las tablas del modelo en estrella .....	130
Código 6.5. Formula val_6610-sh-2 obtenida desde Arelle ( <i>Figura 5.6</i> ) .....	131
Código 6.6 Procedimiento de validación de la aserción <i>val_6610_sh_2_1</i> .....	138



## Índice de Definiciones

---

Definición 4.1 .....	62
Definición 4.2 .....	64
Definición 4.3 .....	66
Definición 4.4 .....	66
Definición 4.5 .....	66
Definición 4.6 .....	67
Definición 4.7 .....	67
Definición 4.8 .....	68
Definición 4.9 .....	68
Definición 4.10 .....	68
Definición 4.11 .....	69
Definición 4.12 .....	69
Definición 4.13 .....	70
Definición 4.14 .....	70
Definición 4.15 .....	71
Definición 4.16 .....	71
Definición 4.17 .....	72
Definición 4.18 .....	72
Definición 5.1 .....	91

Definición 5.2 .....	91
Definición 5.3 .....	92
Definición 5.4 .....	92
Definición 5.5 .....	92
Definición 5.6 .....	92
Definición 5.7 .....	93
Definición 5.8 .....	93
Definición 5.9 .....	95
Definición 5.10 .....	96
Definición 5.11 .....	97

## Índice de Figuras

---

Figura 2.1 Ejemplo de transmisión de datos de un cajero automático .....	15
Figura 2.2 Ejemplo de informe semántico contable .....	18
Figura 2.3 Recorrido de un informe financiero.....	21
Figura 3.1 Estructura de la especificación XBRL.....	30
Figura 3.2 Estructura de un <i>XBRL Instance Document</i> .....	31
Figura 3.3 El modelo de datos XBRL .....	32
Figura 3.4 Estructura XML versus XBRL .....	36
Figura 3.5 Tipo XBRL <i>contextScenarioType</i> .....	42
Figura 3.6 Elemento XBRL <i>linkbase</i> .....	44
Figura 3.7 Elemento XBRL <i>schemaRef</i> .....	45
Figura 3.8 Elemento XBRL <i>context</i> .....	46
Figura 3.9 Ejemplo del elemento XBRL <i>segment</i> .....	47
Figura 3.10 Elemento XBRL <i>dimension Item</i> .....	49
Figura 3.11 Elemento XBRL <i>hypercubeItem</i> .....	50
Figura 3.12 Elemento XBRL <i>xbrl</i> .....	51
Figura 3.13 Elemento XBRL <i>explicitMember</i> .....	54
Figura 3.14 Elemento XBRL <i>typeMember</i> .....	55
Figura 4.1 Diseño de informes semánticos utilizando el paradigma MDA.....	60
Figura 4.2 Ejemplo de informe con semántica.....	62

Figura 4.3 Ejemplo de plantilla .....	63
Figura 4.4 Modelo en estrella en el PIM .....	73
Figura 4.5 Ejemplo de tabla de hechos con atributos calculados.....	75
Figura 4.6 Resumen de los artefactos en UML del modelo de datos en el PIM .....	76
Figura 5.1 Estructura general de la transformación XBRL versus un RDBMS .....	84
Figura 5.2 Ejemplo de informe jerárquico .....	87
Figura 5.3 Diseño gráfico del informe de la <i>Figura 5.2</i> .....	88
Figura 5.4 Estructura jerárquica del informe de la <i>Figura 5.2</i> .....	89
Figura 5.5 Conversión automática de fórmulas entre instancias XBRL y bases de datos, utilizando MDA.....	90
Figura 5.6 Ejemplo de la aserción <i>val_6610sh-2</i> .....	99
Figura 5.7 Ejemplo de la aserción <i>val_6610sh-2</i> presentada gráficamente .....	100
Figura 5.8 Ejemplo con un hecho nulo .....	102
Figura 5.9 Hechos multivaluados .....	104
Figura 5.10 Obtención del API de validación .....	106
Figura 5.11 Proceso de validación .....	107
Figura 5.12 Proceso de aserción del algoritmo de la generación del proceso de validación .....	107
Figura 6.1 FINREP 2012, datos estadísticos en el MDM .....	116
Figura 6.2 Validación de los metadatos.....	117
Figura 6.3 Flujo de datos IS_DIMENSION. Ejecución procedimiento SQL de creación de la tabla de Hechos ( <i>Nieto, 2015</i> ) .....	118
Figura 6.4. Diagrama de base de datos en el MDM de FINREP 2014 ( <i>Nieto, 2015</i> ) .....	123

Figura 6.5. Diagrama..... 128

Figura 6.6 Test de validación para el informe 6610 ..... 139



## Índice de Reglas

---

Regla 4.1.....	64
Regla 4.2.....	68
Regla 5.1.....	94
Regla 5.2.....	94
Regla 5.3.....	94
Regla 5.4.....	94
Regla 5.5.....	94
Regla 5.6.....	95
Regla 5.7.....	95
Regla 5.8.....	95
Regla 5.9.....	96
Regla 5.10.....	96
Regla 5.11.....	97



## Índice de Tablas

---

Tabla 4.1 Definiciones y reglas en el MDM (El PIM) .....	65
Tabla 4.2 Transformación del PIM al PSM .....	78
Tabla 5.1 Desde el CIM al PIM.....	98
Tabla 6.1 Test de validación perteneciente al modelo en estrella UML .....	125
Tabla 6.2 Test de validación perteneciente al modelo relacional (RM) .....	126



## Resumen

---

Desde el año 2000 se ha producido un importante desarrollo de las aplicaciones basadas en XML y en los Almacenes de Datos (*Data Warehouse*). Como este tipo de aplicaciones necesitan interactuar con diferentes tecnologías y de igual modo con otras aplicaciones es, por tanto, un importante factor a tratar su **interoperabilidad**. De una manera paralela, en este mismo periodo de tiempo ha habido una evolución de los datos económicos que necesitan las empresas e instituciones financieras: a) incrementándose el número de estas entidades que necesitan más información; b) la necesidad de que esta información sea fiable y esté en tiempo. Que la información económica o financiera sea fiable no significa que no tenga errores de codificación o de transmisión, ya que en el año 2000 ya había procedimientos suficientemente fiables, sino que estos informes deben ser semánticamente correctos. Es en esta década en la que se observa la necesidad de que economistas y contables obtengan sus informes con contenido semántico y, es entonces cuando aparecen algunas especificaciones para la transmisión de información económica como puede ser el *eXtensible Business Reporting Language* (XBRL) o el *Statistical Data and Metadata eXchange* (SDMX). Los informes de ambas especificaciones tienen contenido semántico y están dedicadas al envío y tratamiento de información económica y/o estadística. Sin embargo, ambas especificaciones nacieron de expertos economistas o estadísticos, y tenían como único objetivo resolver su problemática, sin formalizar sus definiciones y reglas conceptuales, pues no hay modelo conceptual y, el modelo lógico existente está muy próximo al modelo físico, es decir, a su implementación.

El objetivo de esta tesis es formalizar la especificación de informes económico/financieros haciendo uso de la especificación XBRL, analizando y auditando esta especificación. Además, se estudia en profundidad el diseño de estos informes haciendo uso de la metodología del *Model Driven Architecture* (MDA), con el objetivo final de analizar su interoperabilidad. Por último, se muestran diversas pruebas de conceptos acompañadas de las diferentes validaciones de la propuesta de la tesis.

La tesis empieza analizando la necesidad de esta especificación (XBRL), su origen histórico y, en general, el estado del arte que ha llevado a la utilización de esta

especificación como un estándar *de facto*. Además, se estudia su evolución y utilización, tanto histórica, como en la actualidad, como en un futuro.

Una vez establecido el estado del arte de los informes semánticos económicos/financieros, se pasa a analizar la especificación XBRL. Después de dar una visión global del lenguaje o especificación XBRL se empieza a detallar éste y, por último, se muestra el conjunto de meta-metadatos usados en XBRL utilizando UML (*Unified Modeling Language*).

Esta tesis propone el diseño de los metadatos de los informes y de los propios informes. Pero, además, en este trabajo se propone estudiar las diferentes implementaciones de estos informes y su diseño en su totalidad. MDA provee un buen marco consolidado para la generación automática de código en el ciclo de vida del desarrollo del software, además de resolver los problemas de heterogeneidad e interoperabilidad entre sistemas con diferentes plataformas.

Uno de los principales retos de este trabajo de investigación es la formalización de la especificación en el modelo conceptual. También se analizan los diferentes tipos de diseños existentes: los que hacen un uso intensivo de las dimensiones (diseño Europeo) y aquéllos que utilizan menor número de dimensiones (diseño fundamentalmente de EEUU y Canadá). Este trabajo se basa fundamentalmente en los informes de la Supervisión/Regulación Europea, por tanto, se centra en los primeros. Esto es debido, por otra parte, a que el autor de esta tesis pertenece a diversos grupos de armonización de los informes de supervisión Europeos y es empleado del Banco de España. Esta institución es parte integrante del Sistema Europeo de Bancos Centrales (*European System of Central Banks, ESCB*), y éste a su vez del Banco Central Europeo (*European Central Bank, ECB*) y los bancos centrales nacionales europeos (*National Central Banks, NCBs*), que participan en el Eurosistema (Autoridad monetaria en la eurozona (*Eurosystem*)). Por tanto, el modelo conceptual propuesto en esta investigación es el Modelo de Datos Multidimensional (MDM). La propuesta está basada en los informes definidos por la Supervisión/Regulación europea. Estos informes son los datos requeridos a las entidades financieras o empresas supervisadas, siendo éste el *Universo del Discurso* (UD), también llamado Modelo Independiente de Computación (*Computation Independent Model, CIM*), utilizando la metodología del MDA.

Una vez realizadas las definiciones se pasa a analizar los procesos de validación de esta propuesta, mostrando algoritmos para obtener el MDM y la generación del proceso de aserción, o fórmulas, en el modelo de datos XBRL (*XBRL Data Model*, XBRLDM). Además, se mostrará su implementación en los modelos lógico y físico.

Toda esta propuesta está apoyada en diferentes Pruebas de Concepto (POC), presentadas en foros especializados de la Supervisión financiera y económica europeos. Estas pruebas están basadas en tres taxonomías europeas: *Financial Report 2008 (FINREP)*, *2012* y *Solvency II* (Eurofiling 2015). FINREP es una taxonomía auspiciada por la *European Banking Authority* (EBA). *Solvency II* codifica y armoniza la regulación de Seguros de la Unión Europea (UE) y es auspiciada por la *European Insurance and Occupational Pensions Authority* (EIOPA). En estas POC se analizan y muestran todas las propuestas de esta tesis. Además se presentan, a través de tablas, las diferentes pruebas que muestran la validación de esta propuesta en los diversos niveles de la metodología MDA.

También se analiza la interoperabilidad de los informes económicos/financieros, y su necesidad de mapeo en XBRL, hojas de cálculo, ACCESS, Sistemas Gestores de Bases de Datos (*Database Management System*, DBMS), XML y otros.

Por último, se presenta el trabajo de investigación a realizar en un futuro inmediato como consecuencia de esta tesis.



# CAPÍTULO 1

---

## Introducción



# CAPÍTULO 1. Introducción

---

## 1.1. Motivación

Los informes semánticos económico/financieros nacieron ante la necesidad de los reguladores y supervisores de obtener información de las empresas y entidades financieras de una forma fiable, en tiempo y fácilmente tratable por otras aplicaciones. Estos informes fueron diseñados por economistas, contables o financieros de acuerdo a sus necesidades, y los ingenieros informáticos se ocuparon de resolver sus estrictas necesidades.

De estas necesidades apareció la que luego sería la especificación XBRL (*Hamscher y Kannon 2000; Engel et al. 2008*), y que ha llegado a ser un estándar *de facto*. Cuando se analizó que esta especificación no resolvía todas las necesidades de las operaciones de cálculo de los usuarios expertos, se amplió la especificación (*Hernández y Wallis 2006; Engel et al. 2006*). Pero al igual que en su inicio se resolvieron las necesidades que demandaban los supervisores y reguladores, esta especificación ha ido creciendo. Actualmente, esta especificación es activamente utilizada por el *Board of Governors of the Federal Reserve System (FED)*, la *Securities and Exchange Commission (SEC)*, la *ShenZhen Stock Exchange (SZSE)*, la *Shanghai Stock Exchange (SSE)* (*JiMei et al., 2012; Jimei et al., 2013*), el *European Central Bank (ECB)*, la *European Banking Authority (EBA)*, la *European Insurance and Occupational Pensions Authority (EIOPA)*, la *Deutsche Börse*, el *Deutsche Bundesbank*, la *Companies House and HM Revenue & Customs (UK)* y la *Australian Prudential Regulation Authority (APRA)*, entre otras muchas instituciones y agencias. En USA, Canadá, Europa, China, etc. todas las entidades financieras y empresas cotizadas en bolsa tienen que informar obligatoriamente a las autoridades supervisoras y reguladoras utilizando la especificación XBRL. Los estados financieros son gobernados por estrictos requerimientos, tales como el *International Financial Reporting Standard (IFRS, 2015)*, o el *Generally Accepted Accounting Principles (GAAP)*. XBRL es utilizado en vez de XML, ya que los informes tienen una semántica específica dirigida por IFRS, o el GAAP, y estos informes pueden ser de gran tamaño.

Los datos de todos estos informes son almacenados en un gestor de bases de datos (DBMS) y es, por tanto, importante que el mapeo sea fácil, óptimo y automático. Pero es igualmente importante que el diseño de estos informes esté especificado de acuerdo a los modelos de diseño de datos estandarizados y, es en este último aspecto en el que la Unión Europea, a través del proyecto *Eurofiling* (2015), está centrando sus esfuerzos.

Por otra parte, cuando los informes son muy grandes (*XBRL reports* o *XBRL instance documents*), hay problemas en la implementación (especialmente en rendimiento en tiempo de ejecución) utilizando la tecnología XML existente, especialmente en las validaciones de los documentos.

En la actualidad, los departamentos de Tecnología de la Información (IT) de las agencias supervisoras o reguladores tienen enormes dificultades, ya que se crean taxonomías con una gran cantidad de conceptos (*Weller, 2015*) y reglas o restricciones de usuario (validaciones) y muchos informes llegan a tener errores de diseño y sin casos de prueba públicos. Esto es debido a la falta de un ciclo de vida del desarrollo del software completo, una falta de interoperabilidad (*Santos y Castro 2010*), y a su vez, a que los supervisores y/o reguladores están continuamente creando mayores y más complejos modelos.

Por lo tanto, a la vez que se va extendiendo la especificación surgen diversos problemas que hacen frenar su utilización universal. Es una especificación definida por economistas y financieros, implementada en XML, en cierta medida compleja, tanto en conocimiento como en infraestructura.

## 1.2. Objetivos

Resumiendo la problemática anteriormente mencionada:

- Estos informes son diseñados casi exclusivamente por economistas, contables, o financieros de acuerdo a sus necesidades.
- Se ha ido extendiendo la especificación según los usuarios expertos necesitaban nuevos elementos.
- La especificación ha ido creciendo y complicándose.

- Actualmente esta especificación es utilizada activamente.
- Los estados financieros son gobernados por estrictos requerimientos.
- XBRL es utilizado en vez de XML, ya que los informes tienen una semántica específica dirigida por IFRS, o el GAAP.
- Estos informes pueden ser de gran tamaño, ya que se crean taxonomías con una gran cantidad de conceptos y reglas o restricciones de usuario.
- Por otra parte, cuando los informes son muy grandes, hay problemas en la implementación (especialmente en rendimiento en tiempo de ejecución) utilizando la tecnología XML existente, especialmente en las validaciones de los documentos.
- Muchos informes llegan a tener errores de diseño.
- Sin casos públicos de prueba.
- Falta de interoperabilidad.
- Los supervisores y/o reguladores están continuamente creando mayores y más complejos modelos.
- Complejidad de la especificación tanto en conocimiento como en infraestructura.

Esta tesis tratará de resolver los problemas mencionados analizando el diseño de este tipo de informes independientemente de la plataforma, haciendo uso del paradigma *Model Driven Architecture* (MDA, 2015). El MDA estratifica el diseño en tres fases o niveles, y esto conlleva un desarrollo más fácil. Se analiza el *Universo del Discurso* (UD), que es el conjunto de definiciones contables, directivas europeas, circulares, etc. Este UD utilizando el paradigma MDA, es el CIM (*Computation Independent Model*). Desde el CIM se pasará al PIM (*Platform Independent Model*). En el PIM se deberán definir todos los conceptos y reglas de usuario en un modelo lo suficientemente amplio y formal para que la transformación al PSM (*Platform Specific Model*) sea lo más exacta y correcta, y consecuentemente, lo más automática posible. Se analizarán varias plataformas independientes, como el DPM (*Data Point Model*) utilizada por algunos departamentos de IT de los organismos y agencias de regulación o supervisión y el SQL más estándar o Universal. Y por último se mostrarán las reglas de codificación, *Code* en el paradigma MDA. Además se deben analizar diversas cuestiones semánticas que se deben tener en cuenta, ya que pueden ser intrínsecas al modelo o bien generar errores al heredar dichas

cuestiones, desde PIM al PSM. Este estudio evitará errores en la lógica de negocio o errores de ejecución. En definitiva, aplicando esta metodología se asegura que los informes están bien contruidos de acuerdo a la estructura y semántica definida y por lo tanto evita realizar informes erróneos, asegurando fiabilidad, confidencialidad y consistencia.

Además este estudio analizará la interoperabilidad de los modelos e incluso de los objetos físicos generados. De tal forma que desde una taxonomía XBRL se pueda obtener su implementación en el MDM y en un RDBMS, haciendo uso del paradigma MDA.

Por otra parte, se debe mostrar el ciclo de vida del desarrollo del conjunto de metadatos de los informes semánticos, pero también los propios informes.

Además, todo el trabajo de investigación que conlleva esta tesis estará acompañado de un conjunto de congresos y artículos de revistas, tanto académicos como relacionados con el dominio de usuarios expertos de los informes XBRL. Y se acompañará esta tesis con un conjunto de pruebas de concepto que deberá validar esta aproximación.

Adicionalmente, este trabajo de investigación tratará de clarificar y simplificar la compleja especificación XBRL, creada por expertos economistas y contables. Pero a su vez, poniendo la mirada en el abaratamiento de costes en la infraestructura y desarrollo de estos informes semánticos, resolución de problemas de rendimiento y ampliación de su campo de aplicación, de tal forma que pueda ser utilizado, por ejemplo, en el cálculo de una preventa, venta y disponibilidad de los productos de una empresa.

Para conseguir los objetivos anteriores, se debe analizar el dominio junto a los usuarios expertos, y utilizar un paradigma lo suficientemente amplio para el desarrollo del ciclo de vida del software, como puede ser el paradigma DMA y utilizar modelos, lo suficientemente precisos para la transformación entre ellos, como puede ser el MDM. El MDM se debe definir con definiciones formales, para que su transformación, dependiendo de la plataforma, sea lo más aproximado posible. Además este modelado debe ser lo más amplio y general, y así potenciar la interoperabilidad de este tipo de informes.

En definitiva, para la consecución de este trabajo, se deberán cubrir los siguientes objetivos parciales:

- Crear un ciclo de vida del desarrollo del software de los informes semánticos económicos/financieros, formalizando los conceptos y restricciones o reglas de usuario en el PIM, y las reglas de transformación en los diferentes niveles del paradigma MDA. Todo ello haciendo uso de la robustez que aporta la tecnología de modelado multidimensional.
- Al crear un ciclo vida del desarrollo del software se pretende resolver los problemas de nuevas extensiones y/o modificaciones del software.
- Implementación automática en SQL, ya que el destino final de los datos es una base de datos, lo que permite su utilización por agencias y compañías.
- Definición de unas reglas de transformación que resuelvan muchos de los problemas de interoperabilidad presentados a los usuarios de este tipo de informes.
- Al estar estos informes automáticamente descargados en base de datos (un RDBMS), pueden ser fácilmente utilizados por herramientas de Business Intelligence, y en general de análisis de los datos.
- Al analizar las anomalías o cuestiones semánticas se pueden resolver errores de diseño y construir conjuntos de pruebas.

### 1.3. Estructura del documento

Tras esta breve descripción el contenido del documento está estructurado en los siguientes capítulos:

El **capítulo 2**, que tratará el estado del arte de los informes semánticos económico/financieros contables: cómo ha sido su evolución, estado actual y futuro.

El **capítulo 3**, que analizará el modelo de meta-metadatos de la especificación XBRL.

Los **capítulos 4 y 5** que analizarán la propuesta:

- El **capítulo 4** analizará la especificación en sí, para el diseño de estos informes, pero sin el tratamiento de la especificación de fórmulas.
- El **capítulo 5** se centrará en el diseño y automatización de los cálculos (a través de las fórmulas) en estos informes.

El **capítulo 6**, que tratará de validar esta propuesta.

El **capítulo 7**, finalmente, mostrará las conclusiones y trabajo futuro de esta propuesta.

# CAPÍTULO 2

---

El estado del arte



## CAPÍTULO 2. El estado del arte

---

### 2.1. Introducción

La especificación XBRL permite la homogeneización de la información contable y financiera transmitida de forma electrónica. Hasta la década del 2000 este tipo de información se generaba en papel, o en formato electrónico, pero en cualquier caso sin contenido semántico. Si la información se transmitía en formato electrónico, se hacía a través de mensajes, más o menos grandes. En general estos mensajes podían ser transacciones económicas. El formato de estos mensajes podía ser tabular o separado por un delimitador, por ejemplo por un “;”. La *Figura 2.1* muestra un ejemplo de transmisión de datos de un cajero automático de una entidad financiera a un ordenador central de la entidad bancaria (el ejemplo mostrado es una simplificación, no es real, se trata de una aproximación).

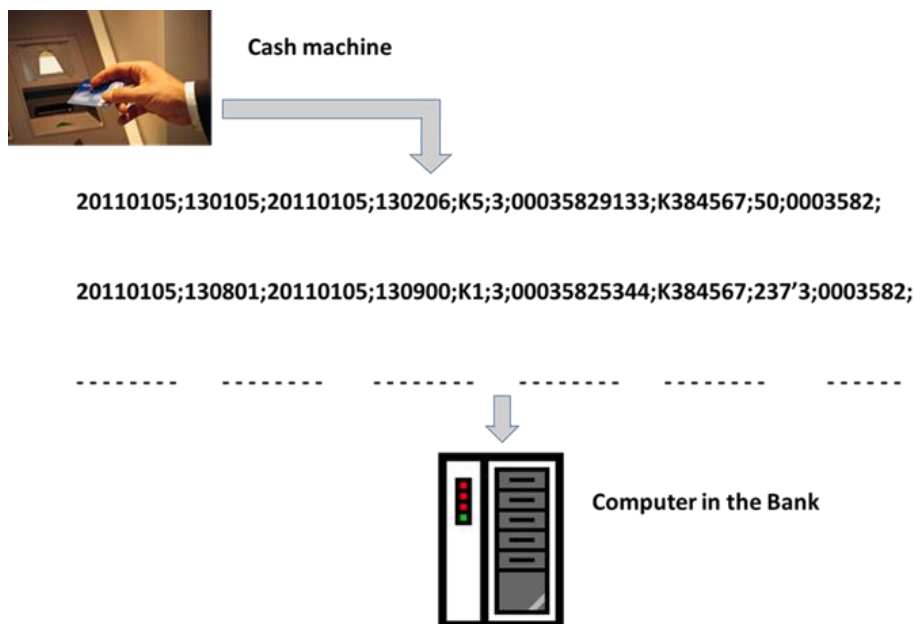


Figura 2.1 Ejemplo de transmisión de datos de un cajero automático

En este ejemplo se muestran dos transacciones. En la primera, un cliente retira dinero de su cuenta corriente: el importe es 50 €, su número de cuenta corriente es 0035829133, el código K5 indica que la operación es retirar dinero, a continuación del código de operación viene el código del resultado de la operación, 3 (que puede significar transacción aceptada). En la transacción se indica la fecha y hora de

comienzo y final de la misma, además se indica el código del cajero (K384567) y la entidad 0003582. La orden de la segunda transacción indica la obtención del saldo de la cuenta corriente, código K1, con un código de éxito 3, indicando que en la cuenta se dispone de 237'3 €.

En una compañía, agencia u organización hay siempre un intercambio de informes económicos o contables. Desde finales de los 90, este intercambio de informes (definidos como mensajes en Sistemas de Información (*Information Systems, IS*)) empezó a incrementarse. Las compañías necesitaban conocer el estado de sus órdenes de compra tan pronto como estuvieran disponibles, y ejecutar cálculos de preventa, venta y disponibilidad de los productos (*Lee et al., 1997; Wagenhofer, 2003*). Hasta entonces, el intercambio de mensajes se hacía a través de ficheros planos y con protocolos de comunicación pre-internet. Esta información estaba formada por un conjunto de registros, donde cada registro o línea consistía en un conjunto de caracteres separados por *comas*, *puntos y comas* o en *columnas*. Este conjunto de datos transmitidos no tenía semántica, únicamente la dada por la propia aplicación que trataba este mensaje. La aplicación daba por válidos los mensajes recibidos salvo problemas de sintaxis, como por ejemplo, que en el código de cuenta apareciera un código extraño, con lo cual el aplicativo pediría la retransmisión del mensaje o la anulación de éste.

En 1996 se formuló el protocolo de Intercambio de Datos Electrónicos (EDI, *Electronic Data Interchange*). EDI implica una secuencia de mensajes entre ordenadores. Este protocolo es muy utilizado en órdenes de compra, facturas, notificaciones de envío, etc. Estos mensajes simulan el intercambio de documentos en hojas de papel.

Otra plataforma muy utilizada mundialmente, es SWIFT (*Society for Worldwide Interbank Financial Telecommunication, Sociedad para las Comunicaciones Financieras Interbancarias Internacionales*; <http://www.swift.com>). Esta compañía nació en 1973, con el apoyo de 293 instituciones financieras. Esta plataforma empezó utilizando el protocolo X.25 (Estándar de la *International Telecommunication Union, ITU*, para el intercambio de paquetes en el *Wide Area Network, WAN*) y el servicio de mensajería FIN (*Financial message*), que constituye el núcleo del servicio de mensajería. FIN está basado en un estándar para los mensajes de servicios financieros, el ISO 20022 (2015) (*Standard for Financial Services Messaging, 2015*). Este estándar consta de un repositorio de metadatos en el que se describen los

mensajes, los procesos de negocio y el proceso para el mantenimiento del contenido del repositorio. Se utiliza de forma muy generalizada en organizaciones de servicios financieros como *FIX Protocol Limited (Financial Information eXchange, FIX)*, utilizado entre otros por NASDAQ (*National Association of Securities Dealers Automated Quotations*) en la Bolsa Americana. También utiliza este protocolo el ISDA (*International Swaps and Derivatives Association*), que es una organización de comercio de derivados y factura miles de millones de dólares en NASDAQ, además de SWIFT o VISA (*Visa Inc.*). Desde el año 2003, SWIFT ya utiliza en su mensajería IPSEC (*Internet Protocol Security*), e *InterAct* (protocolo para intercambiar mensajes de forma automatizada e interactiva, síncronos o asíncronos). Luego SWIFT con la nueva versión de FIN da servicio a aplicaciones-aplicaciones (A2A) y usuarios-aplicaciones (U2A).

En la actualidad se pueden encontrar en un lado los mensajes, que son un conjunto de datos económicos/financieros cortos y con poca semántica, y por otra parte los mensajes largos, o mejor dicho, informes económicos/financieros que necesitan de cierta semántica (*Hodge et al. 2004; Lin et al. 2005; Kernan 2009; Debreceeny et al. 2010; Prasanna et al. 2012*). Un informe “típico”, con cierta semántica, puede ser la concesión de un crédito de una entidad financiera en la que se necesitan unas ciertas garantías como: activos, finanzas, valores cotizados y la autorización de la concesión del crédito. Esto equivale a un conjunto de mensajes, pero relacionados entre sí por una semántica de negocio. Ejemplos típicos son: el SEPA (*Simple Euro Payment Area, 2014*), utilizado para pagos y transferencias; T2 (*Target 2, 2013*), para grandes pagos; y T2S (*Target 2 Securities, 2014*), para operaciones entre valores.

Según *Woodroof y Searcy (2001)* un sistema económico/contable debe tener integridad, confiabilidad, disponibilidad y ser mantenible. Además debe tener buenas prácticas de negocio, integridad en las transacciones, protección de la información y confianza o certeza en la emisión del informe (AICPA/CICA, 1999). Si estos informes no son semánticos, no pueden ser directamente automatizados en procesos internos o externos de la compañía o entidad (*Wagenhofer 2003; Williams et al. 2006*).

Por tanto, con lo visto anteriormente, puede decirse que los procesos de envío y recepción de información económica están perfectamente resueltos. Además esta información puede ser generada, recibida y procesada directamente por las aplicaciones involucradas.

Sin embargo, a principios de la década del año 2000 a los economistas no les bastaba con la tecnología desarrollada para la transmisión de los mensajes económicos. Es más, cuando los informes son grandes, estos suelen tener un fuerte contenido semántico, y antes de la década del 2000 no había prácticamente ninguna solución por parte de Sistemas de Información (IS). Por ejemplo, la *Figura 2.2* muestra un informe con las cuentas de liquidación de una entidad local a la agencia tributaria (Inteco 2009).

Liquidaciones – A1 – Ejercicio - 2009-01-01 - Entidad Local - - AA1 - - Valores en - Euros		
ANEXO 1		
CONCEPTOS	Obligaciones reconocidas	Intereses devengados
1. Intereses de la deuda Interior.	1.000,00	500,00
2. Intereses de préstamos del Interior.	2.000,00	1.000,00
3. Intereses de la deuda exterior.	3.000,00	1.500,00
4. Intereses de préstamos del exterior.	4.000,00	2.000,00
5. Depósitos, fianzas y otros.	5.000,00	2.500,00
6. Gastos de formalización, modificación y cancelación de deudas.	6.000,00	3.000,00
7. Diferencias de cambio.	7.000,00	3.500,00
<b>TOTAL</b>	<b>20.000,00</b>	<b>4.000,00</b>

**Figura 2.2** Ejemplo de informe semántico contable

En este informe los datos están relacionados entre sí, y por tanto se puede establecer una cierta semántica. Por ejemplo el total de las obligaciones reconocidas debe ser la suma de los conceptos 1 al 7, si no, es que el total es incorrecto. Sin embargo, un informe contable puede llegar a ser mucho más complicado y con una semántica mucho más compleja, ya que los conceptos pueden estar relacionados entre sí, dependiendo unos de otros. Esta tesis analiza la semántica de los informes económicos/financieros basados en la especificación XBRL.

## 2.2. Evolución

Desde el año 2001, los reguladores del Mercado de Valores empezaron a demandar mucha más información de negocio en sus informes, reduciendo la cantidad de tiempo en el que se procesaba éste. Mientras que el intercambio de información se hacía en papel, FAX o correo electrónico, estos informes eran mucho más complejos, requiriendo un análisis más profundo. Aunque durante años, ya estaban disponibles algunas soluciones para la transmisión de la información, ninguna era adecuada desde el punto de vista económico de los negocios. Dados los requerimientos, la información incluida en tales declaraciones económicas tenían que estar bien construidas, libre de errores de sintaxis y ser preparadas para su inmediato proceso y almacenamiento.

El *eXtensible Business Reporting Language* (XBRL), basado en el estándar XML, es una especificación utilizada para el intercambio de información económico/financiera. Las principales agencias e instituciones en el mundo, además de muchas empresas y organizaciones locales o estatales usan esta especificación. En USA, Canadá, Europa, China, etc., todas las entidades financieras y compañías cotizadas en bolsa tienen que dar informes obligatoriamente al Supervisor y al Regulador utilizando la especificación XBRL. XBRL se usa activamente por el Board of *Governors of the Federal Reserve System* (FED), la *Securities and Exchange Commission* (SEC), la *ShenZhen Stock Exchange* (SZSE), la *Shanghai Stock Exchange* (SSE) (JiMei et al., 2012; Jimei et al., 2013), el *European Central Bank* (ECB), la *European Banking Authority* (EBA, 2014), la *European Insurance and Occupational Pensions Authority* (EIOPA, 2014), la *Deutsche Börse*, el *Deutsche Bundesbank*, la *Companies House and HM Revenue & Customs* (UK) y la *Australian Prudential Regulation Authority* (APRA), entre otras muchas instituciones y agencias.

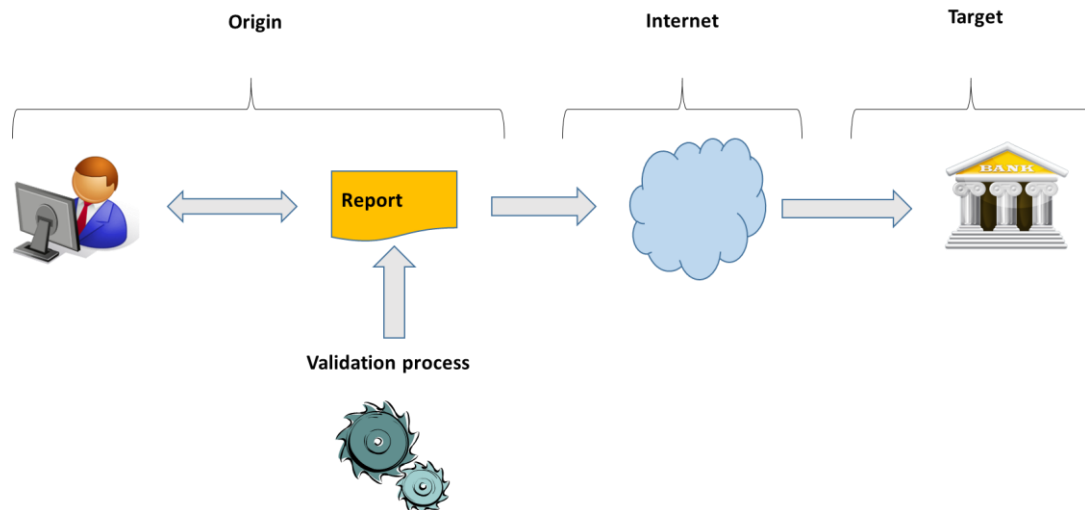
Los estados financieros o contables tienen requerimientos estrictos, tales como el *International Financial Reporting Standard* (IFRS 2015) o, como en el caso de España, el Plan General Contable, *General Chart of Accounts Plan* (Boletín Oficial del Estado 2007). Así, por ejemplo, los estados contables de las instituciones de crédito son estados específicamente definidos por una o más taxonomías, en las que se incluyen la estructura y semántica de estos estados. Como estos estados contables varían con el tiempo mediante decretos (nivel nacional), circulares (*Banco de España*) o directivas (*European Union*), conllevan problemas de versionado y localización. Las diversas leyes o directivas también varían con el tiempo, lo que

obliga a la adaptación de los informes a esas nuevas regulaciones y, al mantenimiento de un versionado. La localización implica que dependiendo desde dónde o quién pida un informe las reglas contables puedan variar, así por ejemplo los países de la *European Monetary Union (EMU)*, o un país en concreto. Los problemas de diseño de estas taxonomías (conjunto de conceptos y reglas) se elevan considerablemente cuando éstas se diseñan en conjunto. Por ofrecer un ejemplo, mientras que los requerimientos contables de flujo de capitales, hasta muy recientemente, estaban basados en Basel II (2004), como resultado de la actual crisis financiera, estas regulaciones han sido modificadas en Basel III (2010) (The Larosière Group (2009)). Además, la gran cantidad de requerimientos y datos conlleva un alto coste en recursos por parte de IS en muchas empresas, instituciones financieras, y económicas.

Después de la bancarrota de *Enron Corporation* en diciembre de 2001, los reguladores del mercado de valores empezaron a demandar una mayor información en los informes enviados a este supervisor en EEUU (*Securities and Exchange Commission, SEC 2015*). Pero, a su vez, se quería reducir el tiempo de proceso de estos informes. La SEC disponía de un sistema llamado EDGAR (*Electronic Data Gathering, Analysis and Retrieval system* [www.sec.gov/edgar.shtml](http://www.sec.gov/edgar.shtml)) que realizaba la recogida de informes en formato electrónico, los analizaba y disponía de un sistema de búsqueda automática. Sin embargo estos ficheros recogidos, que eran estados contables, estaban en cualquier formato, pdf, o escaneados. Actualmente EDGAR ya recoge informes en XBRL. Por tanto, se requirió que esta información estuviera libre de errores, seguros en su entrega y fáciles de procesar. Además, las empresas e instituciones financieras cada vez están más necesitadas de intercambiar información económico/financiera. Mientras este intercambio de información se hacía en papel, FAX o e-mail, el proceso de incorporación a una base de datos era lento y por tanto su posterior análisis se ralentizaba. Ya en el año 2000 había suficientes soluciones en IT para la transmisión de la información, pero hasta esa fecha, ninguna era adecuada desde el punto de vista económico o contable, ya que esta información no incluía contenido semántico, que era lo que se requería.

En Abril de 1998, el contable y auditor, Charles Hoffman, propuso la automatización del intercambio de información financiera desarrollando un prototipo para información contable y auditoría conocido como la base del XBRL (*Hamscher y Kannon 2000*). XBRL es una especificación basada en XML para informes financieros (*financial reporting*) (*Engel et al., 2008*). Los informes XBRL o *XML instance documents* son

generados desde varias fuentes y validados en origen, asegurando su validación sintáctica. Como se puede ver en la *Figura 2.3*, el informe es cumplimentado por el usuario y antes de ser enviado, es validado y si la validación es correcta se envía a destino para ser procesado.



**Figura 2.3** Recorrido de un informe financiero

Como la especificación XBRL representa información financiera, la cual es multidimensional, la localización lógica para su almacenamiento es un *Data Warehouse* (DW) (Golfarelli et al. 2001) (Boixo y Flores, 2005). Dados los requerimientos de las instituciones financieras y supervisores en general, esta información tenía que estar bien construida semánticamente, libre de errores de sintaxis (Debreceeny et al. 2010; Zhu and Fu 2009; Gräning et al. 2011; JiMei et al 2012; Jimei et al 2013 Nickerson et al. 2013). Además, estos informes o *reports* deben estar preparados para su inmediato proceso por IS y su subsecuente almacenamiento en un DW. Añadiendo mayor complejidad a esta tarea, el análisis de la información necesaria para un particular o general propósito de negocio, muchas veces requiere contrastar con información adicional al informe recibido (Callaghan et al. 2002; Di Giovanni, y Piazza, 2009; Loukis, y Charalabidis 2013). Los usuarios expertos, supervisores, reguladores, empresas en general necesitan analizar la información recibida desde varias fuentes, es por ello, la importancia de que los informes sean interoperables (Bennett 2013). En este último aspecto, cuando se trabaja con diferentes especificaciones o estándares, será necesario analizar el intercambio entre procesos y especificaciones con diferentes semánticas (Buder et al. 2009).

### 2.3. Estado actual y futuro

El uso actual de esta especificación está bastante extendido dentro del sector de la Supervisión o Regulación en casi todo el mundo. Sin embargo, poco a poco se van sumando otros sectores, aunque en general relacionados con instituciones, agencias, y organismos públicos. Esto es debido al alto coste de su implementación, tanto en recursos humanos, como de software, ya que se requiere expertos en la especificación y su software. Esta tesis analiza el desarrollo e implementación de los informes económicos financieros semánticos, para ello analizará la especificación XBRL formalizando sus definiciones, reglas y restricciones, estudiando su interoperabilidad, con el objeto de extender el uso de la especificación, abaratando estos costes al hacer uso de tecnología muy generalizada y conocida.

Otra especificación que utiliza informes semánticos es el *Statistical Data and Metadata eXchange* (SDMX 2015). Ésta es utilizada por el Banco Central Europeo (*European Central Bank*, ECB), Banco Internacional de Pagos (*Bank for International Settlements*, BIS), entre otras agencias e instituciones. Este estándar es principalmente usado para el intercambio de datos y metadatos estadísticos entre instituciones y para la diseminación de información estadística. Es una especificación desarrollada por instituciones internacionales, Bancos Centrales Nacionales (*National Central Banks*, NCBs) e institutos Estadísticos Nacionales (*National Statistical Institutes*, NSIs). Sin embargo, no es una especificación especialmente adaptada a otros sectores, por lo que su análisis semántico está fuera de ámbito de investigación de esta tesis.

Los Ingenieros informáticos de la supervisión bancaria europea vieron la necesidad de obtener un modelo de diseño para las taxonomías europeas. A partir de 2010, BR-AG y el Banco de España, haciendo uso de ingeniería inversa empezaron a utilizar el *Data Point Model (DPM)* (*Eurofiling 2015*). Ésta se comenzó a utilizar para las taxonomías del *Balance Sheet Items and Monetary Financial Institutions Interest Rates* (BSI-MIR 2010; Banco de España 2010) y fueron implementados por la compañía polaca de software financiero, BR-AG. Después fueron desarrolladas con el modelo DPM las taxonomías europeas COREP (está centrada en los informes consolidados y sub-consolidados de las reservas, del capital, y de los requerimientos

de capital basados en las directivas de la EU) y FINREP (Informes financieros consolidados y subconsolidados, basados en IAS (*International Accounting Standards*)/IFRS) (*Eurofiling 2012*). Cuando se obtenía la taxonomía, ésta se mapeaba a una hoja de cálculo *Excel*, pero ya en formato DPM, y después era analizada por los usuarios de negocios (los Reguladores) para validar la taxonomía. Dos años después se desarrolló un generador de plantillas *Excel* llamado *Data Point Modeller* (EXGEN 2012) desde las taxonomías XBRL. A continuación, con esta misma metodología fueron desarrolladas dos taxonomías para respectivamente *Cayman Islands Monetary Authority* y *Bermuda Monetary Authority*.

En el año 2013 el *Comité Europeo de Normalisation/European Committee for Standardization* (CEN 2014) estuvo trabajando en la estandarización del diseño del modelo de datos de los informes financieros/contables en Europa. En el CEN se formó un grupo que preparó una serie de *CEN Workshop Agreements* (CWA 2014) que promulgaba entregables para una base de la estandarización en Europa. Este grupo está dividido en tres subgrupos: *CWA1 Standard harmonization topics*, *CWA2 Metadata container and compliance tools*, y *CWA3 Standard regulatory roll-out package for adoption*. El grupo CWA1 tenía como función crear el modelo de metadatos para el diseño de los informes económico/financieros europeos. La primera meta de CWA era obtener un diccionario global de conceptos comunes a todos los informes (*reporting*). El grupo CWA1 trabajó para crear este grupo de conceptos, definiciones y tipos para reducir el presupuesto en la construcción de los informes (*reportings*) a los reguladores de instituciones financieras y también incrementar la coherencia y calidad de los datos (*Debreceeny et al. 2010*). El autor de esta tesis ha pertenecido a este grupo, especialmente en el subgrupo CWA1, aportando el mapeo del DPM al modelo ROLAP (*CWA 2014; Santos et al 2013; Santos 2013*). La meta de esta metodología es identificar el *data point* (el hecho) en el MDM (*Díaz 2012*). Una herramienta para el desarrollo del DPM es el *DPM Architect*, desarrollada por el Banco de España. Esta herramienta está desarrollada para que los supervisores puedan desarrollar taxonomías con DPM. Su objetivo es construir taxonomías utilizando exclusivamente la especificación XBRL, y también proveer de una herramienta gráfica para la creación de fórmulas (*Morilla 2008; Santos y Castro 2011a, 2012a; Santos et al 2016*). El Banca d'Italia desarrolló el predecesor del DPM, el *Matrix Schema* (MS) (*Romanelli 2007*), el cual contribuyó a la iniciativa de *Eurofiling*. MS está basado en una hoja de cálculo donde las filas muestran *dimensiones* y *domain-members* (atributos de dimensión y calculados en MDM), y las columnas muestran las *primary ítems* (dimensión base, será vista más adelante). Las

intersecciones de las filas y las columnas muestran los hechos, después de los cuales, en el modelo de datos XBRL (XBRLDM) se construyen los hipercubos. También se muestran en este modelo lógico las jerarquías (agregadas en el MDM), aunque no sus reglas.

En el momento de escribir esta tesis, los informes que utilizan esta especificación se encuentran con un problema hasta ahora no resuelto, que es el de los informes muy grandes. El problema es que como los *XBRL instance documents* son realmente documentos XML, para su validación requieren estar en memoria y siempre hay un límite. Esto va a implicar, por ejemplo, que en el caso de grandes entidades que tienen que reportar al Supervisor, si no se tiene cuidado con el tamaño del informe puede que no pueda ser tratado, o bien que el rendimiento de la máquina validadora caiga exponencialmente. Para salvar este problema se intenta dividir el informe de tal forma que los hechos y el contexto requerido estén puestos secuencialmente, de tal forma, que no pueda liberar memoria. Sin embargo, esta tecnología *in-stream*, requiere que el diseñador de la taxonomía agrupe los hechos de una manera manual.

## 2.4. Diseño de Informes semánticos

La necesidad de inclusión de semántica en los informes XBRL ha sido ampliamente analizada por diversos autores. En *Callaghan et al., (2002); Callaghan et al., (2006)* se muestran las primeras ideas de los informes semánticos en las finanzas utilizando UML y la especificación XBRL. *Sugumaran et al. (2002)* presenta una evolución de los trabajos indicados anteriormente de Callaghan y dibuja la visión del negocio para este tipo de informe. Además, *Callaghan et al. (2006)* muestra, con un ejemplo, la idea global de los negocios con este tipo de informes. *Nickerson et al. (2013)* presenta a muy alto nivel el desarrollo de taxonomías, pero no muestra de forma detallada cómo se realiza el diseño y sólo se queda en los conceptos y no en las restricciones de usuario. Aunque esta tesis doctoral no trata del tema de las ontologías, éste está presente en el trabajo de *Declerck y Kriege (2003)* quienes analizan un esquema de translación para la base ontológica de XBRL en OWL (*Web Ontology Language*), la cual era una extensión de los principios de contabilidad alemana. *Silveira et al. (2007)* presenta los datos de negocio y clasifica estos datos. *Lara et al. (2006)* muestra la base de la antología XBRL utilizando OWL. *Spies (2010)* presenta el proyecto MUSING (*Multi-industry semantic-based next generation business IntelliGence*), con

este proyecto se hace un análisis de la extracción de la ontología de las taxonomías XBRL, sin embargo este trabajo de investigación no propone su implementación automática. *JiMei et al. (2012)* describe la estructura del XBRLDM, y muestra como objetivo la obtención de una ontología de los datos XBRL.

El diseño de estos informes es extremadamente general y los diseños de las taxonomías (conjunto de conceptos y reglas para los informes semánticos) pueden variar enormemente. En Europa el diseño hace uso intensivo de las dimensiones (*Boixo y Flores, 2005; Felden 2007*). La utilización de dimensiones hace que el proceso de diseño sea más fácil, ya que si el número de dimensiones en el modelo conceptual es alto, es semánticamente más rico y la transformación a la base de datos es más fácil. Adicionalmente, es posible reducir el número de dimensiones a través de vistas en el modelo lógico (el PSM), según el criterio de implementación. En la actualidad, estos informes utilizan la especificación XBRL a través de taxonomías. Es intención de esta tesis, utilizando el paradigma MDA, ayudar al diseño de estas taxonomías o informes semánticos en XBRL, o en otras taxonomías o informes (XML, SDMX, iXBRL, hojas de cálculo, etc).

Existen algunos trabajos previos de mapeo del XBRLDM al MDM publicados, como *Snijders (2005)*, *Santos, y Castro (2011c, 2012)*, *Santos, y Nieto (2014, 2015)*, *Santos et al. (2013)*, y en *Openfiling (2015)*. La contribución de esta tesis va más allá, ya que se indican las reglas de transformación al MDM con formalización matemática (en el PIM). Además se presenta una alternativa de validación basada en base de datos en vez de herramientas XML y potencialmente con mejor rendimiento del *report* o *XBRL instance document*. Finalmente, un conjunto de pruebas de conceptos (POC) de este nuevo trabajo, el mapeo entre el XBRLDM a un RDBMS y viceversa, se han presentado en diversos foros (*Santos y Castro, 2011b, 2011a*), también se incluirán en este estudio. Además, el mapeo presentado no sólo servirá para validar la propuesta, sino también como herramientas de *open-source* (código abierto) para compañías e instituciones financieras con recursos limitados.

## 2.5. Model Driven Architecture

*Model Driven Architecture* de OMG (MDA, 2015) provee una aproximación abierta al reto del cambio tecnológico y de los negocios. Se basa en OMG, el MDA separa la

lógica de negocio de la aplicación de la tecnología. El modelo independiente de la plataforma de una aplicación o sistemas integrados está basado en UML. El ciclo de vida del desarrollo del software es el proceso donde los requerimientos están dentro de un dominio, los sistemas que conllevan los requerimientos deben ser considerados soluciones residentes dentro de un entorno, y esto envuelve la comprensión del problema, resolución e implementación de la solución. Cada proceso del ciclo de vida del desarrollo del Sistema envuelve los tipos siguientes de las actividades del ciclo de vida (*Si Alhir, 2015*): actividades de recolectar los requerimientos, actividades de análisis, actividades de diseño, actividades de implementación, actividades de testeo y actividades de desarrollo. El modelo MDA está relacionado con múltiples estándares, incluyendo el UML, el *Meta-Object Facility (MOF)*, *XML Metadata Interchange (XMI)*, *Enterprise Distributed Object Computing (EDOC)*, el *Metamodelo de Ingeniería de Procesos del Software (SPEM en inglés)*, y el *Common Warehouse Metamodel (CWM)*. Una de las principales metas del MDA es separar el diseño de la arquitectura. El MDA es un marco abierto y de arquitectura independiente del fabricante asociada con los estándares de OMG con los sistemas de desarrollo del ciclo de vida a través de varios dominios y tecnologías. El MDA soporta extensamente diferentes tipos de dominios de aplicación y plataformas tecnológicas. El MDA posibilita la transformación o conversión de modelos independientes de la plataforma (el PIM) para producir modelos específicos de la plataforma (el PSM) usando transformaciones. Dentro del proceso del ciclo de vida del desarrollo del Sistema, el MDA aplica el PIM y el PSM para sostener y apoyar la inversión de los requerimientos, tecnologías, y el ciclo de vida que hace de puente en el hueco entre ellas como un cambio independientemente. Tal aproximación conlleva, generalmente, una gran flexibilidad de implementación, integración, mantenimiento, testeo y simulación, aparte de portabilidad, interoperabilidad y reusabilidad. Hay algunos trabajos que intentan resolver el ciclo de vida de desarrollo de los informes XBRL utilizando UML como *Mendez-Nunez y Trivino (2010)* y *Callaghan et al. (2002 y 2006)*. Esta tesis propone el desarrollo de los informes semánticos utilizando MDA.

# CAPÍTULO 3

---

Modelo de Meta  
metadatos de la  
especificación XBRL



## CAPÍTULO 3. Modelo de Meta metadatos de la especificación XBRL

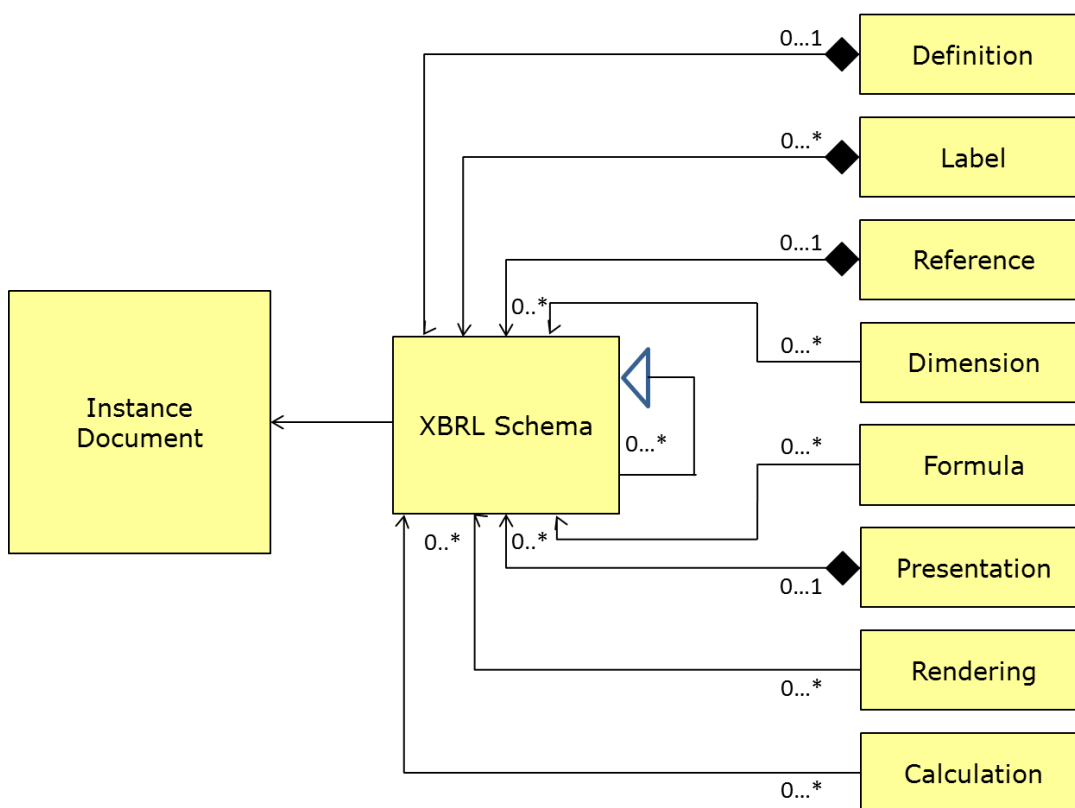
---

### 3.1. Introducción

En este capítulo se introduce la especificación XBRL y se analiza de forma detallada. Se comienza analizando el meta metamodelo.

Un meta metamodelo se entiende como el conjunto de definiciones y reglas de una especificación. Por tanto, nuestro meta modelo no es el meta metamodelo propuesto por OMG. Como se ha indicado, XBRL es una especificación basada en XML, y por lo tanto, se tendrán muchas referencias a él, sin embargo, en esta tesis se propondrán un conjunto de definiciones y reglas en el MDM a partir de la especificación.

Una taxonomía es una ciencia que trata de los principios, métodos y fines de clasificación (definición de la *Real Academia de la Lengua Española*, edición 22ª). En la especificación XBRL, una taxonomía es el conjunto de documentos donde se definen los conceptos que posteriormente se verán representados en los informes (*reports*) económico/financieros, y cómo se relacionan entre ellos. Este conjunto de conceptos se ven presentados de una forma estructurada junto a los datos. Por tanto, XBRL permite modelar la información y el significado semántico comúnmente requerido en los informes económico/financieros. Utiliza una sintaxis XML y las tecnologías relacionadas como *XML Schema*, *XLink*, *XPath* y *Namespaces* para proveerlo de significado semántico (*Hernández-Ros, 2009*).



**Figura 3.1 Estructura de la especificación XBRL**

Un informe que hace uso de esta especificación, como se muestra en la *Figura 3.1*, consta de un *XBRL instance* (*XML instance document*), el cual es el propio informe o *report*, también llamado *XBRL instance document* que consta de un conjunto de *XML Schemas* o *XBRL Schemas*, llamados *Discoverable Taxonomy Set* (DTS), los cuales especifican los conceptos, sus reglas y restricciones (*Engel et al., 2008; Santos y Castro, 2011a, 2011c*). Cada *XBRL Schema* consta de un conjunto de roles o *Linkbases* en el modelo de datos XBRL (*XBRL Data Model, XBRLDM*). Estos roles se muestran en la *Figura 3.2* como un conjunto de constructores de un *XBRL instance document*.

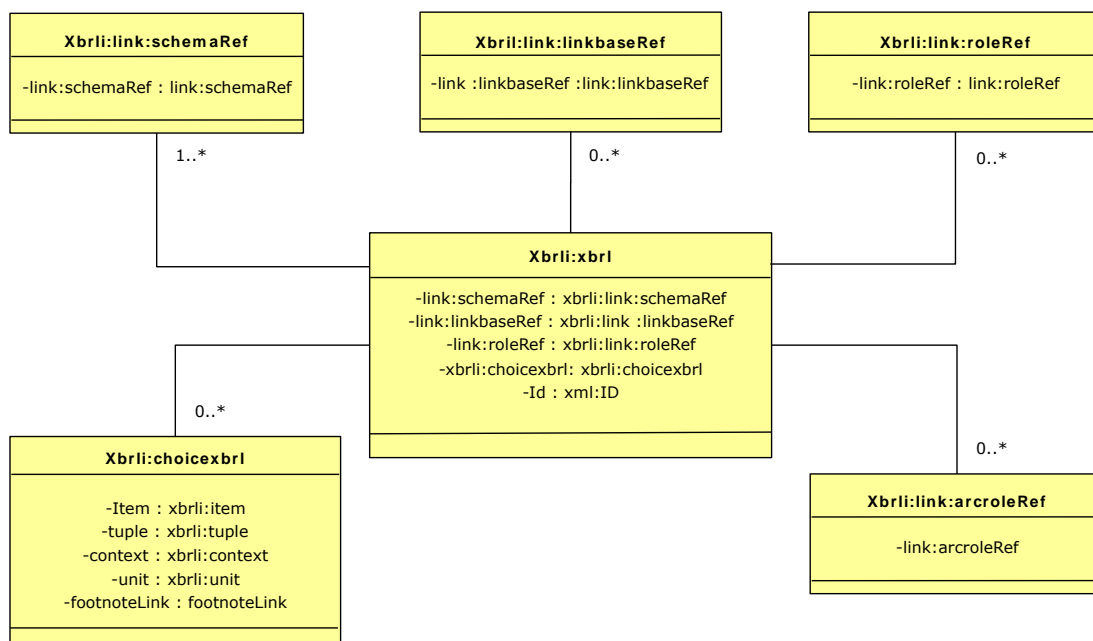


Figura 3.2 Estructura de un XBRL Instance Document

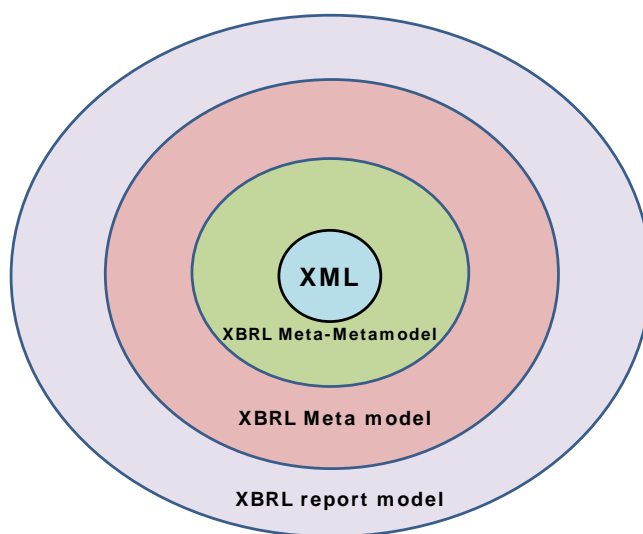
Los roles se definen como:

- **Definition.** Permite el establecimiento de relaciones entre conceptos, y éstos son fijados por la lógica de negocio.
- **Label.** Este *linkbase* da los nombres a los conceptos definidos en diferentes lenguajes.
- **Reference.** Permite la inclusión de referencias a textos legales, sin embargo, el texto no es incluido, sólo la referencia.
- **Dimension.** Permite combinaciones válidas de dimensiones.
- **Formula.** Soporta expresiones complejas basadas en *XPath*, las cuales pueden ser aplicadas a *instance documents* para validar su información.
- **Presentation.** Establece relaciones jerárquicas entre elementos (conceptos) definidos en un esquema. Se utiliza como una guía para la presentación, porque los conceptos están jerarquizados, como se analizará más adelante. Esta jerarquía conlleva un orden, el cual se establece por la lógica de negocio.
- **Rendering.** La presentación de los informes, renderización, está incrustada en la especificación HTML. Y por otra parte, registra las transformaciones especificadas, de tal forma que el informe sería siempre comprensible, sin dar datos incorrectos. En estos momentos no es todavía un estándar.

- **Calculation.** Provee de relaciones simples de cálculo entre diferentes elementos (conceptos). Sólo son posibles operaciones sobre una dimensión, ya que está basada en la jerarquía de los conceptos.

Cada XBRL o XML *Schema* debe tener de 0 a 1 ó N *roles*, dependiendo del tipo de roles (Santos y Castro, 2011a, 2011c). En esta especificación un *Primary ítem* define un esquema de hecho, sin dimensiones en el MDM (Martín Quetglás, 2006; Hernández-Ros y Wallis, 2006; Engel, et al., 2008; Schmehl and Ochocki, 2009). Por ejemplo, un *primary ítem* puede ser un activo (*asset*), pero no se conoce de antemano, si este activo pertenece a un patrimonio neto (*Equities*), o a un cierto tipo de riesgo (*Risk*), etc.

En la *Figura 3.3*, se muestran las capas de la especificación, del modelo de datos XBRL (XBRL Data Model, XBRLDM).



**Figura 3.3** El modelo de datos XBRL

XBRL está basado en XML y por tanto en la *Figura 3.3*, la capa XML se encuentra en el centro de la figura. Sobre ella se construye el conjunto de definiciones y reglas que hacen que pueda ser construida la especificación. En la siguiente capa se encuentra el metamodelo, que son el conjunto de taxonomías que definen un informe. Y la capa más exterior es el propio informe.

En las siguientes secciones se analiza el conjunto de elementos XML necesarios, o sobre los que se sustenta el XBRLDM.

## 3.2. Modelo y objetos XML

Un documento XML sirve para almacenar información (*Morrison, 1999*). En la década de los 70 se pretendía estructurar los documentos en una forma organizada para facilitar su intercambio y manipulación. Con el patrocinio de IBM se creó el Lenguaje de Marcado Generalizado (*Generalized Markup Language, GML*). GML se usó para producir libros, informes y otros documentos. Después, también con la colaboración de IBM, se definió y adoptó el estándar SGML (Lenguaje Estándar de Marcado Generalizado, *Standard Generalized Markup Language*) para el intercambio de información (ISO-8879). Uno de los principales usos de este estándar fue para dar formato y mantener los documentos legales de IBM. SGML se basa en el marcado y etiquetado de documentos, pero no se definen las etiquetas. La estructura de un documento se basa en la DTD (*Document Type Definition*) (*Rusty Harold, 2004*). SGML es un metalenguaje con una serie de normas para definir las etiquetas. Un documento SGML consta de tres partes (*Sperberg-McQueen y Burnard, 2002*): la declaración del SGML, la declaración del documento (DTD) y la propia instancia.

La declaración indica el tipo de caracteres, delimitadores y características del documento. La DTD define la estructura del documento; contiene reglas de nombrado, descripción de contenidos y uso de los elementos. La DTD interpreta un documento y sirve de validador. Y por último, la instancia, que es el propio informe o documento que el usuario de negocio requiere.

Una evolución de la especificación SGML es la especificación *HyperText Markup Language* (HTML). En febrero de 1998 aparece formalmente la especificación XML 1.0 (*eXtensible Marked Language*), como evolución de HTML y apoyada por SGML. XML se definió como un sistema para definir, validar y compartir formatos de documentos de la *Web*. HTML se centra en la presentación de los datos y XML modela los datos (*w3schools, 2015*).

La DTD establece la gramática de un vocabulario XML y determina la estructura de documentos XML. Una DTD consta de elementos, atributos, entidades y notaciones.

La DTD puede estar incrustado en el documento o referenciado en una fuente externa.

Un documento **es válido** cuando cumple los siguientes requisitos (*Morrison, 1999*):

- El documento está bien formado.
- El nombre del elemento raíz es el mismo que el nombre de la declaración de tipo de documento.
- El documento tiene una DTD en la que se declaren los elementos, atributos y entidades como se utilicen en el documento interno, externo o ambos.
- El documento cumple la gramática establecida por la DTD.

En XML un elemento provee estructura propia, permite definir una estructura lógica en el documento. En el siguiente ejemplo, se tiene un documento XML. Este documento comienza por una declaración, que identifica la versión utilizada, así como el código de 16 caracteres. Después debe aparecer un elemento raíz, que contendrá el resto de los objetos del documento.

```
<?xml versión="1.0" encoding="UTF-8" standalone="yes"?>
  <película>
    <titulo> El Señor de los Anillos </titulo>
    <director> Peter Jackson </director>
  </película>
```

### Ejemplo 3.1 Ejemplo de documento XML

En la primera línea del *Ejemplo 3.1* se indica la versión del documento, el código utilizado y si lleva asociado una DTD.

Los atributos llevan asociadas características del elemento y no contienen información compleja, como se muestra en el *Ejemplo 3.2*:

```
<pelicula genero="ciencia ficción" Categoría="superproducción"> El Señor de los Anillos
</película>
```

### Ejemplo 3.2 Ejemplo de atributos en un documento XML

Los atributos pueden ser un conjunto de valores de una lista, o valores predeterminados, y además pueden tener tipo de datos.

Una DTD tenía ciertas limitaciones como la sintaxis, no fácilmente comprensible, y la falta de soporte para el espacio de nombres *XML Xchema*, basado en W3C, el cual define un vocabulario XML (*XML.com, 2015*). *XML Schema* define la estructura de los documentos XML, así como los tipos de datos válidos para cada elemento y atributo.

La estructura *XML Schema* es modular, permitiendo la reutilización de módulos definidos. Además, se establece una relación de jerarquías entre estos módulos. Existen dos tipos de objetos en un esquema XML: los tipos simples y los tipos complejos. Los tipos simples son los que no tienen elementos hijos, ni atributos. Los predefinidos por XML son *double, string, boolean*, etc. Los tipos lista o enumerados y, las uniones de tipos predefinidos son también tipos simples. Los tipos complejos son aquellos que tienen elementos hijos y/o atributos.

Relacionadas con XML aparecen nuevas especificaciones como son XPath, XQuery, XPointer, y XLink<sup>1</sup>, con el auspicio de W3C, *World Wide Web Consortium*.

### 3.3. Modelo de meta-metadatos XBRL

Como se ha indicado en este capítulo, especialmente en las *figuras 3.1 y 3.2*, un informe de la especificación XBRL, llamado *XBRL instance document*, está formado por una serie de *Linkbases* o *roles* (desde el punto de vista del modelado de datos), que forman una taxonomía y la propia instancia. Pero igualmente se podría decir que un *XBRL instance document* está formado por las especificaciones XBRL, basadas en la especificación XML y un conjunto de *XML Schemas*, como se puede observar en la *Figura 3.4*:

---

<sup>1</sup> Ver tabla de *acrónimos*.

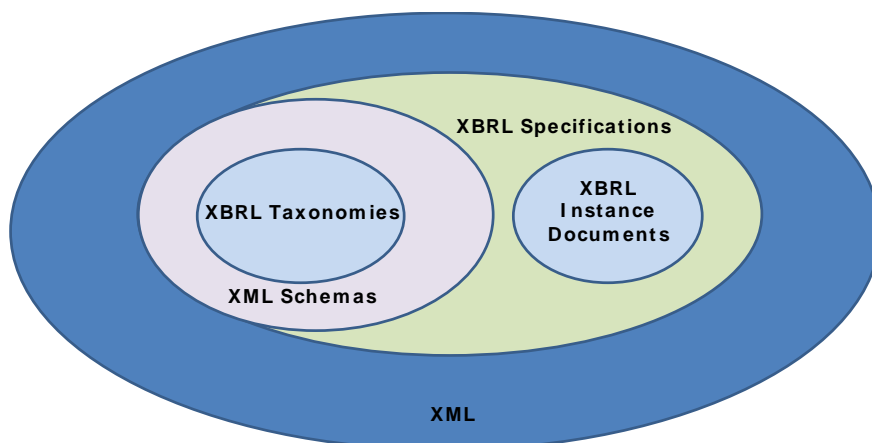


Figura 3.4 Estructura XML versus XBRL

Luego un informe XBRL está formado por diferentes niveles (*XBRL España, 2006*):

- Contenido del informe, formado por un conjunto de hechos contables que se ajustan a una norma contable vigente.
- Definición de los conceptos y de las relaciones externas e internas de los propios conceptos (taxonomías).

Si se analiza la arquitectura de la especificación XBRL se pueden definir **cuatro** capas:

- **Capa conceptual.** En la que se definen los conceptos, su sintaxis y su semántica.
- **Capa de relaciones.** En esta capa se detallan las relaciones entre conceptos.
- **Capa de las taxonomías.** Llamada DTS (*Discoverable Taxonomy Set*), que es la unión de taxonomías y documentos *Linkbases* para interpretar un documento instancia (el informe). Un DTS es el conjunto de *XBRL Schemas* o *XML Schemas* y sus ficheros XML asociados o roles que constituyen una taxonomía y validan un documento instancia XBRL.
- **Capa de extensiones.** Son ampliaciones, actualizaciones y modificaciones de la taxonomía.

### 3.4. Objetos XML y XBRL utilizados en la definición de los meta-metadatos

Seguidamente se analizan los objetos XML y XBRL que forman la especificación XBRL, aunque sin detallarlos, ya que estos se referencian ampliamente en *Valencia (2011)*, y en *Openfiling (2015)*.

Los **objetos XML** utilizados para **definir el modelo de meta-metadatos XBRL** son:

- Objeto *SimpleType*
- Objeto *ComplexType*
- Objeto *element*. *Object attribute*
- Objeto *simpleContent*
- Objeto *complexContent*
- Objeto *attributeGroup*
- Objeto *substitutionGroup*
- Objeto *restriction*
- Objeto *extension*
- Objeto *anyAttribute*
- Objeto *choice*
- Objeto *sequence*
- Data type *NCName*
- Data type *NMTokens*
- Data type *anyURI*
- Data type *processContent*

Seguidamente se detallan los objetos XBRL sobre los que se construye la especificación. Estos objetos están estructurados en: atributos, grupos de atributos, tipos simples, tipos complejos y elementos.

## Objetos XBRL “Atributos”

- *xlink:actuate*
- *xlink:arcrole*
- *xlink:from*
- *xlink:href*
- *xlink:label*
- *xlink:role*
- *xlink:show*
- *xlink:title*
- *xlink:to*
- *xlink:type*
- *xbrldt:closed*
- *xbrldt:contextElement*
- *xbrldt:typeDomainRef*
- *xbrldt:usable*
- *xbrldt:targetRole*
- *xbkli:balance*
- *xbkli:periodType*

## **Objetos XBRL “Grupos de atributos”**

- *xlink:titleType*
- *xlink:simpleType*
- *xlink:resourceType*
- *xlink:locatorType*
- *xlink:extendedType*
- *xlink:arcType*
- *xbri:factAttrs*
- *xbri:itemAttrs*
- *xbri:essentialNumericItemAttrs*
- *xbri:nonNumericItemAttrs*
- *xbri:numericItemAttrs*
- *xbri:tupleAttrs*

## **Objetos XBRL “Tipos simples”**

- *xlink:TypeEnum*
- *xlink:nonEmptyURI*
- *xlink:actuateEnum*
- *xlink:showEnum*
- *xl:useEnum*
- *xbri:DateUnion*
- *xbri:decimalsType*
- *xbri:monetary*
- *xbri:nonZeroDecimal*
- *xbri:precisionType*
- *xbri:pure*

- *xbri:shares*
- *xbri:contextElementType*

### **Objetos XBRL “Tipos complejos”**

- *xl:titleType*
- *xl:simpleType*
- *xl:resourceType*
- *xl:locatorType*
- *xl:extendedType*
- *xl:arcType*
- *xl:documentationType*
- *xbri:anyURIType*
- *xbri:NCNameItemType*
- *xbri:base64BinaryItemType*
- *xbri:booleanItemType*
- *xbri:byteItemType*
- *xbri:dateItemType*
- *xbri:dataTimeItemType*
- *xbri:decimalItemType*
- *xbri:doubleItemType*
- *xbri:durationItemType*
- *xbri:floatItemType*
- *xbri:fractionItemType*
- *xbri:dayItemType*
- *xbri:gMonthDayItemType*
- *xbri:gMonthItemType*
- *xbri:gYearItemType*
- *xbri:gYearMonthItemType*

- *xbri:exBinaryItemType*
- *xbri:integerItemType*
- *xbri:intItemType*
- *xbri:longItemType*
- *xbri:measuresItemType*
- *xbri:monetaryItemType*
- *xbri:negativeIntegerItemType*
- *xbri:nonPositiveIntegerItemType*
- *xbri:normalizedStringItemType*
- XBRL *xbri:positiveIntegerItemType*
- XBRL *xbri:pureItemType*
- *xbri:QNameItemType*
- XBRL *xbri:sharesItemType*
- XBRL *xbri:shortItemType*
- XBRL *stringItemType*
- XBRL *xbri:timeItemType*
- *xbri:tokenItemType*
- *xbri:unsignedByteItemType*
- *xbri:unsignedIntItemType*
- *xbri:unsignedLongItemType*
- *xbri:unsignedShortItemType*
- *xbri:contextEntityType*
- *xbri:contextPeriodType*
- *link:arcroleRef*
- *link:arcroleType*
- *link:calculationAFc*
- *link:calculationLink*
- *link:definition*
- *link:definitionArc*

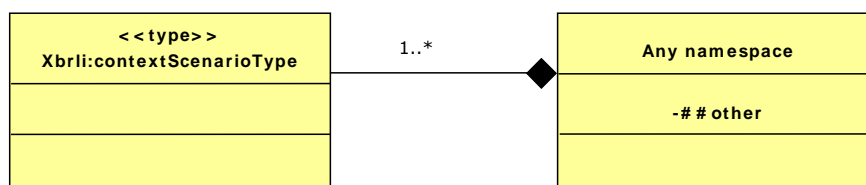
- *link:definitionLink*
- *link:documentation*
- *link:footnote*
- *link:footnoteArc*
- *link:footnoteLink*
- *link:label*
- *link:labelArc*
- *link:labelLink*
- *xbrli:contextScenarioType*

Este tipo (**contextScenarioType**) describe información adicional, equivale a un conjunto de dimensiones en el Modelo de los datos Multidimensional (*Figura 3.5*).

```

<complexType name="contextScenarioType">
  <annotation>
    <documentation>
      Used for the scenario under which fact have been reported.
    </documentation>
  </annotation>
  <sequence>
    <any namespace="##other" processContents="lax"
      minOccurs="1" maxOccurs="unbounded" />
  </sequence>
</complexType>

```



**Figura 3.5** Tipo XBRL *contextScenarioType*

En el *Ejemplo 3.3* se muestra que un escenario está formado por un espacio de nombres, que definirá las dimensiones que forman el escenario y por tanto el contexto.

```

<xbrli:context id="Context_Instant_OtrasEntidades">
  <xbrli:entity>
    <xbrli:identifier scheme="http://www.ecb.int/stats/money/mfi">
      ES9000</xbrli:identifier>
    <xbrli:segment>
      <xbrldi:explicitMember dimension="es-be-cm-dim:Agrupacion">
        es-be-cm-dim:AgrupacionIndividual</xbrldi:explicitMember>
      </xbrli:segment>
    </xbrli:entity>
  <xbrli:period>
    <xbrli:instant>2008-09-30</xbrli:instant>
  </xbrli:period>
  <xbrli:scenario>
    <xbrldi:explicitMember dimension="es-be-d-FR-dist:DistribucionDimension">
      es-be-d-FR-dist:OtrasEntidades</xbrldi:explicitMember>
    </xbrli:scenario>
  </xbrli:context>

```

### Ejemplo 3.3 Ejemplo de escenario

- *link:linkbase*

El *link:linkbase* complementa los roles XBRL sobre la especificación XLink (Figura 3.6), éstos fueron explicados antes (presentación, cálculos, formulas, etc.).

```

<element name="linkbase">
  <annotation>
    <documentation>
      Definition of the linkbase element. Used to contain a set of zero or more
      extended link elements.
    </documentation>
  </annotation>
  <complexType>
    <choice minOccurs="0" maxOccurs="unbounded">
      <element ref="link:documentation"/>
      <element ref="link:roleRef"/>
      <element ref="link:arcroleRef"/>
      <element ref="xl:extended"/>
    </choice>
  </complexType>

```

```

</choice>
<attribute name="id" type="ID" use="optional"/>
<anyAttribute namespace="http://www.w3.org/XML/1998/namespace"
processContents="lax"/>
</complexType>
</element>

```

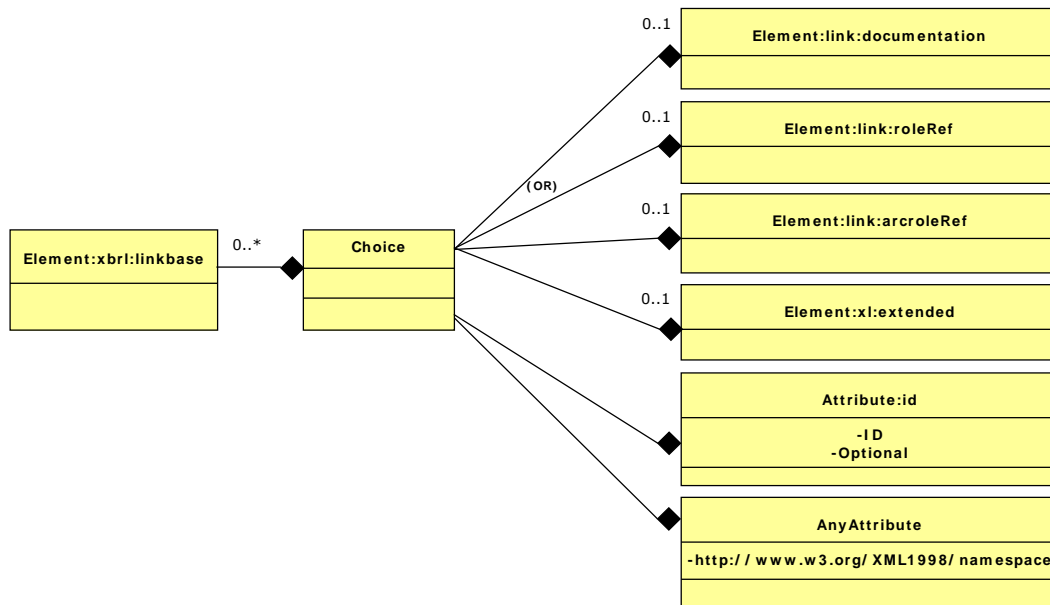


Figura 3.6 Elemento XBRL *linkbase*

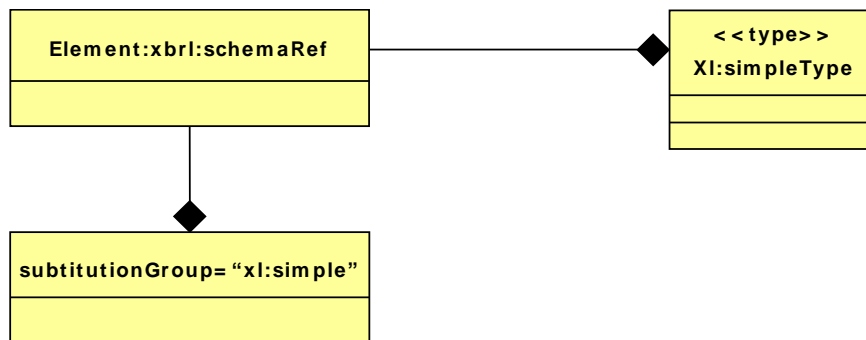
### Objetos XBRL “Elementos”

- *link:linkbaseRef*
- *link:loc*
- *link:part*
- *link:presentationArc*
- *link:presentationLink*
- *link:reference*
- *link:referenceArc*
- *link:referenceLink*
- *link:roleRef*
- *link:roleType*

- *link:schemaRef*

Este elemento se utiliza para que los XBRL documentos instancias referencien a los esquemas taxonómicos, y no reescribir los elementos (*Figura 3.7*).

```
<element name="schemaRef" type="xl:simpleType" substitutionGroup="xl:simple">
  <annotation>
    <documentation>
      Definition of the schemaRef element – used
      to link to XBRL taxonomy schemas from
      XBRL instances.
    </documentation>
  </annotation>
</element>
```



**Figura 3.7** Elemento XBRL *schemaRef*

- *link:usedOn*
- *xbrli:context*

*Context* es el conjunto de dimensiones y atributos de dimensión de un hecho. Está formado por de 0 a 1 segmento (par dimensión, atributo de dimensión) y de 0 a n escenarios, pares <dimensión, atributo de dimensión>, y con la dimensión entidad y periodo de tiempo. En la *Figura 3.8* se muestra su estructura de la definición. El *Ejemplo 3.4* muestra un contexto con su escenario y dos dimensiones, para la entidad *ES9000* en el periodo instantáneo del *2008-09-30*.

```

<element name="context">
  <annotation>
    <documentation>
      Used for an island of context to which facts can be related.
    </documentation>
  </annotation>
  <complexType>
    <sequence>
      <element name="entity" type="xbrli:contextEntityType" />
      <element name="period" type="xbrli:contextPeriodType" />
      <element name="scenario" type="xbrli:contextScenarioType" minOccurs="0" />
    </sequence>
    <attribute name="id" type="ID" use="required" />
  </complexType>
</element>

```

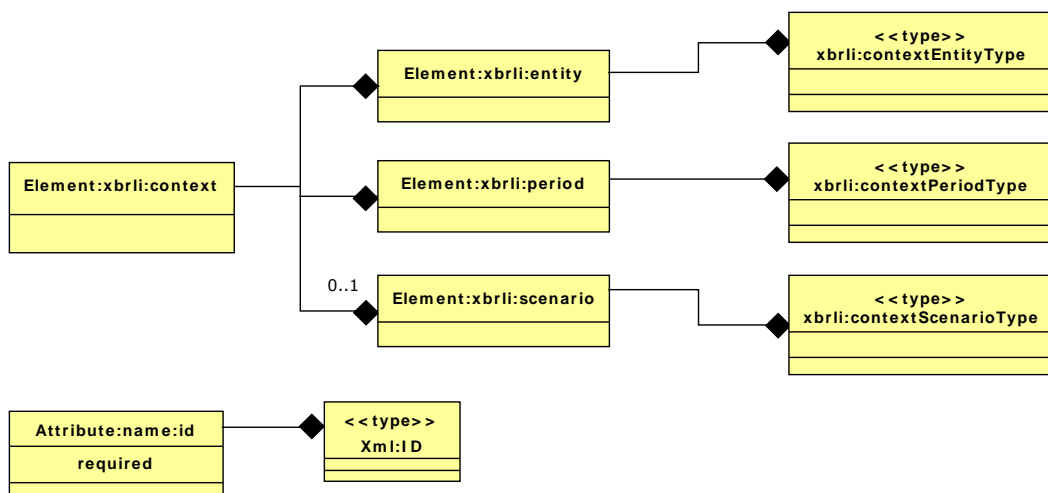


Figura 3.8 Elemento XBRL context

```

<xbrli:context id="Context_Instant_GrupoConsolidableEntidadesCredito">
  <xbrli:entity>
    <xbrli:identifier scheme=http://www.ecb.int/stats/money/mfi>
      ES9000</xbrli:identifier>
    <xbrli:segment>
      <xbrldi:explicitMember dimension="es-be-cm-dim:Agrupacion">
        es-be-cm-dim:AgrupacionIndividual</xbrldi:explicitMember>
      </xbrli:segment>
    </xbrli:entity>
  <xbrli:period>

```

```

    <xbri:instant>2008-09-30</xbri:instant>
  </xbri:period>
  <xbri:scenario>
    <xbri:explicitMember dimension="es-be-d-FR-dist:DistribucionDimension">
      es-be-d-FR-dist:GrupoConsolidableEntidadesCredito</xbri:explicitMember>
    </xbri:scenario>
  </xbri:context>

```

#### Ejemplo 3.4 Ejemplo del elemento XBRL context

- *xbri:denominator*
- *xbri:mumerator*
- *xbri:divide*
- *xbri:item*
- *xbri:measure*
- *xbri:segment*

El elemento *segment* define un segmento de negocio, y en el modelo de datos multidimensional (MDM) define un par <dimension, atributo de dimensión> (Figura 3.9). En el *Ejemplo 3.5* se muestra un segmento, que contiene el par <Agrupación, AgrupaciónIndividual>.

```

<element name="segment">
  <complexType>
    <sequence>
      <any namespace="##other" processContents="lax"
        minOccurs="1" maxOccurs="unbounded" />
    </sequence>
  </complexType>
</element>

```

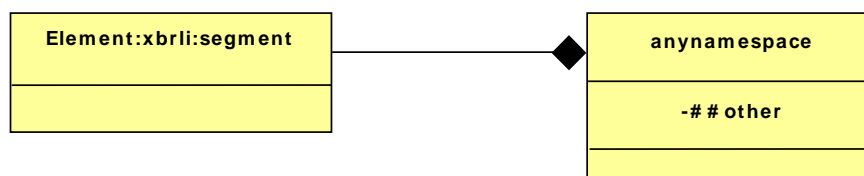


Figura 3.9 Ejemplo del elemento XBRL *segment*

```
<xbrli:segment>
  <xbrldi:explicitMember dimension="es-be-cm-dim:Agrupacion">
    es-be-cm-dim:AgrupacionIndividual</xbrldi:explicitMember>
  </xbrli:segment>
```

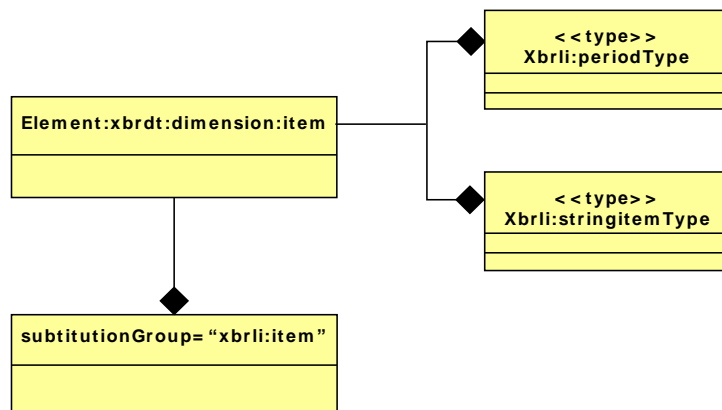
### Ejemplo 3.5 Ejemplo del elemento XBRL *segment*

- *xbrli:tuple*
- *xbrli:unit*
- *ref:Appendix*
- *ref:Article*
- *ref:Chapter*
- *ref:Clause*
- *ref:Example*
- *ref:Exhibit*
- *ref:Footnote*
- *ref:IssueDate*
- *ref:Name*
- *ref:Note*
- *ref:Number*
- *ref:Page*
- *ref:Paragraph*
- *ref:Publisher*
- *ref:Section*
- *ref:Sentence*
- *ref:Subclause*
- *ref:subparagraph*
- *ref:Subsection*
- *ref:URI*
- *ref:URIDate*

- *Xbrldt:dimensionItem*

El elemento “dimensionItem” que aparece en la siguiente figura (*Figura 3.10*) recoge conceptos multidimensionales. En próximos capítulos se profundizará en ellos.

```
<xs:element
  name="dimensionItem"
  id="xbrldt_dimensionItem"
  abstract="true"
  substitutionGroup="xbrli:item"
  type="xbrli:stringItemType"
  xbrli:periodType="duration"/>
```



**Figura 3.10** Elemento XBRL *dimension Item*

- *xbrldt:hypercubeItem*

Define un cubo con uno o varios hechos (*Figura 3.11*). Se hablará en profundidad de este elemento en próximos capítulos.

```
<xs:element
  name="dimensionItem"
  id="xbrldt_dimensionItem"
  abstract="true"
  substitutionGroup="xbrli:item"
  type="xbrli:stringItemType"
  xbrli:periodType="duration"/>
```

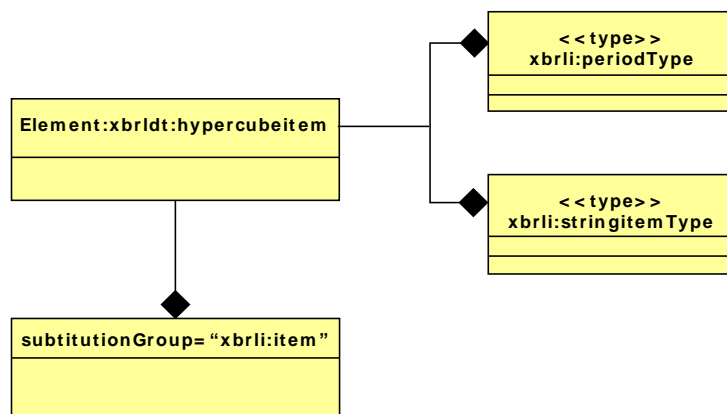


Figura 3.11 Elemento XBRL *hypercubeItem*

- *xbrli:xbrl*

Es el elemento raíz de todos los documentos instancias (*Figura 3.12*).

```

<element name="xbrl">
  <annotation>
    <documentation>
      XBRL instance root element.
    </documentation>
  </annotation>
  <complexType>
    <sequence>
      <element ref="link:schemaRef" minOccurs="1" maxOccurs="unbounded" />
      <element ref="link:linkbaseRef" minOccurs="0" maxOccurs="unbounded" />
      <element ref="link:roleRef" minOccurs="0" maxOccurs="unbounded" />
      <element ref="link:arcroleRef" minOccurs="0" maxOccurs="unbounded" />
      <choice minOccurs="0" maxOccurs="unbounded">
        <element ref="xbrli:item"/>
        <element ref="xbrli:tuple"/>
        <element ref="xbrli:context"/>
        <element ref="xbrli:unit"/>
        <element ref="link:footnoteLink"/>
      </choice>
    </sequence>
    <attribute name="id" type="ID" use="optional" />
    <anyAttribute namespace="http://www.w3.org/XML/1998/namespace"
      processContents="lax" />
  </complexType>
</element>

```

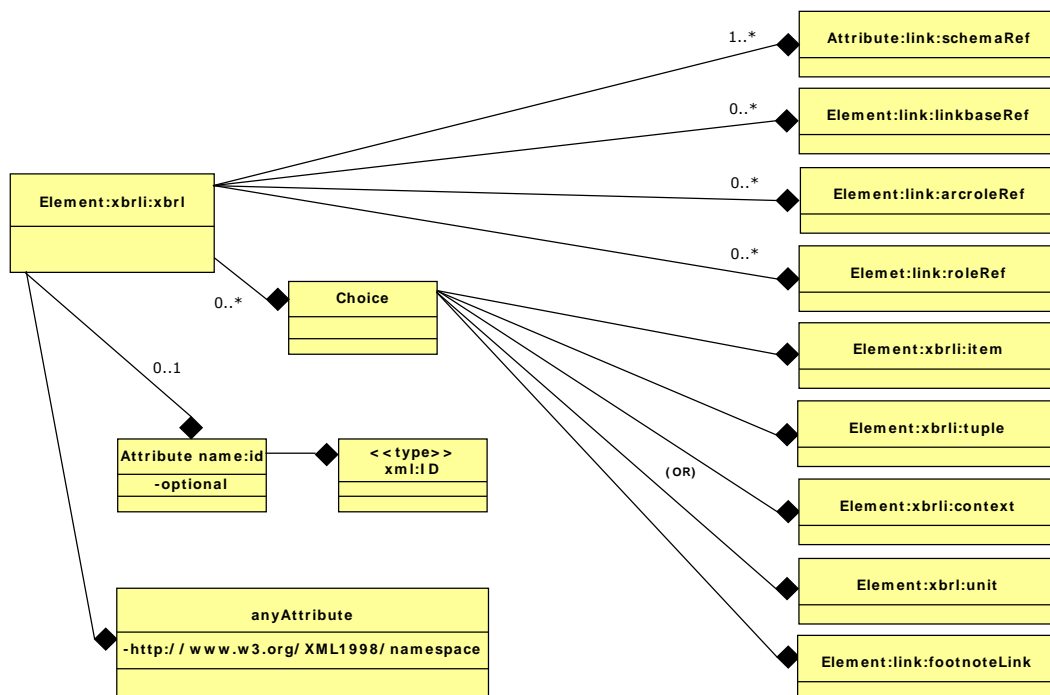


Figura 3.12 Elemento XBRL *xbrl*

En el *Ejemplo 3.6* se muestra un caso de un XBRL documento instancia. En este ejemplo se muestra la definición del segmento como la dimensión “es-be-cm-dim:Agrupacion” y tenemos que el atributo de dimensión es “es-be-ce-dim:AgrupacionIndividual”. Además, que el periodo definido tiene su fecha de comienzo en 2008-07-01 y su fecha de final en 2008-09-30.

```
<xbri:xbri>
  <link:schemaRef xlink:type="simple" xlink:href=
    "http://www.bde.es/es/fr/esrs/finrep/6-2008/2008-11-26/es-be-finrep-consolidado.xsd"/>
  <xbri:context id="Context_Instant"></xbri:context>
  <xbri:context id="IS1"></xbri:context>
  <xbri:context id="Context_Instant_GrupoConsolidableEntidadesCredito">
  </xbri:context>
  <xbri:context id="Context_Instant_EntidadesSeguros"></xbri:context>
  <xbri:context id="pblo_FINREP"></xbri:context>
  <xbri:unit id="EURO"></xbri:unit>
  <es-be-p-FINREP:PromemoriaRiesgosContingentes decimals="-3"
    contextRef="Context_Instant" unitRef="EURO">
    0</es-be-p-FINREP:PromemoriaRiesgosContingentes>
  <ifrs-gp:CashAndBalancesWithCentralBanks decimals="-3" contextRef=
    "Context_Instant" unitRef="EURO">6316000
  </ifrs-gp:CashAndBalancesWithCentralBanks>
  <ifrs-gp:FinancialAssetsHeldForTradingTotal decimals="-3" contextRef=
    "Context_Instant" unitRef="EURO">25680000
  </ifrs-gp:FinancialAssetsHeldForTradingTotal>
  <es-be-p-FINREP:CarteraNegociacionDepositoEntidadesCredito decimals="-3"
    contextRef="Context_Instant" unitRef="EURO">4366000
  </es-be-p-FINREP:CarteraNegociacionDepositoEntidadesCredito>
</xbri:xbri>
```

### **Ejemplo 3.6 Ejemplo de un XBRL documento instancia**

En el *Ejemplo 3.7* se detalla un ítem del *Ejemplo 3.6*.

```
<es-be-p-FINREP:CarteraNegociacionDepositoEntidadesCredito decimals="-3"
contextRef="Context_Instant" unitRef="EURO">4366000</es-be-p-
FINREP:CarteraNegociacionDepositoEntidadesCredito>
```

### **Ejemplo 3.7 Detalle de un ítem en un XBRL documento instancia**

En el *Ejemplo 3.8* se muestra un ejemplo de XBRL documento instancia haciendo uso de tuplas.

```
<pgc07mc-apdo0:PresentacionCuentasTupla>
  <pgc07mc-apdo0:YearFechaInicioCuentas contextRef="D.ACTUAL">2009</pgc07mc-
apdo0:YearFechaInicioCuentas>
  <pgc07mc-apdo0:MonthFechaInicioCuentas contextRef="D.ACTUAL">01</pgc07mc-
apdo0:MonthFechaInicioCuentas>
  <pgc07mc-apdo0:DayFechaInicioCuentas contextRef="D.ACTUAL">01</pgc07mc-
apdo0:DayFechaInicioCuentas>
  <pgc07mc-apdo0:YearFechaCierreCuentas contextRef="D.ACTUAL">2009</pgc07mc-
apdo0:YearFechaCierreCuentas>
  <pgc07mc-apdo0:MonthFechaCierreCuentas contextRef="D.ACTUAL">12</pgc07mc-
apdo0:MonthFechaCierreCuentas>
  <pgc07mc-apdo0:DayFechaCierreCuentas contextRef="D.ACTUAL">31</pgc07mc-
apdo0:DayFechaCierreCuentas>
  <dgi-dat-inf:TotalPagesPresented decimals="0" contextRef="D.ACTUAL"
unitRef="pure">43</dgi-dat-inf:TotalPagesPresented>
</pgc07mc-apdo0:PresentacionCuentasTupla>
<pgc07mc-apdo0:PresentacionCuentasTupla>
  <pgc07mc-apdo0:YearFechaInicioCuentas
contextRef="D.ANTERIOR">2008</pgc07mc-apdo0:YearFechaInicioCuentas>
  <pgc07mc-apdo0:MonthFechaInicioCuentas contextRef="D.ANTERIOR">01</pgc07mc-
apdo0:MonthFechaInicioCuentas>
  <pgc07mc-apdo0:DayFechaInicioCuentas contextRef="D.ANTERIOR">01</pgc07mc-
apdo0:DayFechaInicioCuentas>
  <pgc07mc-apdo0:YearFechaCierreCuentas
contextRef="D.ANTERIOR">2008</pgc07mc-apdo0:YearFechaCierreCuentas>
  <pgc07mc-apdo0:MonthFechaCierreCuentas
contextRef="D.ANTERIOR">12</pgc07mc-apdo0:MonthFechaCierreCuentas>
  <pgc07mc-apdo0:DayFechaCierreCuentas contextRef="D.ANTERIOR">31</pgc07mc-
apdo0:DayFechaCierreCuentas>
  <dgi-dat-inf:TotalPagesPresented decimals="0" contextRef="D.ANTERIOR"
unitRef="pure">0</dgi-dat-inf:TotalPagesPresented>
</pgc07mc-apdo0:PresentacionCuentasTupla>
```

**Ejemplo 3.8** Ejemplo de uso de tuplas en un XBRL documento instancia

- *xbrldi:explicitMember*

Atributos de dimensión de una dimensión (Figura 3.13), los atributos de dimensión se definen explícitamente en la taxonomía (también pueden ser atributos calculados en el MDM).

```

<element name="explicitMember">
  <annotation>
    <documentation xml:lang="en">This element contains the QName of an item that
is a member of an explicit dimension.
    </documentation>
  </annotation>
  <complexType>
    <simpleContent>
      <extension base="QName">
        <attribute name="dimension" type="QName" use="required"/>
      </extension>
    </simpleContent>
  </complexType>
</element>

```

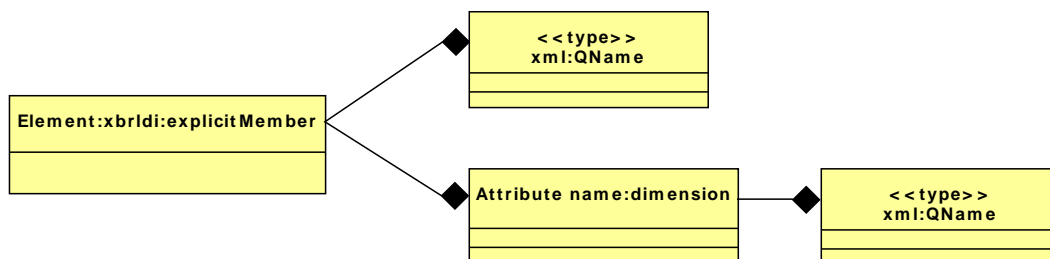


Figura 3.13 Elemento XBRL *explicitMember*

En el Ejemplo 3.9 se muestra la dimensión *Agrupación* y el atributo de dimensión *AgrupacionIndividual* (*ExplicitMember*).

```

<xbrli:context id="Context_Instant_GrupoConsolidableEntidadesCredito">
  <xbrli:entity>
    <xbrli:identifier scheme="http://www.ecb.int/stats/money/mfi">ES9000</xbrli:identifier>
    <xbrli:segment>
      <xbrldi:explicitMember dimension="es-be-cm-dim:Agrupacion">es-be-cm-
dim:AgrupacionIndividual</xbrldi:explicitMember>
    </xbrli:segment>
  </xbrli:entity>

```

```

<xbrli:period>
  <xbrli:instant>2008-09-30</xbrli:instant>
</xbrli:period>
<xbrli:scenario>
  <xbrldi:explicitMember dimension="es-be-d-FR-dist:DistribucionDimension">es-
be-d-FR-dist:GrupoConsolidableEntidadesCredito</xbrldi:explicitMember>
</xbrli:scenario>
</xbrli:context>

```

### Ejemplo 3.9 Ejemplo del elemento XBRL explicitMember

- *typedMember*

Y por último, como se verá en los próximos capítulos una dimensión tipada es una dimensión que no se conocen sus atributos de dimensión en el momento de definir la taxonomía (en tiempo de ejecución), pero sí su tipo. Se llama también dimensión implícita (*Figura 3.14*).

```

<element name="typedMember">
  <annotation>
<documentation xml:lang="en">This element constains one child of anyType.
  </documentation>
</annotation>
<complexType>
  <sequence>
    <any namespace="##other"/>
  </sequence>
  <attribute name="dimension" type="QName" use="required"/>
</complexType>
</element>

```

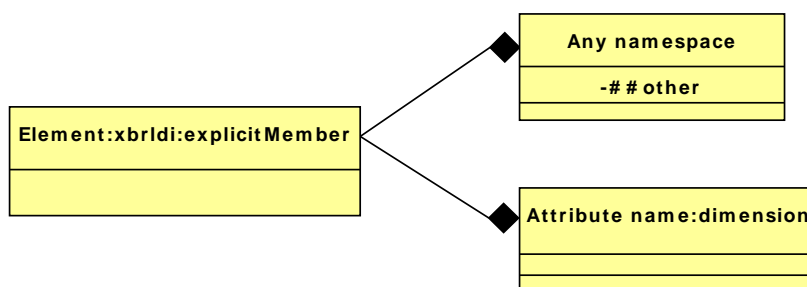


Figura 3.14 Elemento XBRL *typedMember*



# CAPÍTULO 4

---

## Fundamentación y Conceptos Básicos de la Propuesta



## CAPÍTULO 4. Fundamentación y Conceptos Básicos de la Propuesta

---

### 4.1. El Universo del discurso y propuesta

Este capítulo analiza el ciclo de vida del desarrollo del software de los metadatos de los informes semánticos contables, utilizando el paradigma del *Model Driven Architecture* (MDA, 2015). Desde 2008, especialmente en Europa, los Bancos Centrales Nacionales (*National Central Banks, NCBs*), el Banco Central Europeo (*European Central Bank, ECB*), la EBA (*European Banking Authority, EBA*) y en general, reguladores y supervisores europeos, han buscado un modelo de datos para el desarrollo de este tipo de informes (*Eurofiling, 2015*). En el año 2010 se creó una versión estable del *Data Point Model* (DPM), que es un modelo lógico, sólo aplicable a la especificación XBRL (*Eurofiling 2011*). El DPM se utilizó primeramente para las taxonomías del *Balance Sheet Items and Monetary Financial Institutions Interest Rates* (*BSI-MIR, 2010; Banco de España 2010*) y ésta fue implementada por la compañía polaca BR-AG (2015). Dos años después apareció un generador de plantillas Excel desde la taxonomía XBRL (*Data Point Modeller* (*Ruíz et al. 2012*)). El *Comité Européen de Normalisation/European Committee for Standardization* (CEN, 2013) y *Eurofiling* (2015) han estado trabajando en la estandarización del diseño del modelo de datos a través del *Modelo de Punto de Datos* (Data Point Model, DPM) (*Díaz, 2012*). Una herramienta para modelar el DPM es el *DPM Architect for XBRL*, y está bajo desarrollo del Banco de España. Esta herramienta dará a algunos supervisores la posibilidad de desarrollar taxonomías XBRL. El Banco d'Italia desarrolló el predecesor del DPM, la *Matrix Schema* (MS) (*Romanelli, 2007*), el cual contribuyó a la iniciativa de *Eurofiling* (2015). El MS está basado en una hoja de cálculo Excel. Como se ha dicho en otras secciones esta tesis no trata del DPM, por no hacer el trabajo más extenso, aunque el autor de esta tesis ha trabajado en ella (*Santos, 2013; Santos et al., 2013*).

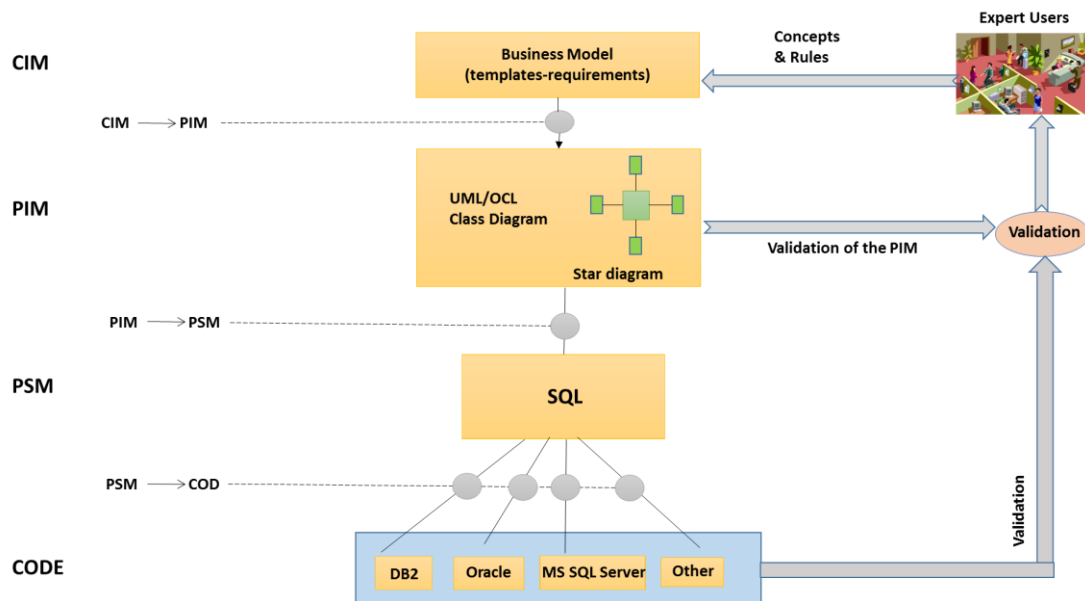


Figura 4.1 Diseño de informes semánticos utilizando el paradigma MDA

La Figura 4.1 muestra parte de la propuesta de esta tesis, el ciclo de vida del desarrollo del software, y será desarrollada en este capítulo. Los economistas, contables y financieros necesitan obtener un conjunto de datos. Estos usuarios expertos, con la ayuda de Sistemas de Información (*Information Systems, IS*), construyen un conjunto de plantillas a través de una o más hojas de cálculo. Por tanto, el mundo real consiste en un conjunto de reglas, leyes, directivas, etc. definidas en un conjunto de datos requeridos en un informe, report (el *Computation Independent Model, CIM*), a través de plantillas. De acuerdo al paradigma MDA, el *Platform Independent Model (PIM)* es obtenido desde el CIM. En el PIM se analizan el conjunto de definiciones, reglas de usuario, y se realiza un mapeo o transformación desde el CIM al PIM. El modelo utilizado en esta tesis es el Modelo de Datos Multidimensional (MDM). El MDM es un modelo sencillo que combina objetos, dimensiones (jerarquías), las medidas y los atributos para representar problemas de negocio reales de trabajo (Kimball, 1996-2004; Inmon, 2005; Jarke et al., 2003). Además, este modelo es el corazón del procesamiento analítico en línea (OLAP), que exige consultas complejas que pueden ser resueltas por el MDM en tiempo real.

La plataforma específica elegida es SQL (*Platform Specific Model, PSM*). El PSM consiste en un conjunto de definiciones, reglas y transformaciones. Al ser utilizado el MDM en el PIM, y SQL en el PSM, la transformación es casi inmediata, utilizando la tecnología ROLAP (*Relational Online Analytical Processing*). Analíticamente se

muestra la transformación del PSM al código. Además se provee de un conjunto automático de validaciones en el PIM y en el PSM.

El diseño de estos informes es extremadamente general y los diseños de las taxonomías (conjunto de conceptos y reglas para los informes semánticos), pueden variar enormemente. En Europa el diseño hace un uso intensivo de las dimensiones (Boixo y Flores, 2005; Felden 2007). La utilización de dimensiones hace que el proceso de diseño sea más fácil, ya que si el número de dimensiones en el modelo conceptual es alto, es semánticamente más rico y la transformación a la base de datos es más fácil. Adicionalmente, es posible reducir el número de dimensiones a través de vistas en el modelo lógico (el PSM), según el criterio de implementación. En la actualidad, estos informes utilizan la especificación XBRL a través de taxonomías. Es intención de esta tesis, utilizando el paradigma MDA, ayudar al diseño de estas taxonomías o informes semánticos en XBRL, o en otras taxonomías o informes (XML, SDMX, iXBRL, hojas de cálculo, etc.).

Este capítulo se divide en cinco secciones, de acuerdo a explicar el ciclo de vida del desarrollo de estos informes, usando el MDA. En el punto 4.2. se analiza el CIM, (Modelo independiente computacional). En el 4.3., el PIM (Modelo Independiente de la plataforma), sus reglas y definiciones. El punto 4.4 estudia las cuestiones semánticas acerca del diseño de los metadatos en el PIM. Y, por último, en el punto 4.5. se muestra el PSM (Modelo específico de la plataforma); también se mostrará el código de esta propuesta. La validación de la propuesta se analizará en el *Capítulo 6* de esta tesis.

## **4.2. Modelo Independiente de Computación (CIM)**

Esta sección empezará analizando el mundo real. Un economista, un contable, sólo quiere obtener un conjunto de datos. En ciertos casos, estos especialistas diseñan un informe como el de la *Figura 2.2*, sin embargo, en la mayoría de los casos, éstos desean recolectar estos datos, pero independientemente de su presentación. Un método generalizado es generar una o más hojas de cálculo o plantillas con los datos que son necesarios. De esta forma, la presentación de los datos está separada desde su definición. De acuerdo a la lógica de negocio, los usuarios expertos crearán una o más hojas de cálculo, cada una con un grupo de celdas. La *Figura 4.2* muestra un

ejemplo simplificado con un informe similar al de la *Figura 2.2*. En la *Figura 4.2* hay un conjunto de conceptos interconectados y un conjunto de valores asignados a esos conceptos. Estos valores deben cumplir ciertas reglas con otros valores y otros conceptos de este informe o conjunto de informes. La figura muestra los activos financieros en un periodo de tiempo de un país con reglas específicas, tales como los préstamos inmobiliarios del banco o institución financiera, que deben ser igual a la suma de los préstamos inmobiliarios con el propio banco y con otros bancos.

Germany Euros

Financial assets, period: 1<sup>st</sup> quarter, 2012

Financial institution	Type of Loan	Type of Asset	Value
BNP Paribas	The bank itself	Real Estate	10,000.00
BNP Paribas	To other banks	Real Estate	20,000.00
ING Group	The bank itself	Real Estate	10,000.00
ING Group	To other bank	Real Estate	15,000.00
BNP Paribas	To bank itself and other banks	Real Estate	30,000.00

**Figura 4.2** Ejemplo de informe con semántica

En primer lugar se propone la definición de un informe semántico económico/financiero.

**Definición 4.1:** *Un informe económico/financiero es semántico si está compuesto de un conjunto de conceptos interconectados entre sí, y de valores asignados a estos conceptos o grupos de ellos. Además, los valores deben cumplir ciertas reglas y/o restricciones entre otros valores y conceptos.*

Sin embargo, en la mayoría de los casos, los usuarios expertos quieren datos independientemente de la presentación. Un método habitual es generar una o más hojas de cálculo, o plantillas, con los datos que son necesarios recolectar. De esta manera, la presentación de los datos está separada de su definición. De acuerdo a la lógica de negocio, cada hoja tiene un grupo de celdas con datos que los economistas o contables quieren obtener en los informes. La *Figura 4.3* muestra un ejemplo simplificado con solo tres celdas, basadas en el informe de la *Figura 4.2*: F(5, 1, 1), F(5, 1, 2), y F(5, 1, 3). Estas celdas se corresponderían a la columna valor de

la Figura 4.2, F(5.1.1) a la fila 1, F(5,1,2) a la fila 2, y la F(5,1,3) podría ser la fila 5, si el concepto *Banks* fuera lo mismo que *To bank itself and other Banks*.

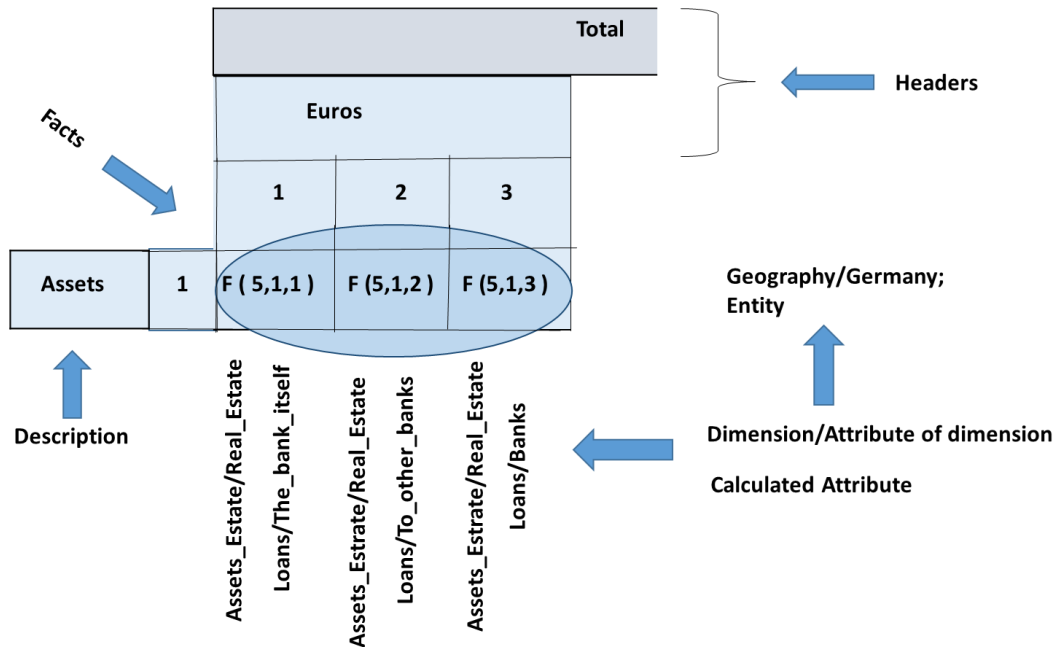


Figura 4.3 Ejemplo de plantilla

Desde estas plantillas el analista de Tecnologías de la Información (IT), junto con el usuario de negocios, extraen los metadatos. En esta plantilla, los usuarios expertos muestran los datos que ellos necesitan recoger. El analista de IT se encuentra con un conjunto de hojas Excel, con un gran número de celdas inconexas entre ellas y con un alto grado de redundancias. La Regulación Europea en 2012 necesitaba 4.500 celdas, en 2015 el número de celdas es de 45.000 (Weller, 2015). Cada plantilla tiene un significado diferente para el usuario de negocio. La plantilla consistirá en un conjunto de celdas, en el que cada celda es un hecho a ser recolectado, y éste está determinado por un conjunto de dimensiones y atributos de dimensión, entre otras cosas. Por ejemplo, F ( 5, 1, 1 ) es un activo inmobiliario, con un préstamo del banco, para una entidad, en euros. Si en esta figura las celdas de los hechos están tachadas, se consideran como no permitidas para el usuario de negocios. Los hechos se listan a través de una tripleta, en la cual el primer número es el número de la plantilla, y posteriormente la fila y la columna. Normalmente los hechos se referencian con números secuenciales, así que se puedan definir:

**Definición 4.2:** Sea  $\varphi$  un hecho. Sea  $i$  un número, donde  $i \in N$ . Entonces  $\varphi \leftrightarrow i$ .

Por otra parte, un hecho puede determinar más de una tripleta (plantilla, fila, columna), ya que un hecho puede estar en más de una plantilla. Por tanto, es posible definir la siguiente regla:

**Regla 4.1:** Sea  $\varphi$  un hecho. Sea  $T_i$  una plantilla,  $\forall i=1,\dots,n$ . Sea  $R_j$  una fila, donde  $R_j \in T_i, \forall j=1,\dots,m$ . Sea  $C_k$  una columna, donde  $C_k \in T_i, \forall k=1,\dots,o$ . Entonces  $\varphi \rightarrow (T_i, R_j, C_k)$ .

Las pruebas de concepto (POC) de esta tesis están basadas en los informes que deben ser enviados desde las instituciones financieras a los supervisores europeos (*Openfiling*, 2015). Estas pruebas de concepto utilizan el borrador de las taxonomías FINREP 2012 (*EBA 2011; Eurofiling. 2012, 2015*), publicado en internet, con uso intensivo de dimensiones.

### 4.3. Modelo Independiente de la Plataforma (PIM): reglas y definiciones

Esta sección analiza el PIM de este modelo. Se utiliza UML para mostrar todas las definiciones y reglas necesarias de esta plataforma. En esta tesis, se propone el uso del diseño en estrella del modelo de datos multidimensional (MDM), con 16 definiciones o reglas como conjunto mínimo para construir un informe semántico económico / financiero. La *Tabla 4.1* resume el conjunto de definiciones. La columna 1 define el nombre del concepto en el MDM y la columna 2 su descripción. Sin embargo, estas definiciones están basadas en el modelo de datos XBRL (XBRLDM). Adicionalmente, esta sección presenta una nueva dimensión a la propuesta, ofreciendo una formalización matemática (lenguaje formal en el MDM).

**Tabla 4.1 Definiciones y reglas en el MDM (El PIM)**

Nombre en el MDM	Descripción
Concepto	Es la definición de un concepto de negocio o ítem. Cada concepto está asociado a un atributo de período de tiempo ( <i>Instant</i> , <i>Period</i> , y <i>Forever</i> ).
Concepto básico	Es un concepto especial que tiene asociado un tipo de dato, un tipo de período de tiempo y un tipo de balance.
Dominio	Es un grupo de conceptos que pertenecen a un campo o espacio de conocimiento, o actividad. En este modelo un dominio puede contener conceptos básicos y no básicos, pero no ambos.
Dimensión base	Es un conjunto con solo conceptos básicos.
Dimensión	Es un conjunto de conceptos de un dominio. Estos conceptos tienen una estructura en árbol.
Dimensión explícita o implícita	Ésta es <b>explícita</b> , si los atributos de dimensión están definidos. Es <b>implícita</b> si no están definidos. Dimensión → Dominio. Dominio → → Dimensión.
Grupo de dimensiones	Grupo de dimensiones de un dominio.
Atributo Calculado	Es un agregado de atributos de dimensión, y/o calculados.
Atributo de dimensión	No son agregados.
Atributo por defecto	Cada dominio tiene un concepto por defecto.
Restricción jerárquica	Los conceptos en una dimensión tienen una estructura en árbol. Se utiliza en las validaciones entre una hoja raíz y sus hojas, es decir para los atributos calculados.
Referencias	Referencias a directivas o leyes de los conceptos.
Dimensión/atributos de dimensión y calculados	Pueden definir un hecho o un conjunto de hechos.
Hecho::= - <dimensión / atributo de dimensión - Concepto básico - Atributo calculado	Un hecho es la representación de un valor de una particular medida para la entidad que reporta el informe.
Hecho permitido	Restricción de usuario.
Hecho prohibido	Restricción de usuario.

La primera definición, de acuerdo a la *Tabla 4.1*, es la definición de concepto o *ítem* de negocio. Cada concepto tiene un atributo de tipo de período de tiempo. El tipo de período de tiempo puede tomar tres valores: *Instant*, *Period* y *Forever*. En la *Figura 4.2*, los conceptos son {*Entity\_Finan*, *BNP Paribas*, *ING Group*, *Royal Bank of Scotland*, *Commerzbank*, *Real estate*, *No real estate*, *Real estate and no real estate*, *Assets*, *Liabilities*, ...}.

Un concepto básico *primary ítem*, en el XBRLDM, es un concepto especial, pues tiene tres atributos asociados: tipo de dato, tipo de período de tiempo del concepto básico y balance (*Hernández-Ros y Wallis, 2006; Santos y Castro. 2011b, c*). De acuerdo al XBRLDM, si el tipo de dato es un concepto básico, y es *monetary*, existe un nuevo Atributo, *balance*, el cual puede tomar dos valores: *debit* o *credit*. Si el tipo de dato no es *monetary*, entonces el valor del atributo *balance* tendrá un valor nulo. En el ejemplo los conceptos básicos son  $\{Assets (Instant, monetary, <balance>), Liabilities (Instant, monetary, <balance>), \dots\}$ .

Todos los conceptos de un dominio deben tener un mismo período de tiempo. Un dominio está formado de un conjunto de conceptos, y cada concepto pertenece a un dominio simple. En este ejemplo el conjunto de dominios son *DEntity*, *DAssets\_Estate*, *DLoans*, y *DGeography*. Donde el dominio *DEntity* consta de los siguientes conceptos  $\{Entity\_Finan, BNP Paribas, ING Group, Royal Bank of Scotland, Commerzbank\}$  y el dominio *DAssets\_Estate* =  $\{Real estate, No real estate, Real estate and no real estate\}$ , y *DLoans* =  $\{Banks, The bank itself, To other banks, To non-financial institutions, The bank itself and other banks\}$ , etc.

Estas definiciones se pueden expresar formalmente como:

**Definición 4.3:** Sea  $\xi_i$  un concepto, donde  $\xi_i \in \Gamma$ , en el que  $\Gamma$  es el conjunto de conceptos,  $\forall i=1, \dots, n$ . Sea  $Do_j$  un dominio, tal que  $Do_j \in Do$ , en el que  $Do$  es el conjunto de dominios,  $\forall j=1, \dots, m$ . Entonces  $Do_j = \bigcup_{i=1}^k \xi_i$ .

**Definición 4.4:** De acuerdo a la definición anterior,  $\forall \xi_i \Rightarrow \xi_i \rightarrow Do_j$ .

**Definición 4.5:** Sea  $\xi_i$  un concepto, donde  $\xi_i \in \Gamma$ , donde  $\Gamma$  es el conjunto de conceptos,  $\forall i=1, \dots, m$ . Sea  $\varrho$  un tipo de período de tiempo, donde  $\varrho = \{“Instant” | “period” | “forever”\}$ . Sea  $Do_j$  un dominio,  $\forall j=1, \dots, n$ , donde  $Do_j \in Do$ , en el que  $Do$  es un conjunto de dominios. Entonces  $\xi_i \rightarrow \varrho$ . Y si  $\xi_i \rightarrow Do_j \Rightarrow Do_j \rightarrow \varrho$ .

Una dimensión es un conjunto de conceptos de un dominio. En el MDM o en la especificación XBRLDM, no puede haber más de un atributo de dimensión de una dimensión que se refiera a un hecho. Sin embargo, en el mundo real puede haber más de un concepto de un dominio que haga referencia a un hecho. La solución en el XBRLDM es crear muchas dimensiones en el mismo dominio, de tal manera que

cada hecho tiene una dimensión (*member-domain* en el XBRLDM), sin solapar los atributos de dimensión en un hecho. En el MDM se crean dimensiones de un dominio con atributos de dimensión superpuestos. Esto significa, que una dimensión determina únicamente un dominio. Entonces, en el ejemplo es posible definir en el dominio *DLoans*, la dimensión  $Loans\_1 = \{ 'The bank itself' + 'To other banks' \}$  y en el dominio *DAssets\_Estate*, la dimensión  $Assets\_Estate\_1 = \{ 'Real estate', 'No real estate' \}$ .

Por ejemplo, en el dominio *DLoans* el concepto *'The bank itself and other banks'* es un atributo calculado de la dimensión *Loans\_1*, donde *'The bank itself and other banks'* = *'The bank itself' + 'To other banks'*. También, el concepto *'Real estate and no real estate'* del dominio *DAssets\_Estate* es un atributo calculado de la dimensión *Assets\_Estate\_1*, donde *'Real estate and no real estate'* = *'Real estate' + 'No real estate'*.

Un atributo de dimensión determina un concepto simple de un dominio, pero los atributos de dimensión determinan de 1 a n dimensiones. Por tanto es posible definir:

**Definición 4.6:** Sea  $D_i$  una dimensión,  $\forall i=1, \dots, m$ , donde  $D_i \in D$ , y  $D$  es el conjunto de dimensiones. Sea  $At_{i_j}$  un atributo de dimensión, tal que  $D_i = \cup_{j=1}^k At_{i_j}$ . Sea  $Do_l$  un dominio, y sea  $\xi_{l_o}$  un concepto, donde  $Do_l = \cup_{o=1}^{k'} \xi_{l_o}$ . Si  $At_{i_j} \rightarrow \xi_{l_o} \Rightarrow \xi_{l_o} \rightarrow Do_l$  y  $\xi_{l_o} \rightarrow \sum_{i=0}^n D_i \in Do_l$ .  $\forall l = 1, \dots, n$  y  $\forall o = 1, \dots, n'$ .

En el XBRLDM una dimensión consiste de un *domain-member*, y no hay diferencias entre atributos calculados y dimensión. Un atributo calculado se define como:

**Definición 4.7:** Sea  $\xi_{i'j'}$  un concepto, y sea  $Do_{i'}$  un dominio,  $\forall i' = 1, \dots, m, \forall j' = 1, \dots, m'$ . Donde  $\xi_{i'j'} \in Do_{i'}$ . Sea  $At_{i_j}$  un atributo de dimensión, donde  $At_{i_j} \in D_i$ , donde  $D_i$  es una dimensión  $\forall i = 1, \dots, n$ , y  $\forall j = 1, \dots, n'$ . Si  $\forall \xi_{i_j} \in Do_{i'}$ ,  $\exists$  a  $\xi_{i'j'}$ , tal que  $At_{i_j} \rightarrow \xi_{i'j'}$ , entonces  $(\xi_{i'j'}, D_{i'}) \rightarrow CAt_{i'j'}$ ,  $CAt_{i'j'}$  es un atributo calculado.

El conjunto de conceptos básicos se agrupan en la dimensión base. La dimensión base determina un dominio y, cada uno de los conceptos básicos determina únicamente los conceptos del dominio. En general, la especificación XBRLDM no especifica si hay una o varias dimensiones base, pero según la guía de buenas

prácticas de *Eurofiling* sólo debe existir una dimensión base por marco (*framework*) o taxonomía (*CEN, 2013*). En el ejemplo la dimensión base es *Base\_Dimension={Assets (Instant, monetary, <balance>), Liabilities (Instant, monetary, <balance>), ...}*.

Por tanto es posible definir:

**Definición 4.8:** Sea  $\xi_j$  un concepto,  $\forall j = 1, \dots, n$ , y  $\xi_j \in \Gamma$ , donde  $\Gamma$  es el conjunto de conceptos. Sea  $\varrho$  el periodo de tiempo, donde  $\varrho = \{\text{"Instant"}, \text{"period"}, \text{"forever"}\}$ . Sea  $o$  un tipo de dato. Entonces  $\xi_j \cup \varrho \cup o \rightarrow Bc$ , donde  $Bc$  es un concepto básico, y  $\xi_i \rightarrow \varrho$ ,  $\xi_i \rightarrow o$ .

**Definición 4.9:** Sea  $Bc_i$  un concepto básico,  $\forall i = 1, \dots, m$ . Sea  $Bd$  una dimensión base, entonces  $Bd = \bigcup_{i=1}^m Bc_i$ .

XBRLDM define una familia de dimensiones como un conjunto de dimensiones, las cuales se agrupan por su semántica. En el MDM se llamarán grupos de dimensiones. Un grupo de dimensiones pertenece a un dominio, y cada dimensión del grupo determina un dominio y este dominio es el mismo que el del grupo. Entonces, un atributo de dimensión puede pertenecer a algunas dimensiones (realmente el concepto) del mismo grupo de dimensión.

**Definición 4.10:** Sea  $D_i$  una dimensión, donde  $D_i \in D$ .  $D$  es el conjunto de dimensiones,  $\forall i = 1, \dots, n$ . Sea  $Do_k$  un dominio donde  $D_i \rightarrow Do_k$ ,  $\forall k = 1, \dots, m$ . Sea  $DG_j$  un grupo de dimensiones, donde  $DG_j \in DG$ ,  $DG$  es el conjunto de grupos de dimensiones,  $\forall j = 1, \dots, m$ . Entonces es  $DG_j = \bigcup_{i=1}^n D_i$ .

Pero también es posible deducir:

**Regla 4.2:** Si  $D_i \rightarrow Do_k$ ,  $\forall i = 1, \dots, m$ ,  $\forall k = 1, \dots, n$ , y si  $D_i \in DG_l$ ,  $\forall l = 1, \dots, o \Rightarrow DG_l \rightarrow Do_k$ ,  $\forall l = 1, \dots, o$ .

Por otra parte, en el XBRLDM, todos los dominios definidos tienen que tener un concepto por defecto, con contenido semántico (*Hernández-Ros y Wallis, 2006; Eurofiling, 2011*). También en este modelo de datos cada dimensión debe tener un concepto por defecto del dominio al cual la dimensión pertenece. De este modo, en

el MDM se define un concepto por defecto en el dominio. Pero el concepto por defecto puede ser un atributo de dimensión o calculado, o ambos. Aunque normalmente será calculado como se ve a continuación.

**Definición 4.11:** Sea  $\Gamma$  el conjunto de conceptos definidos en una taxonomía. Sea  $\xi_i$  un concepto, y  $\xi_i \in \Gamma^{def}$ ,  $\forall i = 1, \dots, n$ , donde  $\Gamma^{def}$  es el conjunto de conceptos definidos por defecto. Sea  $Do_j$  un dominio, donde  $Do_j \in Do$ , y  $Do$  es el conjunto de dominios,  $\forall j = 1, \dots, m$ , entonces  $Do_j \rightarrow \xi_i$ .

**Definición 4.12:** Sea  $Do_i$  un dominio. Sea  $D_{i_l}$  una dimensión y sea  $At_{i_{j_l}}$  un atributo de dimensión, donde  $At_{i_{j_l}} \in Do_i$ , y  $D_{i_l} \rightarrow Do_{i_{j_l}}$ . Sea  $\xi_k$  un concepto, donde  $\xi_k \in \Gamma^{def}$  y  $Do_i \rightarrow \xi_k$ . Sea  $CAt_l$  un atributo calculado, entonces  $At_{i_{j_l}} \rightarrow \xi_k$  o  $CAt_l \rightarrow \xi_k$  y  $\forall l = 1, \dots, 0$ , donde  $CAt_l \rightarrow (\xi_k, D_i)$ .

En el XBRLDM un dominio consta de dimensiones, y estas de *domain-members*, en el MDM un dominio consta de dimensiones y atributos calculados de una dimensión y de un dominio. Los conceptos del dominio, de una dimensión y de sus atributos calculados son jerárquicos (Hernández-Ros y Wallis, 2006; Schmehl y Ochocki, 2009; Santos y Castro 2011a, 2011c). En este modelo, las jerarquías pueden ser utilizadas para validaciones de los conceptos y con semántica del usuario de negocio. Esto significa que en el MDM los conceptos (atributos de dimensión y medidas) de una dimensión están organizados en un árbol jerárquico e interconectado. En el ejemplo, el concepto 'Real estate 'no real estate' del dominio *DAssets\_Estate* es la raíz de los conceptos 'Real estate' y 'No real estate'. Cada concepto puede tener asociada una operación de comparación (en la raíz), y una operación, "+", "-" (en las hojas). A diferencia del XBRLDM, el MDM utiliza atributos calculados para obtener un hecho, pero el XBRLDM no calcula los hechos, sólo sus validaciones.

Por tanto, para obtener el mapeo o transformación entre los dos modelos, un hecho tiene que resolver una cierta regla de validación definida con respecto a un atributo calculado. La validación de jerarquías en XBRLDM utiliza el *Linkbase* de cálculo (operación de la especificación XBRL con sólo una dimensión) (Engel et al., 2008; Santos y Castro, 2011a, 2011c), sin embargo en el grupo de *Eurofiling*, en su guía de buenas prácticas, recomiendan la utilización del *linkbase* de fórmulas (Morilla 2008; XBRL International, 2009; Fischer, 2011), que serán analizados en el siguiente capítulo de esta tesis.

La taxonomía dimensional de XBRLDM (XDT) define dos tipos de dimensiones (Hernández-Ros y Wallis, 2006; Schmehl y Ochocki, 2009). Las dimensiones pueden ser explícitas o implícitas, *Typed* en XBRLDM. Una dimensión explícita tiene definidos los atributos de dimensión en una manera explícita en el modelo de metadatos. Una dimensión es definida como implícita cuando los atributos de dimensión no están explícitamente definidos en el modelo de metadatos, sin embargo pertenecen a un dominio particular. Si una dimensión es implícita no es posible establecer jerarquías en el XBRLDM (*XBRL Dimensional Taxonomies*, XDT). Esto se puede formalizar mediante las siguientes dos definiciones:

**Definición 4.13:** Sea  $D_i$  una dimensión,  $\forall i=1, \dots, n$ ,  $D_i \in D$ , donde  $D$  es el conjunto de dimensiones. Sea  $\xi_j$  un concepto,  $\forall j = 1..m$ . Sea  $Do_{i'}$  un dominio, donde  $Do_{i'} = \cup_{j=1}^m \xi_j$ . Sea  $At_j$  un atributo de dimensión, donde  $D_i = \cup_{j=1}^{m'} At_j$ . Si  $At_j \rightarrow \xi_j$  y  $D_i \rightarrow Do_{i'}$ , entonces  $D_i$  es explícito, si y solo si  $D_i = \cup_{j=1}^{m'} \xi_j$ , donde  $m' \leq m$ .

**Definición 4.14:** Sea  $\varepsilon_{o_i}$  un tipo de dato,  $\forall i=1, \dots, m$ ,  $\varepsilon_{o_i} \in o$ , donde  $o$  es el conjunto de tipos de datos de la especificación o definidos por el usuario. Sea  $\xi_j$  un concepto, donde  $\xi_j \rightarrow Do_k$ ,  $\forall k = 1, \dots, o$ , y  $Do_k \rightarrow \varepsilon_{o_i}$ .  $Do_k \in Do$ , donde  $Do$  es el conjunto de dominios. Sea  $D_o$  una dimensión,  $\forall o=1, \dots, m'$ , y  $D_o \in D$ , donde  $D$  es el conjunto de dimensiones. Donde  $D_o \leftrightarrow Do_k$ , entonces  $D_o$  es una dimensión implícita, si y solo si  $\xi_j \in \varepsilon_{o_i}$  y  $\xi_j \in Do_k$  y  $\xi_j$  si está definida estáticamente.

Cada concepto está asociado a 0 o un número no determinado de referencias. Las referencias son indicaciones a textos legales. Estas referencias sólo indican la referencia a la ley, directiva o circular y no son enlaces a URL's, o URI's (Engel et al., 2008; Santos y Castro, 2011a, 2011c).

En la especificación XBRL se permiten las tuplas o arrays de datos. Sin embargo, la guía de buenas prácticas del grupo de *Eurofilling* no recomienda su uso (CEN, 2013; Eurofilling 2015). De acuerdo a este grupo, el uso de tuplas complica el uso de la *linkbase* de fórmulas (analizadas posteriormente en esta tesis) a causa de que es imposible indicar el elemento a validar de una manera unívoca en el array. Además, este grupo muestra que la extensión de taxonomías sobre tuplas es más complejo,

ya que el mecanismo de añadir conceptos desde tuplas es más complejo en XML. Sin embargo, en el MDM un array es considerado como otra dimensión.

En el XBRLDM, un hecho debe ser definido con el mismo tipo de período de tiempo que los conceptos involucrados en el hecho (*Engel et al., 2008*) y por tanto, esta definición es transferida al MDM. Además, en el XBRLDM un hecho es definido como un conjunto de pares (*dimensión/domain-member*) y un concepto básico (*primary ítem*). En el MDM una tabla de hechos consta de un conjunto de hechos y medidas y estos hechos son determinados por un conjunto de pares <dimensión/atributos de dimensión>, incluyendo el dominio base como adicional dimensión, y/o con atributos calculados. Por ejemplo en la *Figura 4.2* si elegimos la fila: “*BNP Paribas - The bank itself - Real Estate - 10,000.00*”. Esto es equivalente al hecho F(5,1,1) de la *Figura 4.3*. Entonces, el hecho F(5,1,1) es la unión de <Entity, “BNP Paribas”>, <Assets\_Estate 1, “Real estate”>, <Loans\_1, “The bank itself”>, <Geography, Germany> and <Base dimension, Assets>.

Por tanto, se puede definir:

**Definición 4.15:** Sea  $\varphi_i$  un hecho,  $\forall i=1, \dots, m$ . Sea  $D_j$  una dimensión,  $\forall j=1, \dots, n$ . Sea  $At_{j_k}$  un atributo de dimensión,  $\forall k=1, \dots, 0$ . Sea  $Bc_l$  un concepto básico  $\forall l \in N$ . Sea  $CA_{j'_k}$  un atributo calculado,  $\forall j' \in N$ , y  $\forall k' \in N$ . Entonces  $Bc_l \cup (D_j, At_{j_k})_{j=1}^n \cup (CA_{j'_k})_{j'=0}^l \rightarrow \varphi_i$ .

Los hipercubos en el XBRLDM son restricciones en los hechos en el XDT (*XBRL Dimensional Taxonomies*), el cual indica la combinación válida de pares <dimensión, atributos de dimensión>. Un *hypercube* en el MDM es un conjunto de pares <dimensión, atributos de dimensión> y atributos calculados, definiendo uno o más hechos. En un *hypercube* un atributo calculado puede existir, pues este atributo es determinado por un concepto y la dimensión y es el resultado de operar con pares <dimensión, atributos de dimensión> y/o atributos calculados de una dimensión.

**Definición 4.16:** Sea  $\varphi$  un hecho,  $\forall i=1, \dots, m$ . Sea  $D_j$  una dimensión,  $\forall j=1, \dots, n$ . Sea  $At_{j_k}$  un atributo de dimensión,  $\forall k=1, \dots, 0$ , donde  $At_{j_k} \in D_j$ . Sea  $CA_{j'_k}$  un atributo calculado,  $\forall j' \in N$ , y  $\forall k' \in N$ . Entonces un hipercubo es  $(D_j, At_{j_k})_{j=1}^n \cup (CA_{j'_k})_{j'=0}^l \rightarrow \varphi$ .

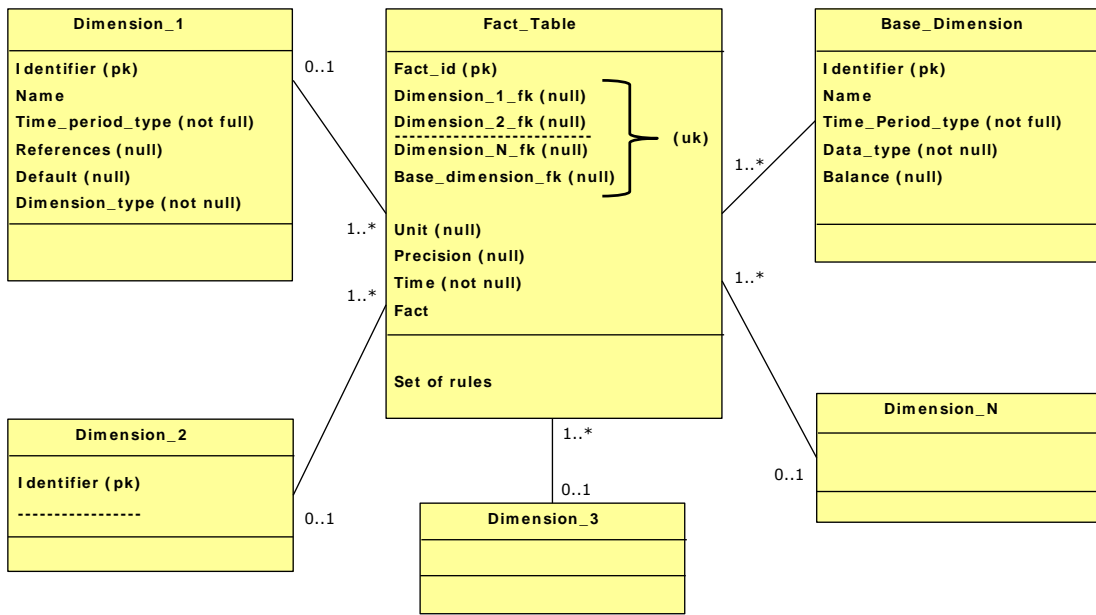
Un *hipercubo permitido* está definido como un hipercubo asociado con un concepto básico que determina un hecho. Un *hipercubo prohibido* está definido como un hipercubo asociado con un concepto básico que no puede determinar ningún hecho. En la especificación XBRL es obligatorio definir todos los hipercubos de los hechos mostrados en el informe. Como tal, se puede definir:

**Definición 4.17:** Sea  $\varphi_i$  un hecho,  $\forall i=1, \dots, m$ . Sea  $D_j$  una dimensión,  $\forall j=1, \dots, n$ . Sea  $At_{j_k}$  un atributo de dimensión,  $\forall k=1, \dots, o$ , donde  $At_{j_k} \in D_j$ . Sea  $CA_{j'_k}$  un atributo calculado, donde  $\forall j' \in N, \forall k' \in N$ . Sea  $B_{c_l}$  un concepto básico,  $\forall l=1, \dots, p$ . Entonces un hipercubo está permitido, sí y solo sí  $B_{c_l} \cup (D_j, At_{j_k})_{j=1}^{n'} \cup (CA_{j'_k})_{j'=0}^l \rightarrow \varphi_i$ .

Un *hipercubo prohibido* se define como un hipercubo asociado con un concepto básico que no determina ningún hecho.

**Definición 4.18:** Sea  $\varphi_i$  un hecho,  $\forall i=1, \dots, m$ . Sea  $D_j$  una dimensión,  $\forall j=1, \dots, n$ . Sea  $At_{j_k}$  un atributo de dimensión,  $\forall k=1, \dots, o$ , donde  $At_{j_k} \in D_j$ . Sea  $CA_{j'_k}$  un atributo calculado,  $\forall j' \in N, \forall k' \in N$ . Sea  $B_{c_l}$  un concepto básico,  $\forall l=1, \dots, p$ . Entonces un hipercubo es prohibido sí, y solo sí  $B_{c_l} \cup (D_j, At_{j_k})_{j=1}^{n'} \cup (CA_{j'_k})_{j'=0}^l \rightarrow \emptyset$ , donde  $\varphi_i \neq \emptyset$ .

La *Figura 4.4* muestra el modelo en estrella del MDM en el PIM. Cada dimensión tiene los siguientes atributos: *Identifier* (o nombre corto), *name* (o descripción del concepto), *time period type*, *references* (puede ser nulo), *default* (si el concepto es el por defecto). La dimensión base o dimensión de conceptos básicos tiene los siguientes atributos: *Identifier* (o nombre corto), *name* (descripción del concepto), *time period type*, *balance* (puede ser nulo). La tabla de hechos tiene los siguientes atributos: *Identifier* (*primary key, pk*), *reference to the dimension* (*foreign key, fk*), *unit* (Euros, Libras, Dólares,...), *precisión*, *time* y *fact*. La unidad de precisión puede ser *null*, si el hecho es no numérico. En la tabla de hechos, el conjunto de referencias externas a dimensiones y a la dimensión base, se asocia con una clave única (*unique key, uk*).



**Figura 4.4 Modelo en estrella en el PIM**

A continuación se describe un primer algoritmo para la extracción de los metadatos desde el CIM al PIM:

```

start
  begin
    leer plantilla;
    repeat
      obtener conceptos; obtener dominios; leer plantillas;
    until ¿no más plantillas?;
    repeat
      Revisar dominios; revisar conceptos; obtener conceptos básicos;
    until ¿no redundancia?;
    repeat
      Establecer raíz y hojas en la estructura en árbol de los conceptos;
    Until ¿Es la estructura arbórea la correcta?;
    repeat
      Obtener dimensiones: obtener atributos de dimension;
    until ¿es semánticamente correcto?;
    repeat
      Obtener atributos calculados;
    until ¿La estructura en árbol está de acuerdo con el dominio?;
      Obtener hechos permitidos; obtener hechos prohibidos; Añadir nuevos
      atributos calculados (restricciones de usuario);
    end
  end
end

```

#### **Algoritmo 4.1 Transformación desde el CIM al PIM**

Los conceptos y los dominios son obtenidos desde las plantillas en las hojas de cálculo y son entonces chequeados para evitar redundancias. Los *conceptos básicos* son obtenidos en esta fase. Después se establece la estructura en árbol en el dominio, excepto en el dominio de los conceptos básicos. Ahora es necesario verificar que la estructura en árbol está bien construida. A continuación se recogen las dimensiones y atributos de dimensión. También es necesario verificar si la semántica de las dimensiones es correcta (en un dominio hay de 1..\* dimensiones con atributos de dimensión) con atributos de dimensión que son conceptos del mismo dominio. Un concepto puede pertenecer a varias dimensiones de un dominio. El siguiente paso es obtener los atributos calculados y chequear que respetan la estructura en árbol de los conceptos en el dominio. Después se obtienen los hechos prohibidos y permitidos y finalmente se añaden los nuevos atributos calculados por el usuario experto para definir nuevas reglas.

Hasta aquí se han definido todos los elementos para la implementación en una base de datos específica. La *Figura 4.5* muestra la tabla de hechos de la *Figuras 4.2 y 4.3*,

que es un modelo en estrella en el PIM, donde 'To Other' es 'To other banks', 'RE' es 'Real Estate', y 'NRE' es 'No Real Estate'.

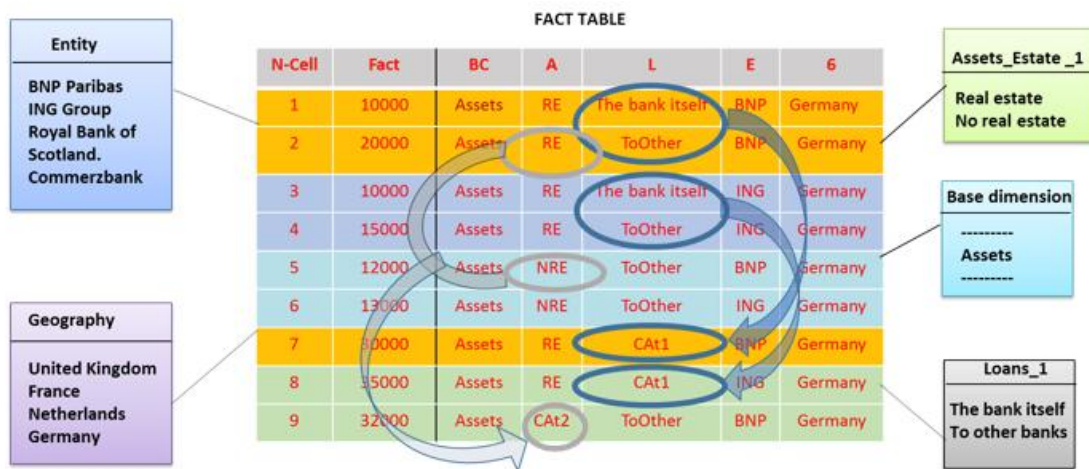


Figura 4.5 Ejemplo de tabla de hechos con atributos calculados

Como se explica arriba, en este modelo hay dos atributos calculados:  $CAT_1 = ('To\ bank\ itself' + 'To\ other\ banks')$   $|_{Loans_1}$ ,  $CAT_2 = ('Real\ Estate' + 'No\ Real\ Estate')$   $|_{Assets\ Estate_1}$ . Para finalizar, es posible verificar que es correcto el hecho número 7, y sin embargo, es incorrecto el 8. Además, es posible inspeccionar un hipercubo permitido como es el hecho 7, que está definido como  $\{(BD, Assets), (Assets\ Estate_1, Real\ Estate), CAT_1, (Entity, BNP), (Geography, Germany)\}$ .

La Figura 4.6 muestra el conjunto de objetos definidos en el MDM en UML. En esta figura se puede ver que un dominio consta de conceptos, y que un concepto puede ser concepto básico o concepto no básico. Un dominio puede tener dimensiones. Un atributo de dimensión puede pertenecer a diferentes dimensiones. Un concepto, no básico, puede ser un atributo de dimensión o calculado, pero determina un dominio. Un hipercubo consta de pares (dimensión, atributos de dimensión) y/o atributos calculados. Los conceptos son de tipo usuario o tipo estándar.

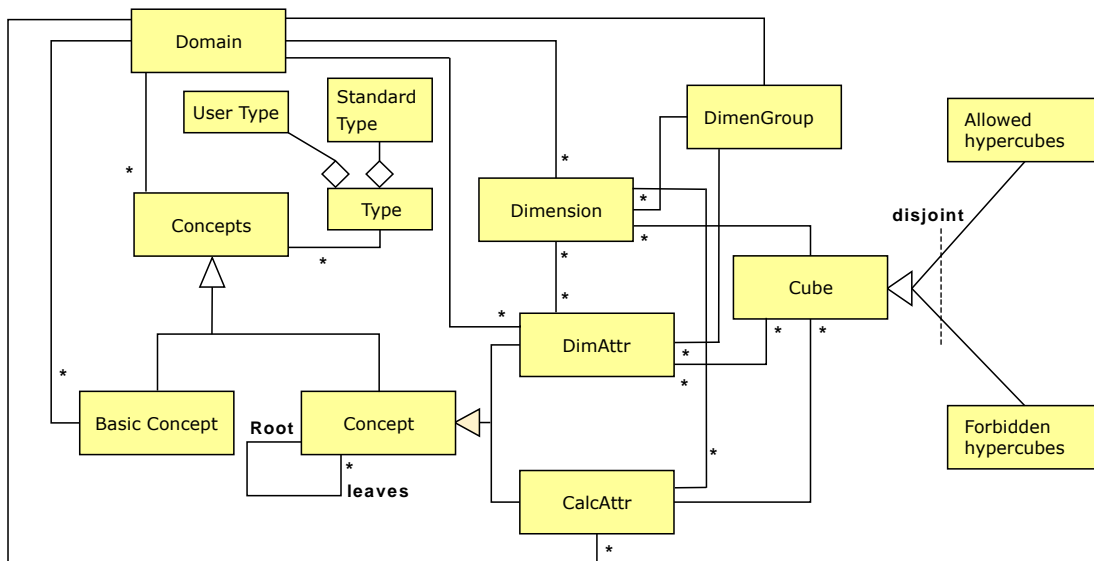


Figura 4.6 Resumen de los artefactos en UML del modelo de datos en el PIM

Hasta aquí se ha mostrado la estructura, definiciones y restricciones de usuario en el PIM. En la siguiente sección, se verifican algunas restricciones del diseño.

#### 4.4. Cuestiones Semánticas acerca del diseño de los Metadatos en el PIM

En esta sección se analizan ciertas cuestiones asociadas con los informes semánticos en el nivel PIM. El objetivo de este nuevo trabajo y de la tesis es obtener soluciones a estas cuestiones que, de no ser resueltas en el MDM (el PIM), serán heredadas en el SQL (el PSM) y producirán errores en *run-time* o bien resultados incorrectos. En primer lugar se analizan las jerarquías. En segundo lugar, se estudian los problemas de las definiciones del tipo de período de tiempo. Como tercer punto se analizan algunas cuestiones de las dimensiones implícitas, para finalizar con los casos en los que varios atributos calculados afectan a un hecho.

Las jerarquías se utilizan en XBRLDM para dar semántica a los metadatos. Una dimensión en el XBRLDM está compuesta de conceptos (en el MDM, atributos de dimensión y atributos calculados o medidas de la dimensión), y estos conceptos tienen una estructura jerárquica en árbol, con sólo una raíz, que en general es el concepto por defecto. En una primera aproximación es necesario testear, si sólo

existe una raíz y la estructura no tiene ciclos. Debe ser testado que los conceptos sin hojas no tienen ninguna operación de validación.

El tipo de período de tiempo de los atributos de dimensión y de los atributos calculados del mismo dominio debe tener el mismo tipo de período de tiempo. Pero de la misma manera, un concepto básico también tiene un atributo con un tipo de período de tiempo y deberá ser el mismo tipo que el de los atributos de dimensión que forman el hecho. ¿Cómo asegurar que ambos tipo de período de tiempo son el mismo? Y, si ellos no son el mismo tipo, ¿cuál es el correcto tipo? Cuando los períodos de tiempo tienen diferentes tipos, una solución de los analistas de IS es crear una dimensión temporal y evitar los problemas en la transformación, en la presentación de éstos, o en la entrada a otros aplicativos.

Aunque los atributos de dimensión de las dimensiones implícitas no son validados en el XBRLDM, debería verificarse si no se corresponden con el tipo específico, y en su caso mostrar un error. El atributo de la dimensión implícita se rellenará con el informe y no estará definido en el conjunto de conceptos de la taxonomía (metadatos). Así, por ejemplo, si la dimensión es una entidad financiera, ello debe ser testado en el informe (*XML instance* o *XBRL instance document*), y deberá contener un conjunto de caracteres. Desde el punto de vista semántico, se debería chequear que el nombre de la entidad tenga una estructura regulada por un estándar, o si puede ser predefinido, y en ese caso testado en *run-time*.

Una diferencia entre el MDM y el XBRLDM es el tema de la validación. En el MDM un hecho puede ser obtenido a través de un atributo calculado. Sin embargo en el XBRLDM un hecho no se calcula, sólo se valida. Si la especificación es mapeada al MDM, entonces un atributo es utilizado para validación. Pero hay una diferencia significativa, un hecho puede ser verificado por uno o más atributos calculados.

## 4.5. Análisis en el Modelo específico de la Plataforma (PSM)

Esta sección analiza la transformación desde el PIM al PSM. La *Tabla 4.2* muestra esta transformación. En esta plataforma, algunos constructores no son transformados, ya que no se usan. La columna 1 muestra los conceptos en el PIM, la columna 2 muestra la transformación en el PSM y en la columna última se añaden algunos comentarios.

**Tabla 4.2 Transformación del PIM al PSM**

PIM	PSM	Comentarios
Dimensión base	Tabla	En el PSM da igual que sea una dimensión base o una dimensión normal.
Concepto básico	Atributo de dimensión	Tiene asociado un tipo de dato, un tipo de período de tiempo y tipo balance. El balance es una restricción.
Dimensión (explícita/implícita)	Tabla	Es explícita si los atributos de dimensión están definidos, e implícita si no lo están.
Atributo calculado	Atributo calculado	Un agregado de los conceptos hoja de la dimensión. Esta regla obtiene los atributos calculados.
Atributo de dimensión	Atributo de dimensión	No es un agregado.
Referencias	Atributo de dimensión	Referencias de los conceptos.
Hechos	Tabla de hechos	Es implícita en el diseño.
Hechos::= - Dimensión/atributo de dimensión - Concepto básico - Atributo calculado	Hecho	Incluyen atributos de dimensión y atributos calculados (medidas).
Hecho permitido	Aserción	Restricción de usuario.
Hecho prohibido	Aserción	Restricción de usuario.

Esta tabla muestra la transformación de la *Figura 4.4* (el MDM) en un modelo de datos relacional, utilizando la tecnología ROLAP. En la primera fase se obtienen las tablas y los atributos de dimensión. Seguidamente se definen las aserciones y las reglas. Se define un conjunto de aserciones para validar los hechos o medidas que están permitidos o prohibidos por el usuario experto. Además se obtienen los atributos calculados. El *Algoritmo 4.2* la esboza con la obtención de los atributos calculados.

```

start
  leer dimensiones;
  repeat
    leer conceptos de la dimension del dominio;
    repeat
      if El concepto no tiene comparación asociada
        then el concepto es un atributo de dimensión;
      else if El concepto no tiene hojas
        then El concepto es un atributo de dimensión;
      else begin
        El concepto es un atributo calculado;
        leer operación de comparación;
        leer conceptos hoja y operaciones asociadas;
      end
    leer conceptos de la dimensión del dominio;
  until no más conceptos de la dimensión;
  leer dimensiones;
until no más dimensiones;
end

```

#### **Algoritmo 4.2 Implementación del modelo de metadata**

Hasta aquí se han definido todos los elementos necesarios para su implementación en una base de datos específica. Las pruebas de concepto (POC) de esta tesis se muestran en *Openfiling (2015)*, *Santos y Castro (2011b; 2012a)* y *Santos y Nieto (2014, 2015)*.



# CAPÍTULO 5

---

Fundamentación y  
Conceptos Básicos de la  
propuesta de fórmulas



# CAPÍTULO 5. Fundamentación y Conceptos Básicos de la propuesta de fórmulas

---

## 5.1. El Universo del Discurso

Este capítulo se centra en la transformación del modelo de datos XBRL (XBRLDM) de la especificación de fórmulas y el modelo de datos multidimensional (MDM), además de su automatización (*Piechocki et al. 2007; Felden 2007*).

El MDM es un modelo que combina objetos, dimensiones (jerarquías), medidas y atributos para representar los problemas del mundo real (*Kimball, 1996-2004; Inmon, 2005; Jarke et al., 2003*). Además, este modelo es el centro del *On-line Analytical Processing* (OLAP), el cual requiere consultas complejas que pueden ser resueltas en el MDM en tiempo real.

La conversión entre los mencionados modelos será hecha utilizando el paradigma del *Model Driven Architecture* (MDA), el cual asegura y resuelve el problema de la interoperabilidad entre sistemas heterogéneos. El MDA tiene tres fases o niveles (OMG, 2015): El *Computational Independent Model* (CIM), el *Platform Independent Model* (PIM) y el *Platform Specific Model* (PSM). Como se analizará en detalle seguidamente, el modelo conceptual (el PIM) se obtiene haciendo ingeniería inversa desde la *Formula Linkbase* de la especificación XBRL (el CIM). Después se obtiene el modelo lógico en el estándar SQL (el PSM), para finalizar obteniendo el código.

UML/MDA (OMG, 2015) es una herramienta poderosa, que ha ayudado en diferentes áreas de Tecnología de la Información (*Information Technology, IT*) a modelar sistemas estructurados y robustos. Sin embargo, las técnicas de verificación y validación del software no están soportadas en el estándar MDA (*Cuadra et al., 2011; Aljumaily et al., 2014*). Por esta razón, la principal contribución de esta tesis relacionada a Sistemas de Información se focaliza en la transformación automática de los modelos, formalización de las reglas de validación en cada fase, y con el objetivo de validar un alto número de reglas en cada fase, las cuales representan cientos de conceptos, dimensiones, atributos de dimensión, cubos, etc. Además, se muestra una nueva validación de demostración del software, que puede ser aplicada a cualquier campo de IT en la cual la interoperabilidad semántica sea requerida.

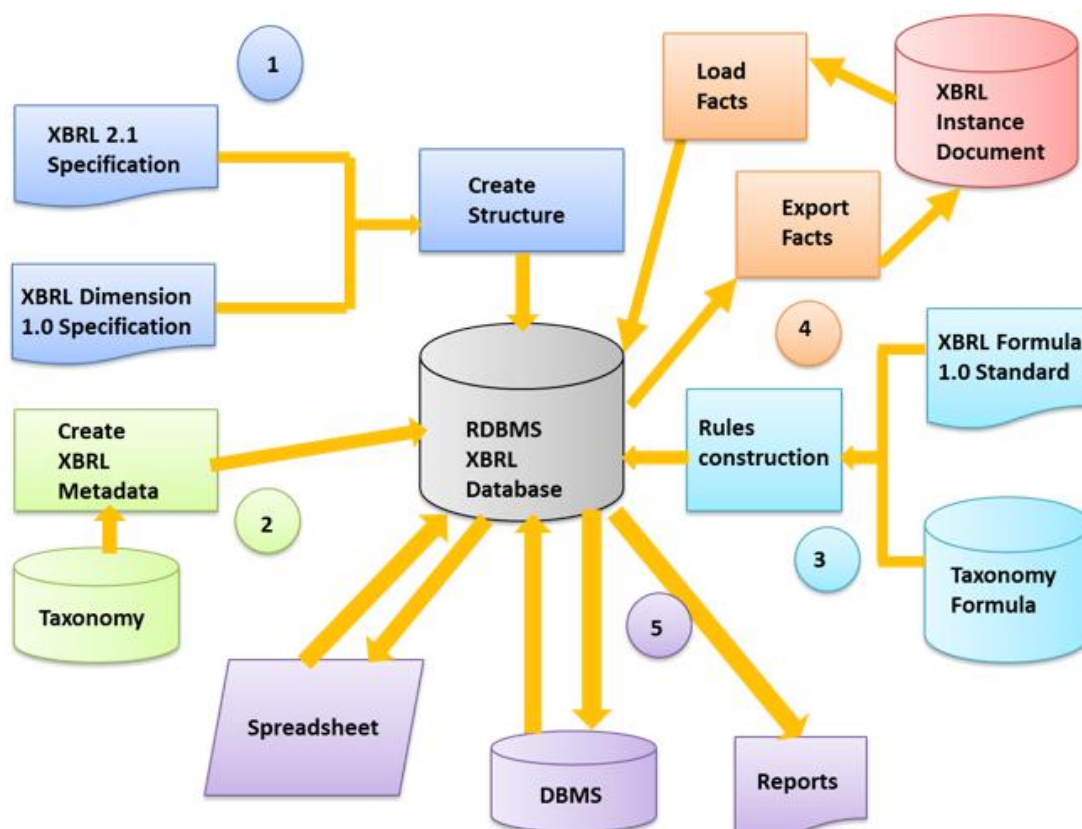


Figura 5.1 Estructura general de la transformación XBRL versus un RDBMS

La Figura 5.1 presenta la estructura general de la propuesta de este capítulo, a analizar.

En esta figura se enumera y recodifica a la vez, con colores, los diferentes pasos del proceso de mapeo. En el paso 1 se obtiene la estructura desde la especificación XBRL 2.1 (Engel et al. 2008) y XBRL Dimensions 1.0 (Hernández-Ros y Wallis, 2006), y las entidades, atributos y relaciones, creadas desde el XBRLDM (Santos y Castro, 2010, 2011b, 2011c). Algunas de estas entidades son: Dimensiones (*explicit dimensions* en el XBRLDM), atributos de dimensión (*Dim Attributes, members* en el XBRLDM); el enlace o relación entre dimensiones y atributos de dimensión, *Relation\_Dim\_AttrDim* (*arc* en XBRLDM); conceptos básicos *primary Items* en XBRLDM); grupo de dimensiones o *Context* (en XBRLDM). La relación entre *context*, dimensiones y atributos de dimensión es *Context\_Dim\_AttrDim* (*arc* en XBRLDM); la unidad de medida, *unit*, el conjunto de cabeceras de los documentos instancias o informes XBRL (utilizada para mapear desde el RDBMS a una instancia XBRL), llamada *Header*; y finalmente el *DataPoint*, que es utilizado para un formato común por el usuario experto, realmente es la tabla de hechos.

El paso 2 del proceso (*Figura 5.1*) crea los metadatos XBRL cargando la Base de Datos con las dimensiones, atributos de dimensión y relaciones (*Santos y Castro, 2011b*). A este fin se crea un API, que genera las sentencias SQL desde la taxonomía (*Openfiling 2015*). El paso 3 se corresponde con la generación del marco de la validación, que es parte del foco de esta tesis, y propuesta, que se explica en mayor detalle en las subsecuentes secciones.

En el paso 4 se cargan los datos desde el documento instancia XBRL (el informe) en la base de datos. De acuerdo a este proceso, el documento instancia se lee y se rellena la base de datos con los hechos y contextos. Similarmente, se comienza un proceso paralelo para la creación del documento instancia XBRL desde la instancia de base de datos. Finalmente, el paso 5 obtiene la transformación de los hechos de la base de datos a otras herramientas (y viceversa) comúnmente utilizados en IT, además de los usuarios expertos.

En la siguiente sección se analiza el XBRLDM para introducir las fórmulas. La introducción a la especificación XBRL de fórmulas se da en la sección 3. La sección 4 propone la arquitectura MDA, las definiciones y formalización matemática en el MDM desde el XBRLDM. La sección 5 analiza cuestiones semánticas de la especificación XBRL de fórmulas. Y la sección 6 presenta la implementación de esta investigación a través de Pruebas de Concepto (POC) y análisis de las fases del PIM al PIM y al código. La validación de la propuesta se verá en el siguiente capítulo.

## 5.2. Modelo de datos y Cálculos en XBRL

La información semántica XBRL está separada del software de la aplicación, utiliza y extiende el estándar XML. Un informe o documento instancia XBRL referencia a un conjunto de esquemas XBRL o XML. Este conjunto de esquemas en XBRLDM se llama *Discoverable Taxonomy Set* (DTS), y especifica los conceptos económicos necesarios. Cada *XBRL Schema* tiene un *role*, analizados en el capítulo 3, sección 1 de esta tesis (*linkbases* en el *XBRLDM*). El *role, calculation* provee relaciones de cálculo simples entre diferentes elementos (conceptos básicos, dimensiones y atributos de dimensión), pero no permite fórmulas o complejas expresiones. Este *role* sólo permite combinaciones válidas que son permitidas por una *Dimension*; mientras

el *role formula*, el cual constituye la nueva contribución de esta tesis, soporta expresiones complejas basadas en XPath, el cual puede ser aplicado al documento instancia para la validación de la información.

Un documento instancia XBRL consta de una cabecera (*header*), con las referencias *XML Schemas*, además de los roles asociados (*linkbases*), el conjunto de dimensiones agrupadas por el contexto, y los ítems (o hechos en el MDM). A través del contexto se asocia cada ítem (en el XBRLDM) en el documento instancia, dimensiones y concepto básico (*primary ítem* en el XBRLDM). Cada *XBRL* o *XML Schema* debe tener 0,1 ó *n* roles dependiendo del *role* (como se ha visto en capítulos anteriores). En esta especificación, una *primary ítem* define el esquema de hecho sin dimensión en el MDM (*Hernández-Ros y Wallis, 2006; Martín Quetglás, 2006; Engel et al., 2008; Schmehl y Ochocki, 2009*). Un concepto básico, *primary ítem*, define un esquema de hecho, pero con una unidad de medida, un tipo, y si es numérico, una precisión de un número específico de posiciones decimales. Por ejemplo, una *primary ítem* puede ser un *Asset* (activo) que puede pertenecer a la dimensión de *equities* (capital), o *risk* (riesgo). Un contexto contiene información acerca de un hecho o conjunto de hechos. Además, puede estar asociado con una entidad o institución financiera para un período específico, e incluye un significado de negocio a través del *segment* (segmento) y un *scenario* (escenario), ambos conjuntos de dimensiones en el MDM. El *scenario* muestra pares específicos <dimensión/atributos de dimensión> desde la lógica de negocio.

### 5.3. Introducción a la especificación XBRL formula

En esta sección se analiza la especificación XBRL de fórmulas desde la perspectiva del MDM, y se desarrolla un conjunto de reglas y definiciones.

La especificación define un conjunto de validaciones sobre ciertos hechos dados (*XBRL International, 2009*). Estas definiciones tienen una sintaxis y un conjunto de reglas. La versión previa a la actual XBRL 2.1 (*Engel et al., 2008*) era la XBRL 1.0 (*Hamscher, y Kannon, 2000*), la cual está basada en jerarquías con estructuras en árbol, ya que XML, en el cual está basado esta especificación, es jerárquico, de tal forma que la versión previa del XBRL operaba con un conjunto de limitaciones. Un ejemplo de documento instancia XBRL generado bajo estas limitaciones se presenta

en la *Figura 5.2 (XBRL International and Novartis International AG, Samples XBRL 1.0, 2002)*, consistiendo en un conjunto de conceptos básicos, con un conjunto de pares <dimensión/atributo de dimensión> (Santos, y Castro, 2011a, 2011c). El ejemplo se presenta gráficamente en la *Figura 5.3*.

Consolidated Balance Sheet (at December 31, 2001 and 2000)			
Assets	Notes	2001 CHF millions	2000 CHF millions
Long-term assets			
Tangible fixed assets	8	9060	9030
Intangible assets	9	6548	5830
Investment in associated companies	11	6715	1531
-----	----	----	----
Deferred taxes	12	3235	3265
Other financial assets	13	7027	5601
<b>Total long-term assets</b>		<b>32585</b>	<b>25257</b>
Current assets			
-----		----	----
-----		----	----
<b>Total current assets</b>		<b>34200</b>	<b>32939</b>
<b>TOTAL ASSETS</b>		<b>66785</b>	<b>58196</b>

Figura 5.2 Ejemplo de informe jerárquico

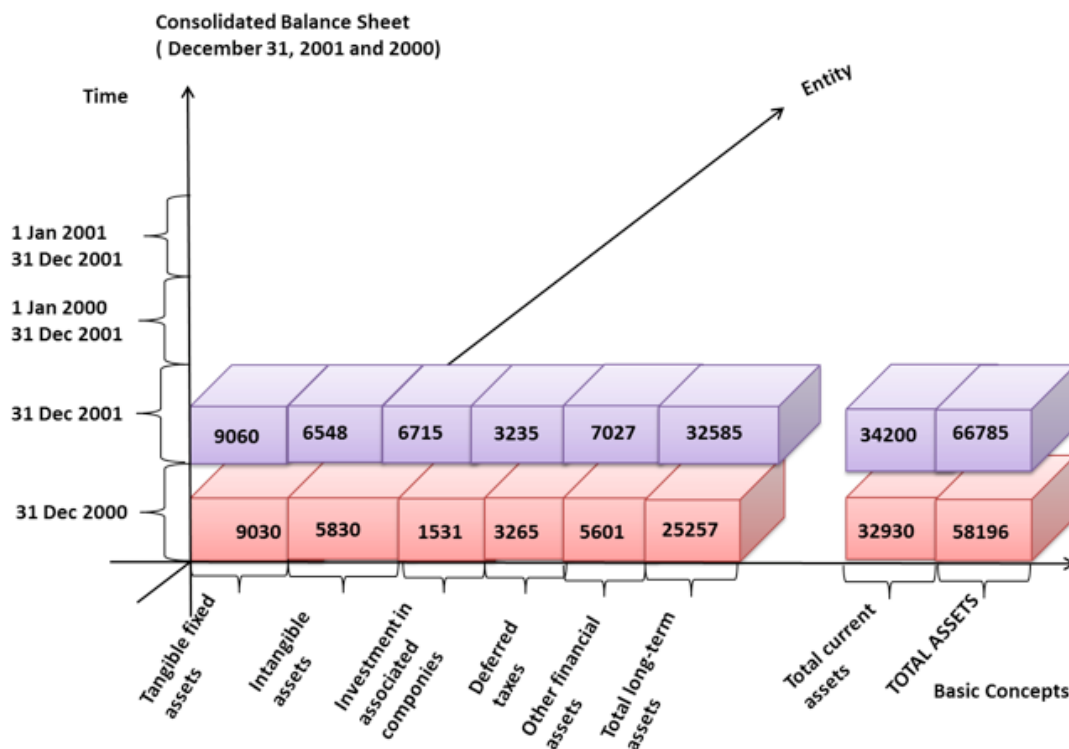


Figura 5.3 Diseño gráfico del informe de la Figura 5.2

En la Figura 5.4 se muestra el informe con su estructura jerárquica. Sin embargo, con la especificación de dimensiones XBRLDM 1.0 no se utiliza exhaustivamente, ya que sólo se utilizan las jerarquías. Esto hace que esté limitado el uso de operaciones matemáticas a una sola dimensión, o en otras palabras, no es posible utilizar diferentes contextos. Por ejemplo, mientras en la Figura 5.3 es posible comparar dos hechos en la misma fecha (es decir el mismo contexto), no es posible comparar dos hechos de diferentes fechas. No obstante, hay una dimensión en cada nivel de la jerarquía con atributos de dimensión y conceptos básicos. En general, cada llave de una jerarquía es la suma de sus llaves (Figura 5.4). Es posible operar con conceptos básicos que tengan el mismo conjunto de pares dimensión-atributo de dimensión (es decir, el mismo contexto). De tal forma, que es posible encontrar que *total assets* sea igual a la suma del total de *total long-term assets* y *total current assets*; sin embargo, sólo para el 31 de diciembre de 2001 y no entre el 31 de diciembre de 2001 y de 2000. Por ejemplo, no es posible validar *total assets* para el 31 de diciembre de 2000 siendo la suma de *total long-term assets* para el 31 de Diciembre de 2001, y *total current assets* para el 31 de diciembre de 2001, como muestran las Figuras 5.3 y 5.4.

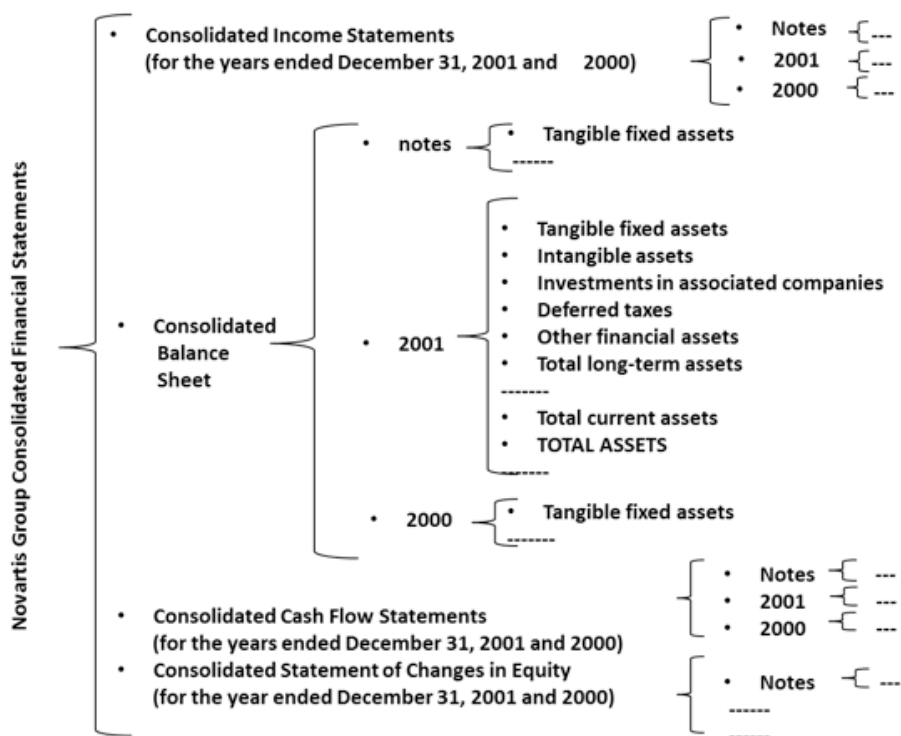


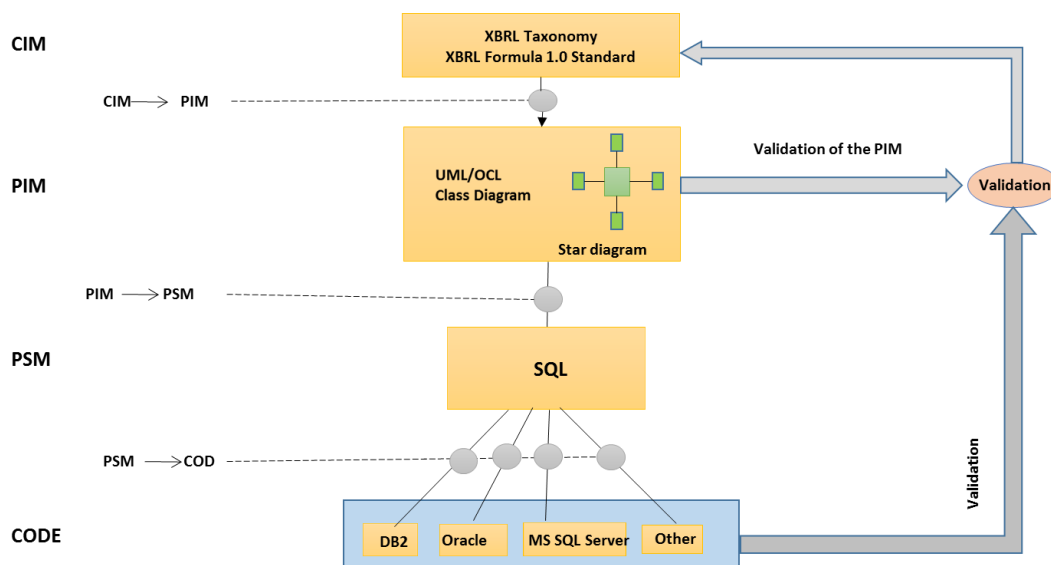
Figura 5.4 Estructura jerárquica del informe de la Figura 5.2

El *role de calculation* trabaja solamente con jerarquías (XBRLDM), o en otras palabras, con un solo contexto y en ningún momento con múltiples contextos (Santos, y Castro, 2010, 2011a, 2011c). En XBRL 1.0, las dimensiones son *entity*, *period of time*, *basic concepts* (es decir el conjunto de conceptos básicos) y el conjunto de dimensiones definidas por el *scenario* (0..n dimensiones) y el segmento (una dimensión). La versión actual del estándar XBRL se creó para responder a los requerimientos de negocio e incluir más información con complejas estructuras y reglas de validación (Hernández-Ros, y Wallis, 2006; Engel et al., 2008).

La validación constituye un objetivo importante de cualquier documento instancia XBRL, y las operaciones lógicas y aritméticas, además de funciones basadas en operaciones previas, son necesarias en cualquier validación. El avance más importante de la especificación de fórmulas XBRL es la disponibilidad de ejecutar operaciones con hechos diferentes y con diferentes dimensiones y contextos. Si la validación es ejecutada en destino, podría ser ejecutada con cualquier lenguaje, como COBOL, Java o .Net. Sin embargo, como la validación ocurre en origen, se ve la necesidad de crear un estándar. Por todo ello se llegó a la publicación de las especificaciones *XBRL Dimensions 1.0* (Hernández-Ros, y Wallis, 2006), y *XBRL fórmula 1.0* (XBRL International, 2009).

## 5.4. Definiciones y semántica de la especificación de fórmulas XBRL

Esta tesis propone una aproximación al MDA en el mapeo entre la especificación de fórmulas XBRL y la base de datos (*Figura 5.5*).



**Figura 5.5** Conversión automática de fórmulas entre instancias XBRL y bases de datos, utilizando MDA

El modelo CIM en este trabajo de investigación es el conjunto de documentos instancia XBRL, las taxonomías y la especificación de fórmulas *XBRL Formula* (secciones 5.2, y 5.3). En esta sección se obtienen las reglas y definiciones desde el CIM, a través de la especificación de fórmulas XBRL. El PIM utilizado en esta propuesta está basado en UML, el cual es el modelo en estrella (el MDM) (*Boixo y Flores, 2005; Piechocki et al., 2007; Felden, 2007; Schmehl y Ochocki, 2009*). Se recogen de una forma automática el conjunto de restricciones, dimensiones y atributos de dimensión desde una taxonomía, como se muestra en el algoritmo de la *Figura 5.12*. El PSM es el conjunto de procedimientos almacenados o programas en Cobol, C++, etc. El algoritmo mostrará la transformación desde el PIM al PSM. En el *código 5.1* se mostrará un ejemplo del código SQL.

El proceso de validación se divide en **dos fases** (Figura 5.5). En la **primera** se realiza el test del modelo en estrella / MDM (el PIM) desde las taxonomías XBRL y los informes XBRL. En la **segunda fase** se valida el conjunto de procedimientos almacenados utilizando la tecnología ROLAP; se compara el modelo en estrella con las dimensiones, atributos de dimensión, y restricciones en el MDM, y el XBRLDM (Capítulo 6).

En esta sección se propone un conjunto de once definiciones y reglas utilizadas en esta especificación, pero en el MDM desde el XBRLDM. Como la propuesta se basa en el MDM (el PIM), el autor de esta tesis simplificará los conceptos y las reglas haciéndolos más comprensibles a los usuarios. Adicionalmente, esta sección presenta una nueva dimensión a la propuesta, ofreciendo una nueva formalización matemática (lenguaje formal en el MDM). Finalmente, al final de esta sección, se dará un ejemplo detallado.

Las fórmulas están localizadas en uno o más documentos, y cada documento consta de un conjunto de aserciones. Cada documento instancia XBRL puede tener desde 0..\* documentos XML del rol de fórmulas, en el que cada uno consiste de un conjunto de 1..\* aserciones. Las aserciones se utilizan para verificar una expresión o condición. Los siguientes párrafos presentan las nuevas definiciones utilizadas en la especificación de fórmulas XBRL.

**Definición 5.1:** *Un hecho es un valor que representa una medida particular provista por la entidad que hace el informe.*

**Definición 5.2:** *Una aserción es una restricción general que hace referencia a un hecho o conjunto de hechos.*

Un documento de fórmulas consiste en un conjunto de grupos de aserciones. Aunque cada grupo de aserciones puede pertenecer lógicamente a un específico documento instancia XBRL, no es posible asegurarlo. Cada definición de aserción debe tener un nombre en un documento XML de fórmula. Las aserciones se agrupan en conjuntos, cada uno de los cuales están relacionados lógicamente a un documento instancia XBRL y empezando con la etiqueta *assertionSet*. Así, cada documento instancia tiene que ser validado contra todas las aserciones de todos los documentos fórmula, ya pertenezcan al grupo de aserciones del documento instancia XBRL o no. Por esta razón, es muy común encontrar documentos instancia XBRL que deben ser validados

con una gran cantidad de conjuntos de aserciones no utilizados en ese documento instancia.

**Definición 5.3:** *Un filtro es un conjunto de dimensiones a las cuales la aserción está confinada.*

Un filtro es un conjunto de pares *dimension-member* en el XBRLDM o de pares *dimensión-atributo de dimensión* en el MDM (XBRL International, 2009a, 2009b).

**Definición 5.4:** *Un filtro implícito es una restricción inherente a todos los hechos circunscriptos por la aserción (por ejemplo, la aserción se restringe a los hechos circunscriptos por el conjunto de pares inherentes dimensión-atributo de dimensión).*

Los filtros implícitos como restricciones inherentes no están escritos en la aserción. Por defecto hay siempre filtros implícitos en una aserción, según el XBRLDM, ahorrando codificación. El parámetro *@implicitFiltering* a falso, indica que hay que definir los filtros implícitos, por defecto es verdad. El filtro implícito consiste en un conjunto de pares *dimensión-atributo de dimensión*, los cuales son comunes a todos los hechos de la aserción. Los filtros implícitos por defecto son *entity*, *period of time* y *unit*.

**Definición 5.5:** *Un filtro explícito es una restricción semántica, al cual todos los hechos, grupo de hechos o hechos de la aserción están circunscritos.*

De acuerdo a la sintaxis de la aserción, la restricción puede ser circunscrita a diferentes entornos en la aserción.

De acuerdo a la Real Academia Española, un concepto es una *idea que concibe o forma el entendimiento*. En el XBRLDM, un concepto es el que define a una dimensión, a un atributo de dimensión o a un concepto básico.

**Definición 5.6:** *Un concepto básico define el nombre de un concepto en el que tiene asociado un tipo de dato, una precisión y una unidad de medida (Santos, y Castro, 2011c).*

En el documento instancia, por ejemplo, del informe de las *Figuras 5.2, 5.3 y 5.4* (XBRL International and Novartis International AG, 2002), *intangible assets* es un concepto básico, el cual define un concepto financiero, tiene un tipo de dato monetario y una unidad de medida, la cual, en este caso es *iso4217:CHF*. En el ISO 4247, publicado por la Organización Nacional de Estándares, la moneda del franco suizo (usada en Suiza y en Liechtenstein), se define con el código del país CHF. Además, el concepto básico tiene una precisión indicada con el número de decimales usados para el cálculo, por ejemplo, *precision='18'*.

Un concepto básico puede ser analizado como un atributo de dimensión, *primary items*, o *basic concepts*.

**Definición 5.7:** *Una constante es un valor o expresión que produce un resultado y cuya expresión es determinista.*

Una constante puede ser, por ejemplo, *var\_BasicThreshold4*, donde esta variable es el nombre de una constante. Aunque deberá tener un valor de  $3 \text{ or } (4 \text{ idiv } 2 + 1)^* 1000$ , y éste debe ser único. El operador *idiv* indica división sin decimales.

Una aserción consta de la definición de la propia aserción, sus parámetros, constantes, precondiciones, variables y filtros. Una aserción tiene una entrada y una salida con el conjunto de hechos a validar y el resultado de la validación.

**Definición 5.8:** *Una variable es un elemento atómico de una expresión en una aserción el cual conlleva un hecho o un grupo de hechos.*

Una variable puede constar de dimensiones, atributos de dimensión, constantes, precondiciones y otras fórmulas (Morilla, 2008; XBRL International, 2009c; Fischer, 2011). Cada variable en cada aserción tiene un nombre, el cual empieza con el símbolo de \$. Las dimensiones y sus atributos constituyen los filtros explícitos. Una variable tiene parámetro en XBRLDM, los cuales en el MDM son atributos, ya que ellos definen propiedades de la variable.

Desde estas definiciones se pueden desarrollar las siguientes reglas:

**Regla 5.1:** Una variable puede tener dimensiones explícitas.

Sea  $v$  una variable. Sea  $DE_i$  una dimensión  $\in D$ , donde  $D$  es el conjunto de dimensiones y  $\forall i = 1, 2, \dots, n$ . Sea  $At_j$  un atributo de dimensión  $\in DE_i, \forall j = 1, \dots, n$ . Entonces  $v$  es  $\langle DE_k At_l \rangle_{k=0}^n, \forall l = 1, \dots, n$ .

**Regla 5.2:** Una aserción puede tener dimensiones implícitas (por ejemplo, estas que son inherentes al diseño) las cuales son omitidas en la aserción.

Sea  $\delta$  una aserción. Sea  $DI_i$  una dimensión  $\in D$ , donde  $D$  es el conjunto de dimensiones,  $\forall i = 1, \dots, n$ . Sea  $At_j$  un atributo de dimensión  $\in DI_i, \forall j = 1, \dots, n$ . Entonces  $\delta$  tiene  $\langle DI_k, At_l \rangle_{k=0}^n, \forall l = 1, \dots, n$ .

**Regla 5.3:** Una aserción puede tener dimensiones explícitas, las cuales aparecen en la aserción como un filtro explícito en el XBRLDM.

Sea  $\delta$  una aserción. Sea  $DE_i$  una dimensión  $\in D$ , donde  $D$  es el conjunto de dimensiones  $\forall i = 1, \dots, n$ . Sea  $At_j$  un atributo de dimensión  $\in DE_i, \forall j = 1, \dots, n$ . Entonces,  $\delta$  tiene  $\langle DE_k At_l \rangle_{k=0}^n, \forall l = 1, \dots, n$ .

**Regla 5.4:** Una variable, con sus atributos, filtros explícitos y filtros generales a toda la aserción (implícitos o explícitos) determinan de 0 a  $n$  hechos.

Sea  $\delta$  una aserción  $\langle DI_k, At_l \rangle_{k=0}^n, \forall l = 1, \dots, n$ , donde  $DI$  son dimensiones implícitas  $\in \delta$ . Además sea,  $\langle DE_k At_o \rangle_{k=0}^n, \forall o = 1, \dots, n$ , donde  $DE$  son dimensiones explícitas  $\in \delta$ . Sea  $v$  una variable, y  $\langle DVE_k At_p \rangle_{k=0}^n, \forall p = 1, \dots, n$ , donde  $DVE$  son dimensiones explícitas  $\in v$ .  
Sea  $\varphi$  un hecho  $\in v$ . Entonces,  $\langle DI_k At_l \rangle_{k=0}^n$  y  $\langle DE_k At_o \rangle_{k=0}^n$  y  $\langle DVE_k At_p \rangle_{k=0}^n \rightarrow \rightarrow \varphi$  ó es  $\emptyset$ .

**Regla 5.5:** Una aserción valida un conjunto de hechos a través de una expresión.

Sea  $X$  una expresión regular. Sea  $\sigma$  una expresión, donde  $\sigma \in$  tipo de dato Boolean. La operación de  $X, \Theta(X) \rightarrow \sigma$ , y  $\sigma$  es un resultado determinista.  $\sigma$  es true o false.

**Definición 5.9:** El parámetro *fallbackValue* (en el XBRLDM) o atributo (en el MDM) define un valor por defecto para un hecho o un conjunto de hechos, salvo que el valor de este atributo sea nulo.

A través de los filtros implícitos y explícitos, una variable define un hecho o un conjunto de hechos. Si un hecho o conjunto de hechos es nulo, entonces tiene el valor por defecto del parámetro *fallbackValue*, para la variable dada si *fallbackValue* no es nulo. Un ejemplo de esto se muestra más adelante. Si el hecho o conjunto de hechos obtenidos desde una variable es nulo y el atributo *fallbackValue* también es nulo, entonces la aserción no es validada.

De estas definiciones, se pueden inferir las siguientes reglas:

**Regla 5.6:** Si el hecho obtenido por la variable es nulo y el atributo *fallbackValue* existe entonces el hecho toma el valor de *fallbackValue*.

Sea  $v$  una variable. Sea  $\varphi_h$  un conjunto de hechos  $\in v, \forall h = 1, \dots, n$ , y  $fallbackValue \neq null, fallbackValue \in v$ , y  $\varphi_h = null$ , entonces  $\varphi_h ::= fallbackValue$ .

**Regla 5.7:** Una aserción valida un conjunto de hechos a través de una expresión. Una expresión consiste en un conjunto de variables, las cuales ejecutan operaciones entre ellas. El resultado de la aserción es boolean, que es o verdadero o falso.

Sea  $\delta$  una aserción y  $X$  una expresión regular. Sea  $\sigma$  una expresión, donde  $\sigma \in$  tipo de dato Boolean. Sea  $v_i, \forall i = 1, \dots, n$ , un conjunto de variables, donde  $v_i \in \delta$ . donde  $X = (v_1, \dots, v_n)$ . La operación de  $X, \theta(X) \rightarrow \sigma$ .  $\sigma$  es true o false.

**Regla 5.8:** Una aserción valida a un conjunto de hechos a través de una expresión, una expresión consiste en un conjunto de variables, filtros (por ejemplo restricciones de dimensión) y constantes que ejecutan operaciones entre ellas y determinan un conjunto de hechos. Si los hechos no existen, o un hecho en la expresión es nulo, la aserción no es evaluada.

Sea  $\delta$  una aserción. Sea  $X$  una expresión regular, y  $X \in \delta$ .  $X$  es la expresión a ser validada.

Sea  $v_j$  un conjunto de variables, donde  $v_j \in \delta$  y  $\forall j = 1, \dots, m$ .

Sea  $\varphi_h$  un conjunto de hechos  $\in v_h, \forall h = 1, \dots, n$ . Entonces,  $\delta$  es true si  $(\theta(v_1(\varphi_h), \dots, v_m(\varphi_h)))_{h=1}^n \rightarrow True$ . Si no,  $\delta$  es false donde  $X = (\theta(v_1(\varphi_h), \dots, v_m(\varphi_h)))_{h=1}^n \forall \varphi_h$ , donde  $h = 1, \dots, n$ . Pero si  $\varphi_h \rightarrow \emptyset$  entonces  $\theta(X)$  no es evaluado.

**Definición 5.10:** El parámetro Precondition o atributo de la aserción define los hechos que pueden ser evaluados.

Si un hecho o conjunto de hechos obtenidos en una variable o en una aserción no cumple la precondition ésta no es evaluada.

De esta definición se pueden inferir las siguientes reglas:

**Regla 5.9:** Una precondition de variable es una expresión que evalúa los hechos obtenidos por una variable. Si el hecho cumple la precondition el hecho es evaluado en la aserción, en caso contrario no es evaluado. En el último caso es evaluado pero sin los hechos que no cumplen la precondition.

Sea  $\delta$  una aserción y sea  $v$  una variable, donde  $v \in \delta$ .

Sea  $Y$  una expresión regular. Sea  $\varphi$  un hecho, donde  $\varphi \in v$ .

Sea  $Z$  una precondition, donde  $Z \in Y$ .

Si  $\varphi$  resuelve  $Y$ , entonces  $\varphi$  es evaluado. Por otra parte,  $\varphi$  es ignorado.

Esto se puede explicar a través de la lógica proposicional:

Sea la preconditionRule  $(\varphi, Y, \Gamma)$  una regla, donde  $\Gamma$  es el resultado de evaluar  $\varphi$  en  $Y$ . Y sea  $evaluable(\Gamma) \begin{cases} 1 & \text{si } \Gamma \text{ es true.} \\ 0 & \text{si } \Gamma \text{ es false.} \end{cases}$  Entonces, la regla semántica

preconditionRule es definido como  $\forall \varphi, Y, \Gamma (\Gamma, preconditionRule(\varphi, Y, \Gamma) \rightarrow True(\Gamma))$ .

**Regla 5.10:** Una precondition en una aserción es una expresión que evalúa los hechos de la aserción. Si el hecho cumple la precondition, el hecho es evaluado. En

otro caso, no lo es. En este caso la aserción es evaluada pero sin los hechos que no cumplen la precondición.

Sea  $\delta$  una aserción. Sea  $Y$  una expresión regular. Sea  $\varphi$  un hecho  $\in \delta$ . Sea  $Z$  una precondición, donde  $Z \in Y$ , y  $Z \in \delta$ .  
Si  $\varphi$  cumple  $Y$ , entonces  $\varphi$  es evaluado. En otro caso,  $\varphi$  es ignorado.

Esta regla se puede explicar con la lógica proposicional:

Sea la *preconditionRule*  $(\varphi, Y, \Gamma)$  una regla, donde  $\Gamma$  es el resultado de evaluar  $\varphi$  en  $Y$ . Y sea  $evaluable(\Gamma) \begin{cases} 1 & \text{si } \Gamma \text{ es true.} \\ 0 & \text{si } \Gamma \text{ es false.} \end{cases}$  Entonces, la regla semántica *preconditionRule* se define como  $\forall \varphi, Y, \Gamma (\Gamma, preconditionRule(\varphi, Y, \Gamma) \rightarrow True(\Gamma))$ .

**Definición 5.11:** Un *filter group* define un conjunto de filtros explícitos para el cual todas las aserciones están circunscritas.

El conjunto de filtros es un conjunto de pares *dimensión-atributo de dimensión* donde cada par es comparado independientemente con todas las variables. Por ejemplo, para un grupo de filtro explícito  $(D1, A1), (D1, A3)$  donde  $D1$  es una dimensión y  $A1, A3$  son atributos de dimensión, suponiendo una variable con un filtro explícito, en este caso un concepto básico (*Primary ítems, C1*), entonces los hechos a ser evaluados son:

- $(D1, A1)$  y  $C1 \rightarrow Fact1$ .
- $(D1, A3)$  y  $C1 \rightarrow Fact2$ .

Mientras  $((D1, A1), (D1, A3), C1) \rightarrow Fact3$  no es evaluado.

En este punto se puede deducir la siguiente regla:

**Regla 5.11:** Un conjunto de pares *dimensión-atributo de dimensión* en una aserción se puede definir como un *filter group*.

Sea  $\delta$  una aserción. Sea  $\langle DE_k At_l \rangle_{k=0}^n$  un grupo de dimensiones,  $\forall l = 1, \dots, m$ , donde  $DE$  son dimensiones explícitas  $\in \delta$ . Entonces,  $\langle DE_k At_l \rangle_{k=1}^n$  es un filter group (FG). Sea  $\varphi_{q=1}^n$  un conjunto de hechos  $\in v_i \in \delta$ . Entonces,

$$\left( (DE_k At_l) \cup (DI_h At_o) \cup (DV_v At_p) \right)_{k=1}^n \rightarrow \varphi_q, \quad \forall k = 0, \dots, n, \forall l = 0, \dots, m, \forall h = 0, \dots, n, \quad \forall o = 0, \dots, n, \forall v = 0, \dots, n, \forall p = 0, \dots, n.$$

La *Tabla 5.1* muestra el mapeo desde las fórmulas del XBRLDM al MDM, es decir, el mapeo desde el CIM al PIM. Donde la columna 1 muestra el número de la definición, la columna 2 muestra la definición en el XBRLDM, y la columna 3 el mapeo al MDM o modelo en estrella UML.

**Tabla 5.1 Desde el CIM al PIM**

Def	XBRLDM, el CIM	MDM o modelo en estrella (el PIM)
1	Hecho	Hecho
2	Assertion	Restricción
3	Filtro	Conjunto de pares <dimensión/atributo de dimensión>
4	Filtro Implícito	Conjunto de pares <dimensión/atributo de dimensión>
5	Filtro Explícito	Conjunto de pares <dimensión/atributo de dimensión>
6	Concepto Básico	Par <dimensión/atributo de dimensión>
7	Constante	Restricción
8	Variable	Conjunto de pares <dimensión/atributo de dimensión>
9	FallbackValue	Restricción
10	Precondition	Restricción
11	Filter group	Conjunto de pares <dimensión/atributo de dimensión>

Las precedentes definiciones y reglas se explican con el siguiente ejemplo usando la hoja 6610 del balance del sector público consolidado del Banco de España del 2008 (*Banco de España, 2011*). El ejemplo está orientado a clarificar y simplificar la especificación, enfocando al MDM. La fuente del ejemplo es código XML. A través de la plataforma de código abierto XBRL de Arelle (*Arelle, 2015*) la aserción de este ejemplo se obtiene de una manera fácil. Como uno de los objetivos de esta tesis es clarificar la especificación de fórmulas XBRL, seguidamente se analiza como ejemplo una aserción de un informe completo (*Figura 5.6*). La aserción se muestra gráficamente en la *Figura 5.7*.

<i>assertionSet</i>	<i>assertionSet</i>
<i>valueAssertion</i>	<i>val_6610-sh-2</i> <i>(((\$CreditInstitutions + \$InsuranceEntities + \$OtherEntities + \$settlementAdjustments) eq \$TotalGroupEntities) and (\$TotalGroupEntities eq 0))</i>
<i>conceptName</i>	<i>ifrs-gp:InvestmentsInSubsidiariesAtCost</i>
<i>factVariable</i>	<i>\$creditInstitutions fallbackValue =0</i>
<i>explicitDimension</i>	<i>CreditInstitutionConsolidatedGroup</i> <i>sp-bs-d-FR-dist:DistributionDimension</i>
<i>factVariable</i>	<i>\$InsuranceEntities fallbackValue =0</i>
<i>explicitDimension</i>	<i>InsuranceEntities</i> <i>sp-bs-d-FR-dist:DistributionDimension</i>
<i>factVariable</i>	<i>\$otherEntities fallbackValue =0</i>
<i>explicitDimension</i>	<i>otherEntities</i> <i>sp-bs-d-FR-dist:DistributionDimension</i>
<i>factVariable</i>	<i>\$settlementAdjustments fallbackValue =0</i>
<i>explicitDimension</i>	<i>settlementAdjustments</i> <i>sp-bs-d-FR-dist:DistributionDimension</i>
<i>factVariable</i>	<i>\$TotalGroupEntities</i>
<i>explicitDimension</i>	<i>TotalSectorial</i> <i>sp-bs-d-FR-dist:DistributionDimension</i>

**Figura 5.6 Ejemplo de la aserción *val\_6610sh-2***

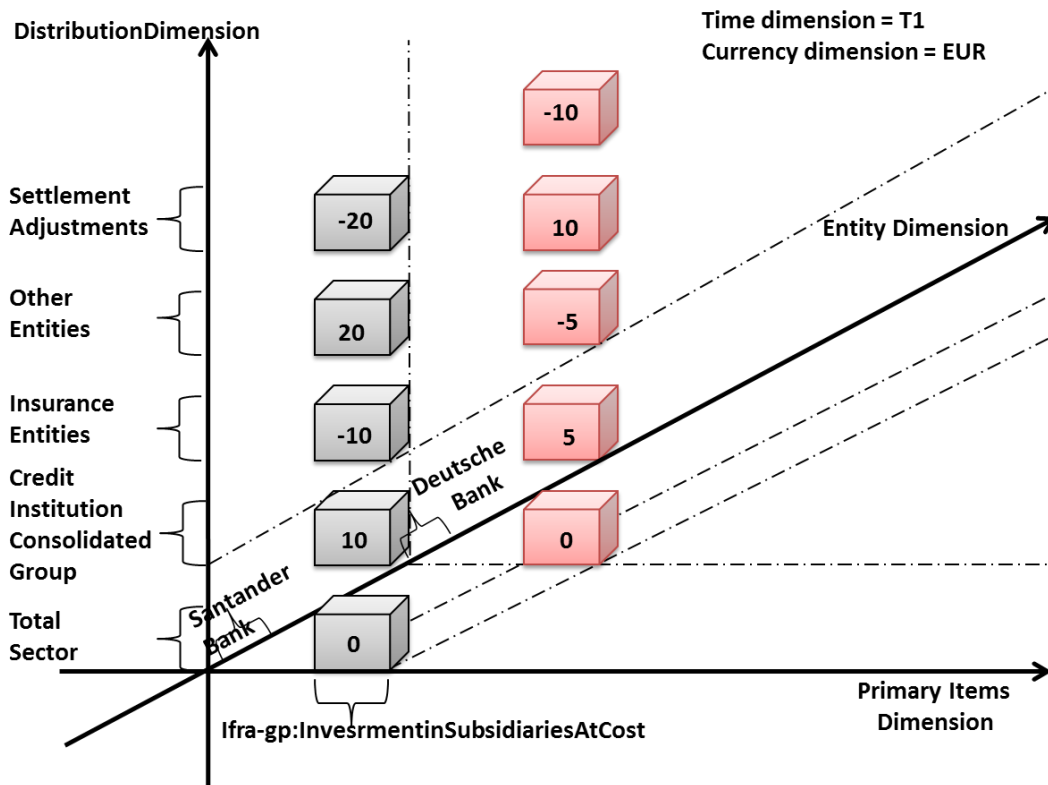


Figura 5.7 Ejemplo de la aserción *val\_6610sh-2* presentada gráficamente

Un conjunto de aserciones empieza con la etiqueta *assertionSet*. Normalmente, este grupo se relaciona a un documento instancia XBRL, en este caso el informe 6610; sin embargo, la asociación es sólo lógica, ya que no hay manera de identificar el conjunto con un documento instancia XBRL.

La aserción empieza con su nombre, aquí *val\_6610sh-2* y entonces la expresión a validar (ver Figura 5.6) es:

$((($CreditInstitutions + $InsuranceEntities + $OtherEntities + $SettlementAdjustments) eq $TotalGroupEntities) and ($TotalGroupEntities eq 0))$

La expresión de arriba consta de cinco variables, llamadas: *\$CreditInstitutions*, *\$InsuranceEntities*, *\$OtherEntities*, *\$SettlementAdjustments* y *\$TotalGroupEntities*. Esta expresión significa que, por una parte, la suma de *\$CreditInstitutions*, *\$InsuranceEntities*, *\$OtherEntities* y *\$SettlementAdjustments* debe ser igual a *\$TotalGroupEntities* y que, por otra parte, *\$TotalGroupEntities* debe ser igual a cero.

Seguido de las variables aparece *conceptName ifrs-gp:InvestmentsInSubsidiariesAtCost*, que es un filtro explícito. Con la etiqueta *conceptName*, la dimensión puede ser identificada como un concepto básico, en el que el atributo de dimensión es *ifrs-gp:InvestmentsInSubsidiariesAtCost* y el par dimensión-atributo de dimensión: *basic concepts, ifrs-gp:InvestmentsInSubsidiariesAtCost*. El concepto básico, *ifrs-gp* denota la taxonomía de propósito general de IFRS (*IFRS General Purpose taxonomy*), mientras *InvestmentsInSubsidiariesAtCost* denota el concepto básico.

Después de la definición de la aserción aparecen las variables:

```
factVariable $creditInstitutionsfallbackValue =0
  explicitDimension CreditInstitutionConsolidatedGroup
sp-bs-d-FR-dist:DistributionDimension
```

En el código de arriba, *factVariable* es la etiqueta, mientras que la variable, cuyo nombre empieza con \$, es *\$creditInstitutions*. El atributo *fallbackValue* se define también y además como *fallbackValue = 0*, a todos los hechos determinados por la variable que sean nulos (por ejemplo los hechos que no existen) se les dará el valor de *fallbackValue*, que en este caso es cero. Seguido de esto, aparece el filtro explícito (par dimensión-atributo de dimensión), el cual en este caso es: *sp-bs-d-FR-dist: DistributionDimension, CreditInstitutionConsolidatedGroup*. En la dimensión, *sp* denota España, *bs* el Banco de España, *d* que es una dimensión, *FR* significa que pertenece a la taxonomía de FINREP, y *dist* es la distribución. Así que *sp-bs-d-FR-dist* es el nombre lógico del concepto referenciado por la taxonomía. En este informe, hay tres filtros implícitos, formalmente llamados: *Period of time, Entity* y *Currency*. Estos filtros implícitos no se muestran en la especificación de la fórmula.

## 5.5. Cuestiones semánticas en el PIM de la especificación de fórmulas XBRL

Un problema particular se encuentra cuando el documento instancia utiliza filtros implícitos y un hecho es nulo. Por ejemplo, sean las variables *\$A*, *\$B*, y *\$C*, con la regla de validación  $A=B+C$ , teniendo en cuenta que cada variable tiene un filtro explícito diferente, y cuyo concepto básico es *C1*, *C2*, Y *C3*, respectivamente. Con

respecto a los filtros implícitos (dimensiones): *Entity* y *TimePeriod*, se puede suponer que tenemos los atributos de dimensión *E1* y *E2*, mientras el último sólo tiene un atributo de dimensión *T1*, sea el siguiente ejemplo en el que se tienen los siguientes hechos *FA1*, *FA2*, *FB1*, *FB2*, *FC1* y *FC2*, donde:

$$1 \quad C1 \cup E1 \cup T1 \rightarrow FA1$$

$$2 \quad C1 \cup E2 \cup T1 \rightarrow FA2$$

$$3 \quad C2 \cup E1 \cup T1 \rightarrow FB1$$

$$4 \quad C2 \cup E2 \cup T1 \rightarrow FB2$$

$$5 \quad C3 \cup E1 \cup T1 \rightarrow FC1$$

$$6 \quad C3 \cup E2 \cup T1 \rightarrow FC2$$

Seguidamente se analiza cuando, por ejemplo, la variable \$A no tiene el atributo *fallbackValue* y el hecho *FA2* también es nulo (Figura 5.8).

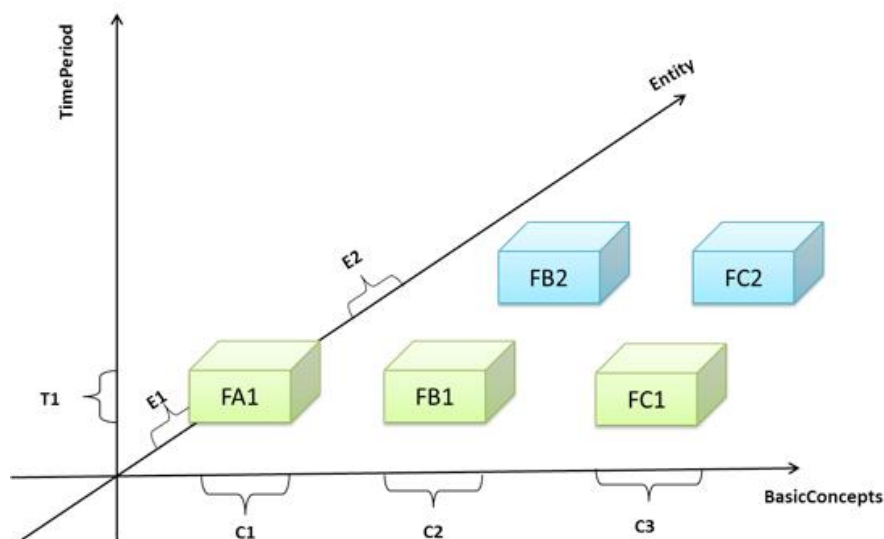


Figura 5.8 Ejemplo con un hecho nulo

El procesador XBRL, o el DBMS tiene que validar:

$$1 \quad FA1 = FB1 + FC1$$

$$2 \quad FA2 = FB2 + FC2$$

Ya que  $FA2$  no está de acuerdo a la *Regla 5.8*, como se muestra arriba y la aserción  $FA2 = FB2 + FC2$ , no está validada.

Otro caso potencialmente problemático es cuando el documento instancia utiliza filtros implícitos y una variable tiene dos hechos, ya que contiene una dimensión adicional (es decir un filtro explícito mas) comparado a las otras dos variables. Por ejemplo, sean las variables  $\$A$ ,  $\$B$ , y  $\$C$ , con la regla de validación  $\$A = \$B + \$C$  y cada variable teniendo un filtro explícito diferente, cuyo concepto básico respectivamente es  $C1$ ,  $C2$ , y  $C3$ . En relación a los filtros implícitos (dimensiones) *Entity* y *TimePeriod*, se puede suponer que poseen los atributos de dimensión  $E1$  y  $E2$ , mientras el último posea solo uno,  $T1$ . Sea, por ejemplo los siguientes hechos  $FA1$ ,  $FA2$ ,  $FB1$ ,  $FB2$ ,  $FC1$ , y  $FC2$ , donde:

- 1  $C1 \cup E1 \cup T1 \rightarrow FA1$
- 2  $C1 \cup E2 \cup T1 \rightarrow FA2$
- 3  $C2 \cup E1 \cup T1 \rightarrow FB1$
- 4  $C2 \cup E2 \cup T1 \rightarrow FB2$
- 5  $C3 \cup E1 \cup T1 \rightarrow FC1$
- 6  $C3 \cup E2 \cup T1 \rightarrow FC2$

El procesador XBRL o el DBMS tiene que validar:

- 1  $FA1 = FB1 + FC1$
- 2  $FA2 = FB2 + FC2$

Si otra dimensión está envuelta en un hecho, por ejemplo la dimensión  $D1$  con atributos  $A1, A2, A3, \dots$ , entonces, si  $C1 \cup E2 \cup T1 \cup (D1,A1) \rightarrow FA2$  y  $C1 \cup E2 \cup T1 \cup (D1,A2) \rightarrow FA2'$ . Sin embargo, la nueva dimensión no está definida en la aserción. En tal caso es posible que ocurra:  $C1 \cup E2 \cup T1 \rightarrow FA2$ . Así que es posible generar los contextos con el mismo concepto básico y dos filtros diferentes por cada uno. En la *Figura 5.9* se muestra un ejemplo.

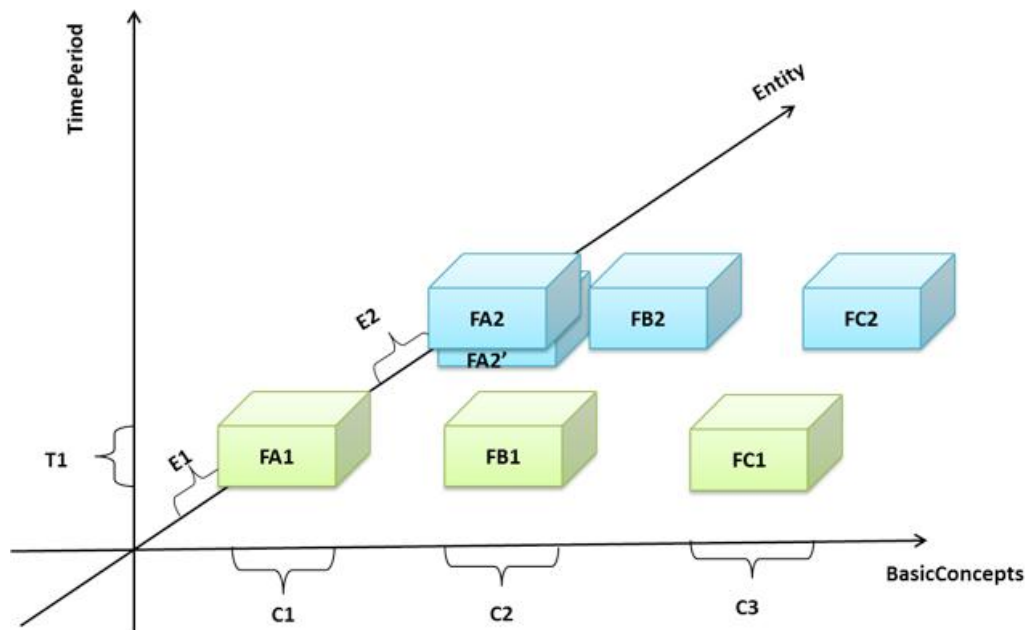


Figura 5.9 Hechos multivaluados

Si  $C1 \cup E2 \cup T1$  es  $FA2$  y  $FA2'$ , entonces es necesario decidir entre:

- 1  $FA2 = FB2 + FC2$  y  $FA2' = FB2 + FC2$
- 2  $FA2 = FB2 + FC2$  o  $FA2' = FB2 + FC2$
- 3 Nada

Mientras el estándar no especifique nada acerca de la posible elección, alguien tiene que elegir. Sin embargo, el problema es raro, ya que los diseñadores de la instancia o fórmula de la taxonomía tienden a evitarlo, aunque no en una forma automática, como por ejemplo con una herramienta CASE (*Computer-Aided Software Engineering*).

## 5.6. Implementación y prueba de concepto (POC), en el PSM

Esta sección presenta la implementación de la propuesta de esta tesis, la cual mapea y valida documentos e instancias XBRL, de una forma automática a través de un RDBMS, y constituye una contribución nueva a este campo. En orden a validar la

propuesta, la POC se ha presentado en conferencias específicas (*Santos, y Castro, 2011b, 2012a*).

En una presentación anterior del autor de la tesis, mostró el mapeo entre el XBRLDM al RDBMS y viceversa. Esta transformación y automatización provee su utilidad por dos razones principales:

- 1 Los expertos de XBRL pudieron validar los estudios realizados por los autores (*Santos, y Castro, 2010, 2011a, y 2011c*).
- 2 El mapeo y la automatización podrían ser usados como una herramienta de código abierto por las empresas y reguladores con recursos limitados, un hecho pertinente a la luz de la crisis financiera.

En esta sección se presenta la arquitectura de validación de un documento instancia XBRL en un DBMS, y un proceso para la validación automatizada en una base de datos de documentos instancias en la especificación XBRL de fórmulas. Mientras en las presentaciones mostradas en el congreso por el autor de la tesis, las cuales ha generado algún interés entre los reguladores en España, se ofreció una POC, para la validación de documentos instancia XBRL en un DBMS, aunque sólo se mostraron códigos, gráficos y demos (*Santos, y Castro, 2012a*). El interés de los reguladores en la POC, incluso en la versión más limitada presentada, fue debida en parte al hecho de que el repositorio generado de un documento instancia XBRL es siempre una base de datos, algo que ellos reconocieron como un importante desarrollo, dada la falta de rendimiento de muchas instancias XBRL con grandes volúmenes de datos.

La arquitectura del mapeo de la especificación de fórmulas XBRL a un RDBMS se divide en dos etapas. En la primera, se genera el código necesario para crear un API de validación. En la segunda etapa, se ejecuta esta API para validar los hechos del informe o documento instancia XBRL en una base de datos.

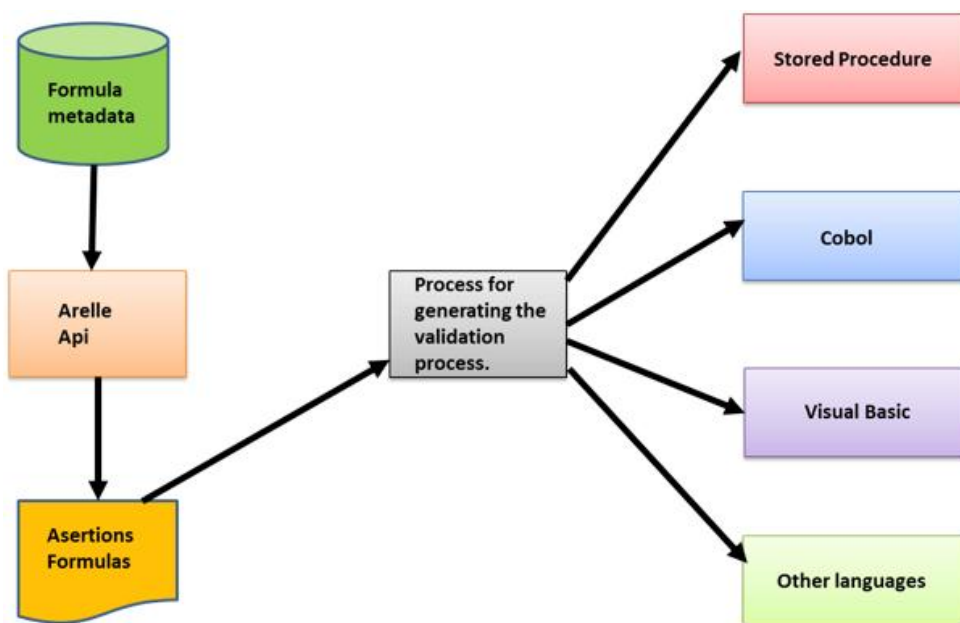


Figura 5.10 Obtención del API de validación

En la presentación de la *Figura 5.1* la primera etapa empieza en el paso 3, cuando se crea la estructura de metadatos tales como dimensiones y atributos de dimensión. Sin embargo, es importante mencionar que para el proceso de generación del API de validación no es necesario tener previamente cargados los metadatos y los hechos en la base de datos. Además, se puede generar el API en cualquier momento anterior a la validación de un documento instancia de la base de datos, después de haber creado la estructura de metadatos XBRL. La *Figura 5.10* muestra esta primera etapa. Primeramente, se obtienen las fórmulas y aserciones a través de *Arelle*, ya que se obtiene en un formato más simple. El API de *Arelle* convierte las fórmulas XBRL (originalmente en formato XML) en un texto plano. Seguidamente la nueva API propuesta por el autor de esta tesis genera otra API para la creación del proceso de validación. Este proceso de validación puede estar implementado en procedimientos almacenados en base de datos, o puede ser generado en otros lenguajes, como COBOL, Visual Basic (VB) y C++, que trabajan con base de datos. Este estudio toma la primera aproximación. Una vez que se crea el API de validación y se cargan en base de datos los metadatos y hechos, se ejecuta el API de validación (ver *Figura 5.11*). Este proceso de validación se puede repetir tantas veces como se desee.

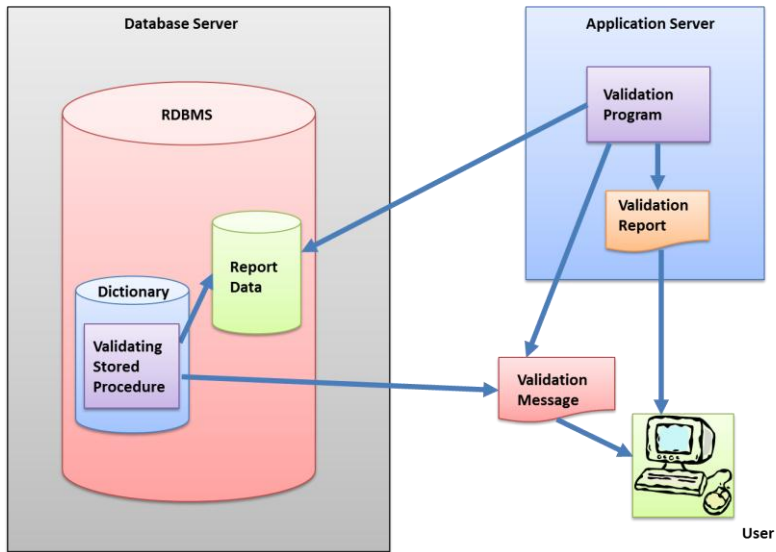


Figura 5.11 Proceso de validación

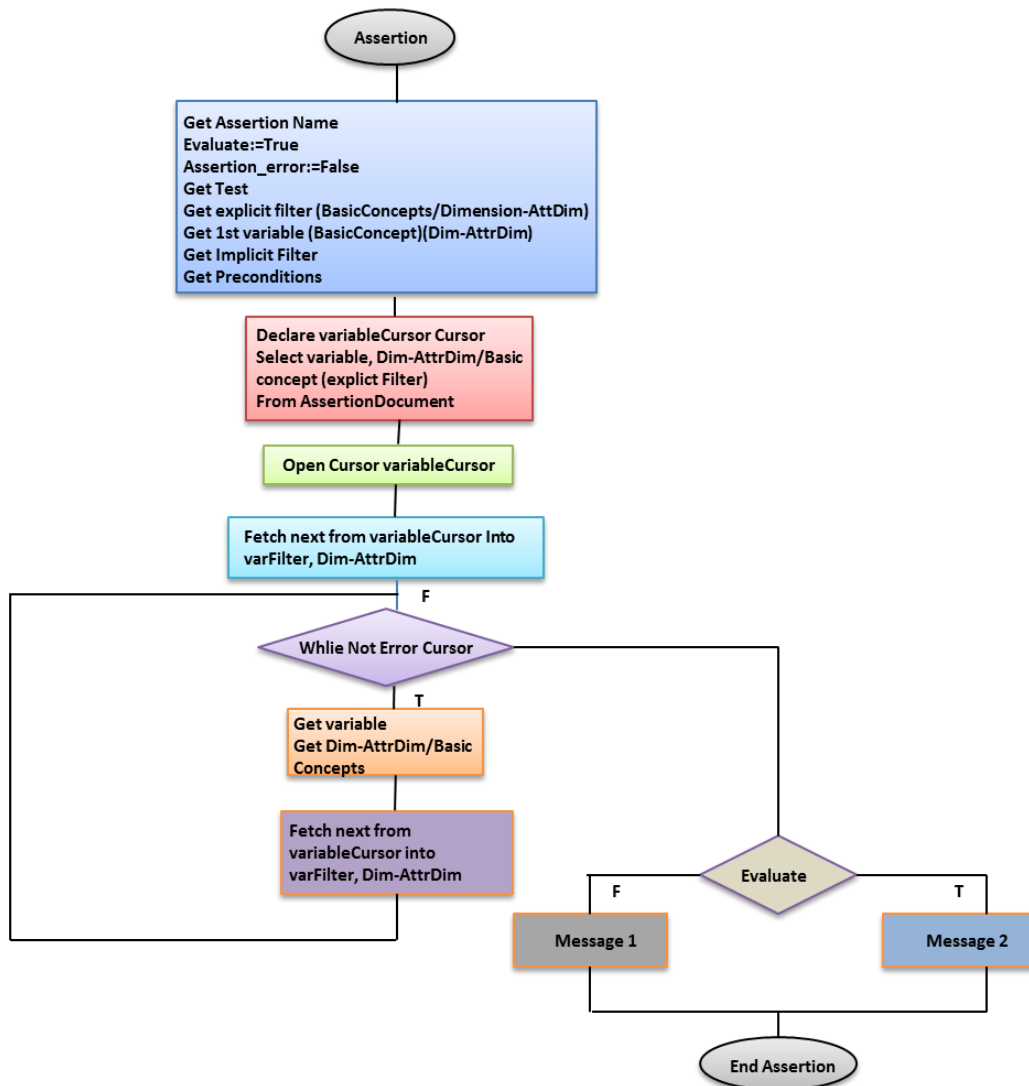


Figura 5.12 Proceso de asección del algoritmo de la generación del proceso de validación

Seguidamente se explica brevemente el proceso de generación del API de validación. Primeramente, se leen las fórmulas XML, hasta que se alcanza el final de fichero (EOF). Cada aserción es entonces procesada por el algoritmo de aserción. El algoritmo (CIM  $\rightarrow$  PIM) anota el nombre de la aserción, el test a validar y las precondiciones, si las tiene (ver *Figura 5.12*). El algoritmo entonces obtiene las dimensiones implícitas a procesar. Después se leen las variables y se va a los procesos de estas variables. Finalmente, el algoritmo tiene que verificar la validación, si la aserción puede ser evaluada o no y se obtendrá un mensaje. Todo este proceso genera unas plantillas con diferentes códigos SQL en un DBMS. En la *Figura 5.12* la evaluación implica el algoritmo de creación de las plantillas SQL (PIM  $\rightarrow$  PSM), y dependiendo si hay precondiciones, variables, etc. En total encontraremos siete plantillas, como se muestra en el *Algoritmo 5.1*.

```
Start
  begin
    if hay precondiciones
      then if si hay variables
        then if filtro explícito tiene conceptos básicos y no dimensiones
          then plantilla SQL V
        else if filtro explícito no tiene conceptos básicos y tiene dimensiones
          then plantilla SQL VI
        else plantilla SQL VII
      else plantilla SQL IV
    else if filtro explícito tiene conceptos básicos y no dimensiones
      then plantilla SQL I
    else if filtro explícito no tiene conceptos básicos y tiene dimensiones
      then plantilla SQL II
    else plantilla SQL III
  end
End
```

#### **Algoritmo 5.1 Creación de plantillas SQL**

Estas plantillas obtenidas en el *Algoritmo 5.1* son un conjunto de procedimientos almacenados (si la validación es implementada SQL), que dependiendo de si hay precondiciones, variables, etc., en total habrán siete plantillas. Este es el paso del

PIM → PSM del paradigma MDA. El código puede ser analizado en las diferentes pruebas de concepto del autor (Santos, y Castro, 2012a; <http://www.openfiling.info/academy/>).

Seguidamente se describe la codificación (Código 5.1) de la plantilla SQL I, mostrada en el Algoritmo 5.1, en el paradigma MDA. La plantilla SQL I se utiliza para validar una aserción sin precondiciones, con un solo concepto. El Código 5.1 está resumido. Este código se corresponde con el ejemplo de la Figura 5.6.

```

CREATE PROCEDURE val_6610_sh_2_1 (val_6610_sh_2_1_Message integer out)
-- First implicit dimension. Entity
DECLARE implicitFilter1 CURSOR FOR
select id_entity from Facts_Table
where [id_dimension_es-be-d-FR-dist_DistributionDimension] like '%es-be-d-FR-
dist:CreditInstitutionConsolidatedGroup%' and id_primaryItems like '%ifrs-gp:InvestmentsInSubsidiariesAtCost%';

open implicitFilter1 fetch next from implicitFilter1 into entity;
while (FETCH_STATUS<>-1)
begin
    -- Second implicit dimension. Period of time.
    DECLARE implicitFilter2 CURSOR FOR
    select timePeriod_start from Facts_Table
    where [id_dimension_es-be-d-FR-dist_DistributionDimension] like '%es-be-d-FR-
dist:CreditInstitutionConsolidatedGroup%' and id_primaryItems like '%ifrs-gp:InvestmentsInSubsidiariesAtCost%'
    and id_entity=entity

open implicitFilter2
    fetch next from implicitFilter2 into timePeriod_start;
    while (FETCH_STATUS<>-1)
    begin
        -----
        -- First variable.
        -- If in the next select more than 1 value is obtained, then it is the second case of semantic questions.
        select CreditInstitutionConsolidatedGroup=value, entity=id_entity from Facts_Table
        where [id_dimension_es-be-d-FR-dist_DistributionDimension] like '%es-be-d-FR-
dist:CreditInstitutionConsolidatedGroup%' and id_primaryItems like '%ifrs-gp:InvestmentsInSubsidiariesAtCost%'
        and timePeriod_start=timePeriod_start
        if CreditInstitutionConsolidatedGroup is null
        begin
            if fallbackvalue is not null CreditInstitutionConsolidatedGroup_N := fallbackvalue
            else
                Begin
            Evaluate:=0; assertion_error:=1
            end
        end
        else
            CreditInstitutionConsolidatedGroup_N := cast(CreditInstitutionConsolidatedGroup as integer);

        -- Second variable.
        -- Third variable.
        -- Fourth variable.
        -- Fifth variable.
        -- Test of validation
        if evaluate =0 -- If 0 no validation
            if assertion_error=1 val_6610_sh_2_1_Message:=0
            else '-- It is necessary to validate.

```

```

        if not((((CreditInstitutionConsolidatedGroup + InsuranceEntities + otherEntities +
settlementAdjustments)=TotalGroupEntities) and (TotalGroupEntities=0))) – The validation
            val_6610_sh_2_1_Message:=0
            fetch next from implicitFilter2 into timePeriod_start;
        end
        close implicitFilter2;
        fetch next from implicitFilter1 into entity;
    end
--
close implicitFilter1;

```

### **Código 5.1 Plantilla I (SQL I), resumen del código**

Cuando se valida un hecho o conjunto de hechos con una aserción es importante el número de hechos envueltos en la aserción, ya que el proceso de ejecución puede llegar a ser incontrolable (*Agrawal et al., 1997; Feng et al., 2010*). Sólo en tiempo de ejecución (durante la validación del informe) se sabrá el número de hechos envueltos en la validación. Este número puede variar enormemente afectando al rendimiento del sistema. Sin embargo, la propuesta desarrollada se implementa en un gestor de base de datos, de tal forma que en general el proceso es abordable.

# CAPÍTULO 6

---

## Validación de la propuesta



## CAPÍTULO 6. Validación de la propuesta

---

### 6.1. Introducción a la validación

Este capítulo está dedicado a la validación de la propuesta y se divide en dos partes. La **primera sección** incluye la validación del modelo de metadatos. En la **segunda sección** se validan las fórmulas definidas en la propuesta para recoger toda la semántica de la taxonomía XBRL.

El principal objetivo de esta validación es asegurar la disponibilidad de esta propuesta para ser utilizada. En ambas secciones la validación es ejecutada en dos fases. En la primera fase se hace para asegurar que la transformación del CIM al PIM se hace correctamente. En esta fase el resultado de la transformación se valida, es decir, se comprueba que el resultado del PIM (o modelo en estrella UML), es correcto. La segunda fase se hace para asegurar que la transformación del PIM al código (o modelo relacional), se ejecuta correctamente. Aunque pareciera redundante utilizar más de un tipo de validación, se considera que de esta manera se incrementa la confianza en los resultados. Este dominio de informes semánticos se asocia a cientos de conceptos, dimensiones, atributos de dimensión, cubos, etc., por lo tanto, la validación del diseño se considera fundamental. De acuerdo a *Gogolla et al. (2007)*, la validación de los modelos conceptuales en las primeras fases de su desarrollo, puede ayudar a corregir el diseño a un punto donde los errores pueden tratarse con relativa facilidad.

### 6.2. Validación del desarrollo de los metadatos de los informes semánticos

Esta sección analiza la validación de la propuesta y está dividida en **dos fases**. En la **primera fase** se obtiene el modelo de estrella UML desde el conjunto de plantillas desarrollado por los usuarios expertos, y en la **segunda fase** se obtiene el modelo relacional utilizando la tecnología ROLAP.

La validación consistirá en probar que los datos obtenidos del desarrollo de la propuesta se emparejan con los requisitos de los usuarios expertos. Las pruebas de

concepto tratadas en este trabajo de investigación se basan en las taxonomías de *FINREP 2012, 2014* y *Solvency II*. *FINREP* es una taxonomía auspiciada por la EBA, y son informes financieros consolidados y subconsolidados basados en IAS/IFRS (*Eurofiling 2012*). *Solvency II* codifica y armoniza la regulación de Seguros de la Unión Europea (UE) y es auspiciada por la EIOPA. Podríamos decir que los usuarios expertos desean obtener unos informes del tipo de la *Figura 4.2*. Sin embargo, es tal el número de conceptos económicos (sólo en *FINREP* se pasó de recoger 4.500 datos en 2012 a 45.000 en 2015 (*Weller, 2015*)), que es imposible trabajar sobre un listado, por lo tanto, los usuarios expertos construyen plantillas en hojas de cálculo, en las que cada celda son los hechos a recoger de las instituciones financieras. De una forma manual, IT analiza cada una de esas celdas junto con los usuarios expertos, y se obtienen los dominios, los conceptos, las jerarquías entre conceptos, las dimensiones, atributos de dimensión, etc. Especialmente en el desarrollo inicial (en el CIM), estas plantillas se encuentran con un gran número de celdas inconexas y con un alto grado de redundancias. En esta primera fase, haciendo uso del *Algoritmo 6.1*, y a partir de los diversos elementos de las plantillas originales (*Figura 4.3*), los datos se introducen en el modelo relacional de la *Figura 4.6*. Aplicando este algoritmo reiteradamente se van analizando las celdas inconexas y redundantes. Esta figura muestra el conjunto de artefactos en UML, los cuales compondrán la estructura inicial en la validación automática del modelo, en el PIM. El *Algoritmo 6.1* utiliza las definiciones y reglas de las secciones 4.3 y 4.4, y verifica las cuestiones semánticas de la sección 4.5. Es realmente este proceso el que hace la validación estructural (*Santos y Nieto, 2014, 2015*), verificando si las jerarquías de los conceptos son válidas en un dominio con respecto a las dimensiones, atributos de dimensión y atributos calculados (si los atributos de dimensión pertenecen a un dominio y no a dos a la vez, etc.).

```

start
  read tipo de datos, dominios, conceptos, conceptos básicos;
  read dimensiones, grupos de dimensiones;
  verificar las jerarquías de los conceptos y las dimensiones;
  obtener atributos de dimensión, atributos calculados;
  obtener cubos permitidos, cubos prohibidos;
  obtener modelo en estrella UML;
  create tables de dimension desde las dimensiones y atributos de dimensión en el
  modelo en estrella;
  create stored procedure con atributos calculados;
  create dimensión base;
  create hechos desde los cubos permitidos;
end

```

### Algoritmo 6.1 Extracción del modelo de metadatos

Seguidamente se obtiene de una forma automática el modelo en estrella UML, como el de la *Figura 4.4*. Para lograr esta transformación en la prueba de concepto, en esta tesis se utiliza *SQL Server Integration Services (SSIS)* una herramienta ETL (Extracción, transformación y carga de datos), que es un producto de *Microsoft*. La segunda fase obtiene el modelo relacional, validándose de una forma semi-automática que la implementación es la correcta, es decir, que el número de dimensiones, atributos de dimensión, atributos calculados, etc., es el correcto. En esta segunda fase se obtiene el código para la tecnología ROLAP.

Tras lo comentado previamente, se va a analizar un poco más en detalle con ejemplos la validación, aunque todos estos son analizados en detalle en *Openfiling (2015)*, incluida su implementación. La primera fase verifica que es correcta la transformación del conjunto de plantillas en el modelo de estrella. Por ello, primeramente se validan las plantillas originales de una forma iterativa, hasta eliminar redundancias. Seguidamente, se prueban estas plantillas descubriendo algunos errores en las plantillas iniciales, como no discernir a que taxonomía se correspondían los conceptos y en las jerarquías de los conceptos (*Santos y Nieto, 2014*). Posteriormente, y para realizar un juego completo de pruebas, se modifican las plantillas con inserciones o modificaciones de elementos incorrectos, como muestra la *Tabla 6.1*, después de lo cual, se verifica si la salida es la esperada. De la misma manera, se prueba la transformación al modelo relacional, se modifica el conjunto de datos con inserciones o modificaciones de elementos incorrectos, como

muestra la *Tabla 6.2*, y después se verifica si la salida es la esperada. El proceso de Santos y Nieto (2014, 2015) se basa en más de 20 módulos, en los que se analiza concepto a concepto, por ejemplo, si la jerarquía de los conceptos en un dominio es correcta. En este ejemplo se cargan los elementos en una tabla con dos hojas, y en cada inserción se ejecuta un disparador para comprobar que no hay ciclos. Cada uno de estos módulos carga cada concepto correspondiente en la estructura UML de la *Figura 4.6*, verificando directamente las restricciones del modelo (elementos repetidos, referencias a otros elementos, etc.). Estos módulos son realmente paquetes de *Integration Services*, y una vez realizada la validación, escribe en un log las validaciones realizadas, correctas e incorrectas. Y finalmente cada módulo ejecuta de una forma automática la creación e inserción de los diversos elementos en la estructura en estrella en el PIM (*Figura 4.4*).

<b>Number of concepts</b>	187																	
<b>Number of domains</b>	17																	
<b>Number of Basic Concepts</b>	16																	
<b>Group of dimension</b>	16	CT	CI	AT	PL	SE	GA	CU	TI	CD	BA	CL	RP	RT	MA	RS	EC	
<b>Number of dimensions</b>	26	5	1	1	1	1	3	1	4	1	1	1	1	1	1	1	2	
<b>Number of concepts by domain</b>	171	44	18	20	23	15	6	3	12	3	2	3	9	8	3	2	0	
<b>Number of dimension attributes</b>	157	38	14	14	22	16	8	2	18	2	2	2	9	6	2	2	0	
<b>Number of calculated attributes</b>	82	24	5	6	16	6	3	1	10	1	1	1	4	2	1	1	0	
<b>Number of implicit dimensions</b>	2																	

**Figura 6.1** FINREP 2012, datos estadísticos en el MDM

Como se ha indicado en esta sección, para este conjunto de pruebas se eligen tres plantillas: FINREP 2012, FINREP 2014, y Solvency II. *Eurofiling* coordinó el desarrollo del FINREP 2012 (*Eurofiling*, 2012), sin embargo, FINREP 2014 y Solvency II fueron coordinados a través de la EBA y EIOPA junto con *Eurofiling*. Seguidamente se analiza la taxonomía de FINREP 2012, que es más sencilla que las otras dos seleccionadas en el juego de pruebas. Esta taxonomía se define en una hoja de cálculo, y consta de once hojas y trece plantillas. La *Figura 6.1* muestra un resumen de los conceptos publicados en *Eurofiling* (2012). En esta versión hay 187 conceptos, 17 dominios, 16 conceptos básicos, 16 grupos de dimensión (tales como *Categories*, CT; *Comprehensive income*, CI; *Amount types*, AT; *Portfolios*, PL; etc.),

157 dimensiones (tales como *Category of assets, AS; Category of assets and liabilities, AL; etc.*), atributos de dimensión y atributos calculados. La columna del grupo de dimensión EC (*Entity codes*) no tiene conceptos, ya que es una dimensión implícita, es decir, sin atributos de dimensión en la definición de la taxonomía. En esta muestra (*FINREP 2012*) no se especifican los hipercubos permitidos, o prohibidos. Además, es necesario crear 82 reglas para los atributos calculados. La siguiente versión de FINREP 2014 es similar a la de FINREP 2012, pero con nuevos requerimientos del Parlamento Europeo (*EBA, 2013*); más información acerca de esta estructura puede ser encontrada en *Santos et al. (2013)* y la *EBA (2014)*. En relación a la taxonomía *Solvency II*, se utilizará un primer prototipo de ésta. Más información se puede encontrar acerca de la estructura en *Santos et al. (2013)* y la actual taxonomía *EIOPA (2014)*. Las pruebas de concepto de la propuesta y la validación para los usuarios expertos se muestran en *Openfiling (2015)* y en *Santos y Castro (2011b, 2012a)* y *Santos y Nieto (2014, 2015)*. En la *Figura 6.2* se muestran las plantillas desarrolladas para la POC (divulgada en el foro *Eurofiling*). En el proceso de validación, se seleccionan los elementos de las plantillas, y se validan por grupos, mostrando en un log los errores e incidencias de la validación.

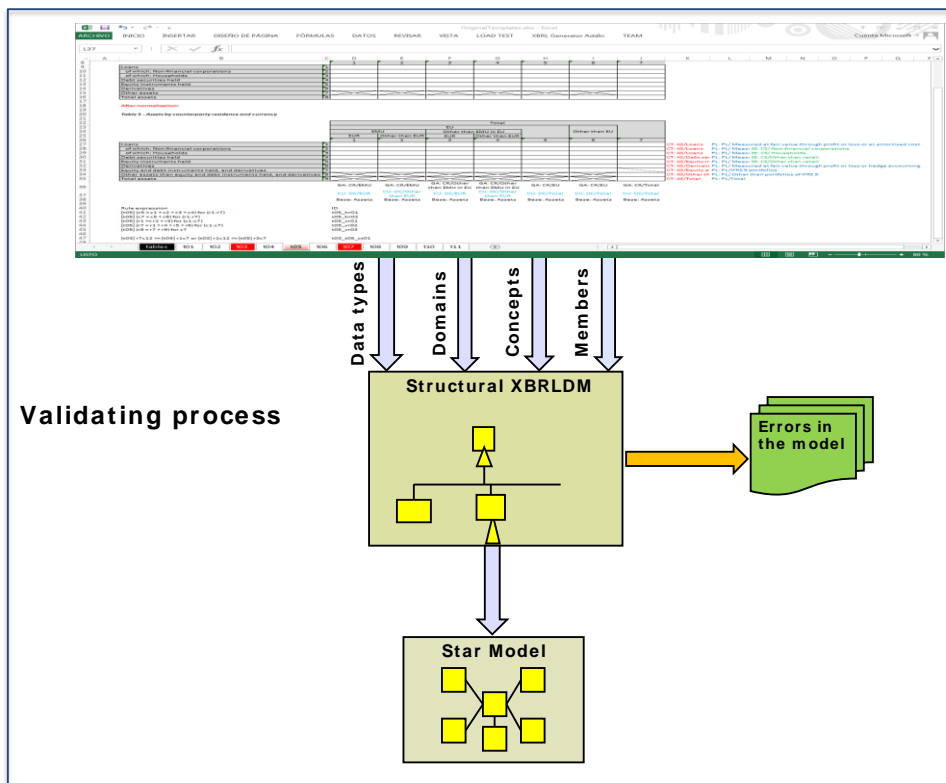
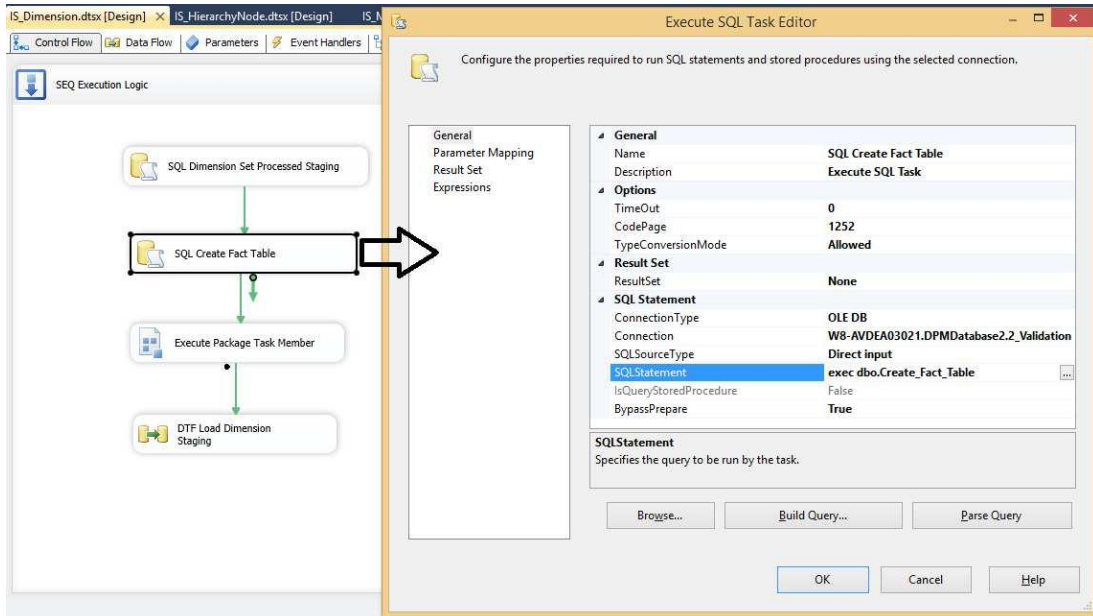


Figura 6.2 Validación de los metadatos

En la *Figura 6.3* se muestra uno de los paquetes de *Integration Services*, que valida las dimensiones, y ejecuta el procedimiento de crear la tabla de hechos.



**Figura 6.3** Flujo de datos IS\_DIMENSION. Ejecución procedimiento SQL de creación de la tabla de Hechos (Nieto, 2015)

El *Código 6.1* muestra un fragmento de la creación de la estructura de las tablas del modelo en estrella (Nieto, 2015).

```

USE [DPMDatabase2.2_Validation]
GO
/***** Object: StoredProcedure [dbo].[Create_Fact_Table] Script Date: 21/09/2015
18:53:42 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author: <Abel Nieto Cano>
-- Create date: 01-05-2015
-- Description: Create Fact Table
-- =====

CREATE PROCEDURE [dbo].[Create_Fact_Table]
AS
BEGIN
/***** Object: Table [FACT].[FactTable] Script Date: 23/05/2015 14:05:50 *****/
DROP TABLE [FACT].[FactTable]
CREATE TABLE [FACT].[FactTable](
    [IdFactTable] [bigint] IDENTITY(1,1) NOT NULL,
    [FactValue] [varchar](255) NULL,
    [ContentType] [int] NOT NULL,
    [ContentUnit] [int] NOT NULL,
    [Precision] [float] NULL,
    [Balance] [int] NOT NULL
CONSTRAINT [PK_FactTable] PRIMARY KEY CLUSTERED
.....
ALTER TABLE [FACT].[FactTable] WITH CHECK ADD CONSTRAINT
[FK_FactTable_Balance] FOREIGN KEY([Balance])
.....
GO

CREATE PROCEDURE [dbo].[sp_CreateTableDimension] @DimensionID [int]
WITH EXECUTE AS CALLER
AS
BEGIN
SET NOCOUNT ON;

DECLARE @sqlcreate NVARCHAR(MAX)
DECLARE @CUENTA AS INT
DECLARE @DimID AS varchar(30)
SET @Dimid = cast (@DimensionID as varchar)
SET @CUENTA = 0
SELECT @CUENTA = count(*) FROM INFORMATION_SCHEMA.TABLES WHERE

```

```

substring(TABLE_NAME,10,7) in (@Dimid) and
TABLE_SCHEMA = 'DIM'
IF @CUENTA = 0
BEGIN
SET @sqlcreate = '
CREATE TABLE [DIM].[Dimension'+@Dimid+'](
    [Dimension'+@Dimid+'ID] [int] NOT NULL,
    [DomainID] [int] NULL,
    [DimensionCode] [nvarchar](max) NULL,
    [DimensionLabel] [nvarchar](max) NULL,
    [DimensionDescription] [nvarchar](max) NULL,
    [IsTyped] [bit] NOT NULL,
    [IsImpliedIfNotExplicitlyModelled] [bit] NOT NULL,
    [DimensionXbrlCode] [nvarchar](255) NULL,
    [ConceptID] [int] NULL,
    CONSTRAINT [PK_Dimension'+@Dimid+'] PRIMARY KEY CLUSTERED
    ([Dimension'+@Dimid+'ID] ASC)) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
ALTER TABLE FACT.FactTable
ADD Dimension'+@Dimid+'ID int NULL

```

**Código 6.1 Creación de las tablas del modelo en estrella (Nieto, 2015)**

Según se validaban los elementos, dominios, dimensiones y atributos de dimensión se crearon dos disparadores. Uno para controlar que no se insertan hojas en el nivel inferior, en aquéllas en las que en su nivel superior posean una hoja diferente (Código 6.2), ya que si se diera el caso, se tendría un ciclo, y no sería un árbol.

El segundo disparador, se utiliza para crear las tablas de las dimensiones e insertar los atributos de dimensión de las dimensiones (Código 6.3).

```

USE [DPMDatabase2.2_Validation]
GO
/***** Object: Trigger [dbo].[HIERARCHY_COMPARISION] Script Date: 15/09/2015
3:43:07 *****/
GO
CREATE TRIGGER [dbo].[HIERARCHY_COMPARISION] ON [dbo].[HierarchyNode]
AFTER INSERT AS
DECLARE @CUENTA AS INT
DECLARE @Hid AS VARCHAR(6)
DECLARE @HNid AS VARCHAR(6)
SET @CUENTA = 0
SELECT @CUENTA = COUNT(*)
FROM inserted WHERE INSERTED.HierarchyID <> INSERTED.ParentHierarchyID
SELECT @Hid = cast(INSERTED.HierarchyID as varchar) , @HNid =
    cast(INSERTED.ParentHierarchyID as varchar)
FROM inserted WHERE INSERTED.HierarchyID <> INSERTED.ParentHierarchyID and
INSERTED.ParentHierarchyID != null

IF @CUENTA <> 0
BEGIN
    --INSERT INTO ERROR (CADENA) VALUES ('ERROR')
    INSERT INTO [STG].[LOG_ERRORS]([CODE_TABLE],[ERROR_DATE]
,[DES_PROCESS],[DES_TASK],[COD_ERROR],[COD_USER])
VALUES(NULL,GETDATE(),'IS_HierarchyNode',null,'-000000001'
,'Description: "Cannot insert the value distinct to HierarchyID = '+ @Hid
+' into column ParentHierarchyID = '+ @HNid + ' , table DPM_Testing_Database
2.0.1.dbo.HierarchyNode; column does not allow distinct values into two columns.
INSERT fails.'
,null)
END
GO

```

**Código 6.2. Busca ciclos en la jerarquía de los conceptos (Niefo, 2015)**

```

USE [DPMDatabase2.2_Validation]
GO
/***** Object: Trigger [dbo].[DIMENSION_CREATE] Script Date: 21/09/2015 18:55:59
*****/
GO
CREATE TRIGGER [dbo].[DIMENSION_CREATE] ON [dbo].[Dimension]
AFTER INSERT AS
DECLARE @SqlInsert NVARCHAR(MAX)
DECLARE @CUENTA AS INT
DECLARE @DimensionID AS int
DECLARE @DomainID AS int
DECLARE @DimensionCode AS nvarchar(max)
DECLARE @DimensionLabel AS nvarchar(max)
DECLARE @DimensionDescription AS nvarchar(max)
DECLARE @IsTyped AS bit
DECLARE @IsImpliedIfNotExplicitlyModelled AS bit
DECLARE @DimensionXbrlCode AS nvarchar(255)
DECLARE @ConceptID AS int
DECLARE @Dimid AS VARCHAR(6)

SET @CUENTA = 0
SELECT @CUENTA = COUNT(*)
FROM inserted
SELECT
@DimensionID = cast(INSERTED.DimensionID as int)
, @DomainID = cast(INSERTED.DomainID as int)
, @DimensionCode = INSERTED.[DimensionCode]
, @DimensionLabel = INSERTED.[DimensionLabel]
, @DimensionDescription = INSERTED.[DimensionDescription]
, @IsTyped = INSERTED.[IsTyped]
, @IsImpliedIfNotExplicitlyModelled = INSERTED.[IsImpliedIfNotExplicitlyModelled]
, @DimensionXbrlCode = INSERTED.[DimensionXbrlCode]
, @ConceptID = cast(INSERTED.[ConceptID] as int)
FROM inserted

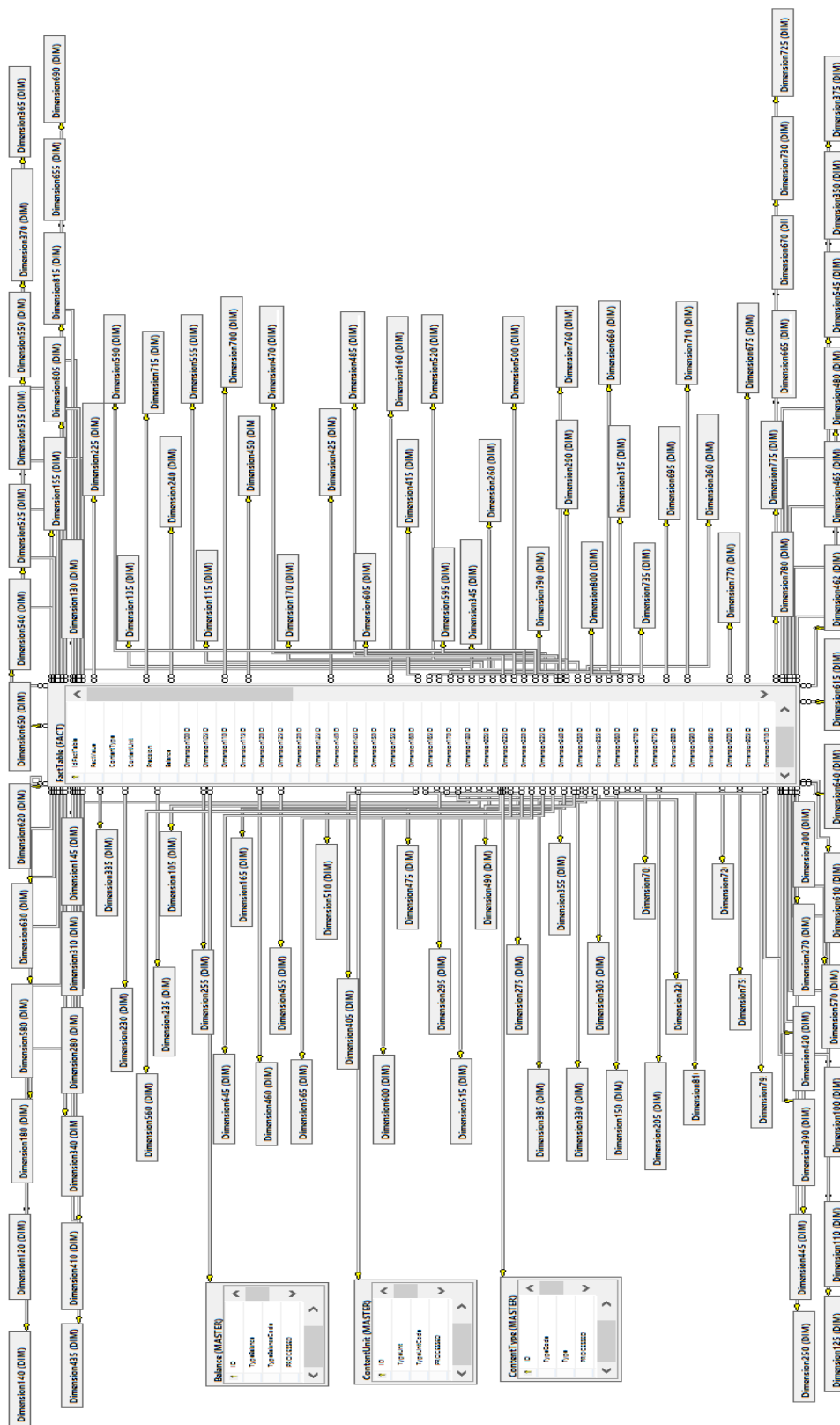
IF @CUENTA <> 0
BEGIN
EXEC [dbo].[sp_CreateTableDimension] @DimensionID = @DimensionID
EXEC [dbo].[sp_InsertTableDimension] @DimensionID = @DimensionID, @DomainID
=@DomainID, @DimensionCode = @DimensionCode,
@DimensionLabel = @DimensionLabel, @DimensionDescription =
@DimensionDescription,
@IsTyped = @IsTyped, @IsImpliedIfNotExplicitlyModelled =
@IsImpliedIfNotExplicitlyModelled, @DimensionXbrlCode = @DimensionXbrlCode,
@ConceptID = @ConceptID
END

GO

```

**Código 6.3. Crea las tablas de dimensiones y carga sus atributos de dimensión (Nieto, 2015)**

En la *Figura 6.4* se muestra el diagrama MDM de FINREP 2014 obtenido al final.



**Figura 6.4.** Diagrama de base de datos en el MDM de FINREP 2014 (Nieto, 2015)

Las *Tablas 6.1* y *6.2* verifican el conjunto de pruebas de validación para esta propuesta. La columna 1 muestra el número de prueba. La columna 2 indica si la validación pertenece al modelo de estrella UML o al modelo relacional (RM). La columna 3 señala el test a validar. Esta columna muestra el caso de test, por ejemplo, en el test 1, *Dos conceptos repetidos* que significa que es imposible repetir dos conceptos. Las columnas 4, 5 y 6 son entradas al test. Estas columnas refieren el conjunto de objetos correctos y el conjunto de objetos incorrectos a probar. Por ejemplo, el test número 1 refiere 187 conceptos, más tres repetidos. Finalmente, la última columna indica la salida del test. En el test número uno, en las tres muestras (FINREP 2012, 2014 y Solvency II) tres conceptos están repetidos y son insertados, respectivamente, así que la salida tendrá tres errores, con tres conceptos repetidos respectivamente.

**Tabla 6.1 Test de validación perteneciente al modelo en estrella UML**

Número de test	Fase	Test a validar	Entrada FINREP 2012	Entrada FINREP 2014	Entrada Solvency II	Salida del Test
1	UML	Dos conceptos repetidos	187 conceptos+ 3 conceptos repetidos	1632 conceptos + 3 conceptos repetidos	145 conceptos + 3 conceptos repetidos	3 conceptos repetidos
2	UML	Dos dominios repetidos	18 dominios + 2 dominios repetidos	35 dominios + 2 dominios repetidos	1 dominio + 2 dominios repetidos	2 dominios repetidos
3	UML	Un concepto en 2 dominios	187 conceptos y 2 conceptos en 2 dominios	1632 conceptos y 2 conceptos en 2 dominios	145 conceptos y 2 conceptos en 2 dominios	2 conceptos en dos dominios
4	UML	Dos dimensiones repetidas	26 dimensiones + 1 dimensión repetida	92 dimensiones + 1 dimensión repetida	2 dimensiones + 1 dimensión repetida	1 dimensión repetida
5	UML	Un grupo de dimensiones repetido	16 grupos de dimensiones + 1 repetida	35 grupos de dimensiones + 1 repetida	1 Un grupo de dimensiones + 1 repetida	1 Grupo de dimensión repetido
6	UML	Un grupo de dimensión en 2 dominios	16 grupos de dimensión y 1 grupo en dos dominios	35 grupos de dimensión y 1 grupo en dos dominios	1 grupos de dimensión y 1 grupo en dos dominios	1 grupo de dimensión en 2 dominios
7	UML	Ciclos de los conceptos de una dimensión	2 dimensiones, y una dimensión con un ciclo	92 dimensiones y una dimensión con un ciclo	2 dimensiones, y una dimensión con un ciclo	1 dimensión con un ciclo
8	UML	Creación de atributos de dimensión	26 dimensiones, 157 atributos de dimensión + 1 atributo de dimensión incorrecto	4 dimensiones, 30 atributos de dimensión + 1 atributo de dimensión incorrecto	2 dimensiones, 2 atributos de dimensión + 1 atributo de dimensión incorrecto	1 atributo de dimensión incorrecto
9	UML	Creación de atributos calculados	26 dimensiones, 82 atributos calculados + 1 atributo de dimensión incorrecto	4 dimensiones, 18 atributos calculados + 1 atributo dimensión incorrecto	2 dimensiones, 2 atributos calculados + 1 atributo de dimensión incorrecto	1 atributo de dimensión incorrecto
10	UML	Los conceptos de una dimensión tienen una sola raíz	26 dimensiones, 171 conceptos. Dos raíces en una dimensión	92 dimensiones, 1632 conceptos. Dos raíces en una dimensión	2 dimensiones, 4 conceptos. Dos raíces en una dimensión	Dos raíces en una dimensión

Número de test	Fase	Test a validar	Entrada FINREP 2012	Entrada FINREP 2014	Entrada Solvency II	Salida del Test
11	UML	Tipo de periodo de tiempo, el mismo en la dimensión y en el concepto básico en un hipercubo	10 hipercubos y uno con diferente período de tiempo	(No hay conceptos básicos, se considera una dimensión más)	2 hipercubos y uno con diferente período de tiempo	1 diferente tipo de periodo de tiempo
12	UML	Cada dimensión implícita está asociada con un tipo de dato	2 dimensiones implícitas, y una sin tipo	5 dimensiones implícitas, y una sin tipo	1 dimensiones implícitas, y una sin tipo	1 sin tipo

**Tabla 6.2 Test de validación perteneciente al modelo relacional (RM)**

Número de test	Fase	Test a validar	Entrada FINREP 2012	Entrada FINREP 2014	Entrada Solvency II	Salida del Test
1	RM	Creación de atributos de dimensión	26 dimensiones, 157 atributos de dimensión + 1 atributo de dimensión incorrecto	4 dimensiones, 30 atributos de dimensión + 1 atributo de dimensión incorrecto	2 dimensiones, 2 atributos de dimensión + 1 atributo de dimensión incorrecto	1 atributo de dimensión incorrecto
2	RM	Creación de atributos calculados	26 dimensiones, 82 atributos calculados + 1 atributo de dimensión incorrecto	4 dimensiones, 18 atributos calculados + 1 atributo de dimensión incorrecto	2 dimensiones, 2 atributos calculados + 1 atributo de dimensión incorrecto	1 atributo calculado incorrecto
3	RM	El mismo tipo de periodo de tiempo en la dimensión y en el concepto básico, en un hipercubo	10 hipercubos y uno con un diferente tipo de periodo de tiempo	(Ningún concepto básico en esta taxonomía)	2 hipercubos y uno con un diferente tipo de periodo de tiempo	1 diferente tipo de periodo de tiempo
4	RM	Cada dimensión implícita está asociada con un tipo de dato	2 dimensiones implícitas, una sin tipo	5 dimensiones implícitas, una sin tipo	1 dimensión implícita, una sin tipo	1 sin tipo

La validación realizada en las POCs en el PIM de cada muestra (*FINREP 2012, 2014, Solvency II*) describe todas las validaciones estructurales en un 95%, quedando fuera, por falta de datos reales, las relaciones de los conceptos con la taxonomías, y en *FINREP 2014 y Solvency II* que no aplican la dimensión base. Con respecto a la validación en el PSM, las diferentes POCs muestran que están validadas en un 75% ya que no se ha probado suficientemente la generación de procedimientos almacenados de los atributos calculados de los atributos de dimensión de una dimensión en un dominio. Esto es debido a que la guía de buenas prácticas de *Eurofiling* recomienda utilizar las fórmulas en vez de los cálculos. Por otra parte, los datos publicados por la EBA y la EIOPA en *FINREP 2014 y Solvency II* también modelan las plantillas y los ejes de los cubos, pero este tipo de datos en el modelado son redundantes y se utilizan a nivel de presentación para los usuarios expertos.

Esta propuesta produce metadatos bien contruidos para los informes económico financieros semánticos, porque es una validación estructural. Sin embargo, sería necesario continuar la validación con los usuarios expertos en orden a que la validación sea semánticamente completa. Para lograr esto, es necesario un estudio económico financiero y contable de los conceptos dominios, jerarquías, etc. y esto se deja para trabajos posteriores.

### **6.3. Validación de los cálculos utilizando la especificación XBRL de fórmulas**

Esta sección se basa en varias pruebas de concepto (POC). La más importante, y con la que comenzaremos, es la mostrada en *15<sup>th</sup> Eurofiling Workshop (Santos y Castro, 2012a; Santos et al., 2016)*. La plataforma de esta POC es un servidor de base de datos, el RDBMS elegido es Microsoft SQL Server 2012 y, el lenguaje de programación es VBScript y T-SQL. Con respecto al sistema operativo, es Windows Server 2008 R2. El cliente es Windows 7.

En la POC se testea el informe 6610, con un conjunto de pruebas, que consisten en 12 aserciones identificadas numéricamente. En primer lugar, se crea la estructura necesaria para esta prueba de concepto, es decir, el conjunto de tablas necesarias. Sin embargo, en esta prueba en vez de utilizar el modelo en estrella de la *Figura 4.4*, se antepone entre las dimensiones y la tabla de hechos una tabla con los contextos,

es decir con todos los pares <dimensión – atributo de dimensión> que pertenecen a un hecho (Figura 6.5).

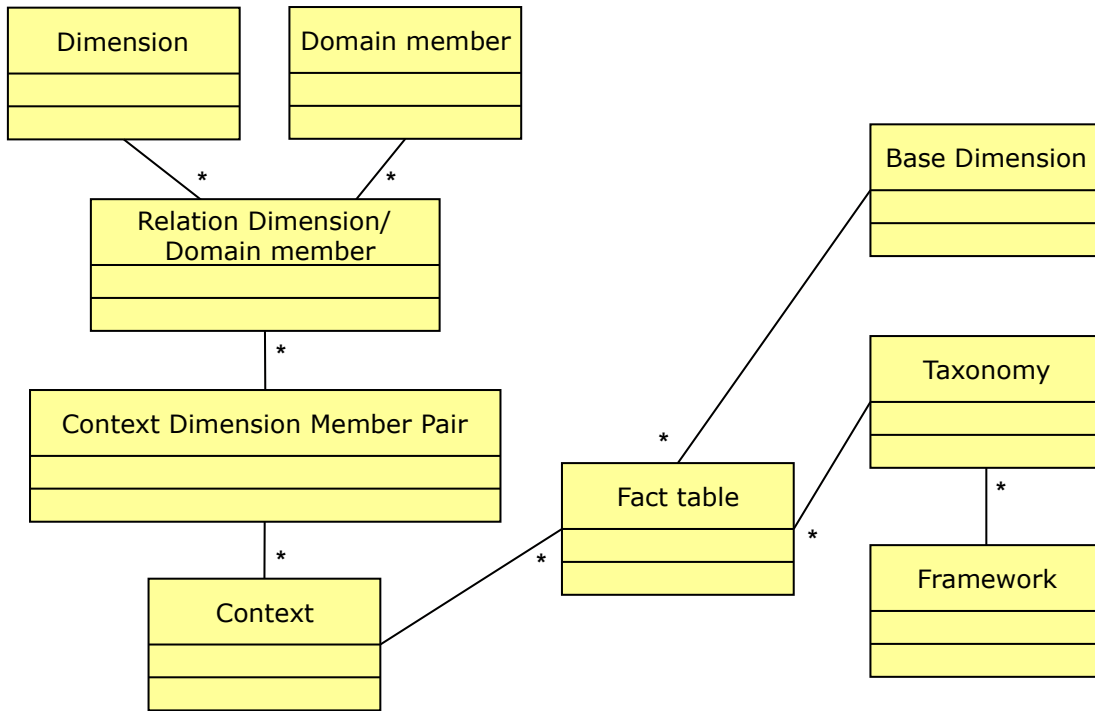


Figura 6.5. Diagrama

Después, como se indica en la sección 5.1, a partir de la *api* de Arelle (2015), herramienta de código abierto, se obtienen todas las dimensiones, atributos de dimensión, conceptos básicos, etc. en un fichero plano, pero con formato del XBRLDM. Desde este fichero plano, y con todos los elementos necesarios, se cargan todos los metadatos en la estructura en estrella (Código 6.4).

```

--
-- xbrls001_create.sql
--
--
-- This script creates tables of the XBRL Data Model.
--
--
-- Ignacio-J. Santos
-- Carlos III University of Madrid
-- July 30th, 2011
--
use TXBRL_Report
go

-----

-- 1. Delete the tables if they exist.
-- 2. create the tables.
-----

IF OBJECT_ID(N'Facts', N'U') IS NOT NULL
DROP TABLE Facts;
go
IF OBJECT_ID(N'Context_Dim_AttrDim', N'U') IS NOT NULL
DROP TABLE Context_Dim_AttrDim;
go
IF OBJECT_ID(N'PrimaryItems', N'U') IS NOT NULL
DROP TABLE PrimaryItems;
go

IF OBJECT_ID(N'Relation_Dim_AttrDim', N'U') IS NOT NULL
DROP TABLE Relation_Dim_AttrDim;
go

IF OBJECT_ID(N'Dimensions', N'U') IS NOT NULL
DROP TABLE Dimensions;
go

IF OBJECT_ID(N'DimAttributes', N'U') IS NOT NULL
DROP TABLE DimAttributes;
go

IF OBJECT_ID(N'Context', N'U') IS NOT NULL
DROP TABLE Context;
go

IF OBJECT_ID(N'Header', N'U') IS NOT NULL

```

```

DROP TABLE Header;
go

IF OBJECT_ID(N'Unit', N'U') IS NOT NULL
DROP TABLE Unit;
go

IF EXISTS (SELECT name FROM sys.indexes
           WHERE name = N'IX_groupDim')
    DROP INDEX IX_groupDim ON Dimensions;
GO
declare @exist_column_ID int;
select @exist_column_ID=0
select @exist_column_ID=count(*)
from sys.sysobjects as a join sys.syscolumns b
    on a.id=b.id
where upper(a.name)='DIMENSIONS' and upper(b.name)=upper('groupDim');
--print '@exist_column_ID....:' + cast(@exist_column_ID as char(2))
if @exist_column_ID=1
    begin
        alter table Dimensions
        drop constraint FK_groupDim
        alter table Dimensions
        drop column groupDim;
    end
go
IF OBJECT_ID(N'GroupDim', N'U') IS NOT NULL
DROP TABLE GroupDim;
go

IF OBJECT_ID(N'dataPoint', N'U') IS NOT NULL
DROP TABLE dataPoint;
go

```

**Código 6.4 SQL script que crea las tablas del modelo en estrella**

Independientemente de la carga de la tabla de hechos, es decir, de los informes, de forma paralela se pasa al proceso de creación de los procedimientos almacenados con las validaciones de las fórmulas. El proceso de validación de las fórmulas, parte de las fórmulas en XBRLDM, que se descargan en un fichero plano (Código 6.5) con la *api* de Arelle (Figura 5.6).

```

Formula object,,,,Label,Cover,Com-ple-ment,Bind as se-quence,Expression
assertionSet,,,,assertionSet,,,,

,valueAssertion,,val_6610-sh-2,,,,((($entidadesCredito + $entidadesSeguros +
$otrasEntidades + $ajustesEliminaciones) eq $entidadesGrupoTotal) and
($entidadesGrupoTotal eq 0))

,,conceptName,,fil_ifrs-gp_InvestmentsInSubsidiariesAtCostd390e11,,,,ifrs-
gp:InvestmentsInSubsidiariesAtCost

,,factVariable
$entidadesCredito,,var_DistribucionDim_GruposolidableEntidadesCredito390e15,,fals
e,fallbackValue =0
,,,explicitDimension,fil_DistribucionDim_GruposolidableEntidadesCredito390e16,true,fa
lse,,es-be-d-FR-dist:DistribucionDimension

,,factVariable
$entidadesSeguros,,var_DistribucionDim_EntidadesSegurosd390e20,,,,false,fallbackValue
=0
,,,explicitDimension,fil_DistribucionDim_EntidadesSegurosd390e21,true,false,,es-be-d-FR-
dist:DistribucionDimension

,,factVariable
$otrasEntidades,,var_DistribucionDim_OtrasEntidadesd390e25,,,,false,fallbackValue =0
,,,explicitDimension,fil_DistribucionDim_OtrasEntidadesd390e26,true,false,,es-be-d-FR-
dist:DistribucionDimension

,,factVariable
$ajustesEliminaciones,,var_DistribucionDim_AjustesEliminacionesd390e30,,,,false,fallbackV
alue =0
,,,explicitDimension,fil_DistribucionDim_AjustesEliminacionesd390e31,true,false,,es-be-d-
FR-dist:DistribucionDimension

,,factVariable $entidadesGrupoTotal,,var_DistribucionDim_TotalSectoriald390e35,,,,false,
,,,explicitDimension,fil_DistribucionDim_TotalSectoriald390e36,true,false,,es-be-d-FR-
dist:DistribucionDimension

```

**Código 6.5. Formula val\_6610-sh-2 obtenida desde Arelle (Figura 5.6)**

Mediante un algoritmo mostrado en la *Figura 5.12*, el *Algoritmo 5.1*, y el *Código 5.1*, se obtiene el conjunto de procedimientos almacenados de validación, uno por aserción (*Figura 5.10*). El *Código 6.6* muestra el procedimiento almacenado de validación de la aserción de la *Figura 5.6*, obtenido de forma automática.

```
-- Carlos III University of Madrid
-- Ignacio Santos, April 2012

-- Open Source for validating XBRL Document Instances (Stored Procedures)

-- Generated at:20/05/2012 : 21:14:38.

Use [TXBRL_Lab6]
go

go
IF OBJECT_ID ( 'val_6610_sh_2_1', 'P' ) IS NOT NULL
    DROP PROCEDURE val_6610_sh_2_1;
go
CREATE PROCEDURE val_6610_sh_2_1 @val_6610_sh_2_1_Message int out
AS

-- Assertion Type I.

declare @evaluar tinyint; -- When a fact is null and fallback is null then @evaliar os 1.
declare @fallbackvalue int;

set @val_6610_sh_2_1_Message=1    -- Assertion correct, 0 is incorrect.
declare @assertion_error tinyint;
declare @entity nvarchar(100);
declare @instantPeriod varchar(20);

declare @entidadesCredito_A varchar(200);
declare @entidadesCredito int;

declare @entidadesSeguros_A varchar(200);
declare @entidadesSeguros int;

declare @otrasEntidades_A varchar(200);
declare @otrasEntidades int;
```

```

declare @ajustesEliminaciones_A varchar(200);
declare @ajustesEliminaciones int;

declare @entidadesGrupoTotal_A varchar(200);
declare @entidadesGrupoTotal int;

DECLARE implicitFilter CURSOR FOR
select b.entityID, b.instantPeriod
from TXBRL_Lab6.dbo.Facts a inner join TXBRL_Lab6.dbo.Context b on a.contextRef=b.ID
     inner join TXBRL_Lab6.dbo.Context_Dim_AttrDim c on a.contextRef=c.contextID
where c.dimension ='es-be-d-FR-dist:DistribucionDimension'
     and c.attrDim like '%GrupoConsolidableEntidadesCredito%'
     and a.baseID='ifrs-gp:InvestmentsInSubsidiariesAtCost'

open implicitFilter

fetch next from implicitFilter
into @entity, @instantPeriod;

while (@@FETCH_STATUS<>-1)
begin
    --select '@entity...:', @entity, '@instantPeriod....:', @instantPeriod;
    set @assertion_error=0
    -----
    set @evaluar=1;
    set @entidadesCredito_A=null;
    set @entidadesCredito=null;
    set @fallbackvalue=0;
    select @entidadesCredito_A=a.Value
        from TXBRL_Lab6.dbo.Facts a inner join TXBRL_Lab6.dbo.Context b on
a.contextRef=b.ID
        inner join TXBRL_Lab6.dbo.Context_Dim_AttrDim c on a.contextRef=c.contextID
        where c.dimension ='es-be-d-FR-dist:DistribucionDimension'
        and c.attrDim like '%GrupoConsolidableEntidadesCredito%'
        and a.baseID='ifrs-gp:InvestmentsInSubsidiariesAtCost'
        and b.entityID=@entity and b.instantPeriod=@instantPeriod
    --select '@entidadesCredito'_A, @entidadesCredito_A
    if @entidadesCredito_A is null
        begin
            if @fallbackvalue is not null
                begin
                    set @entidadesCredito=@fallbackvalue
                    --select '@entidadesCredito', @entidadesCredito

```

```

        end
    else
        begin
            set @evaluar=0
            set @assertion_error=1
        end
    end
else
    begin
        set @entidadesCredito=cast(@entidadesCredito_A as int);
        --select '@entidadesCredito...:', @entidadesCredito
    end

    set @entidadesSeguros_A=null;
    set @entidadesSeguros=null;
    set @fallbackvalue=0;
    select @entidadesSeguros_A=a.Value
        from TXBRL_Lab6.dbo.Facts a inner join TXBRL_Lab6.dbo.Context b on
a.contextRef=b.ID
        inner join TXBRL_Lab6.dbo.Context_Dim_AttrDim c on a.contextRef=c.contextID
    where c.dimension ='es-be-d-FR-dist:DistribucionDimension'
        and c.attrDim like '%EntidadesSeguros%'
        and a.baseID='ifrs-gp:InvestmentsInSubsidiariesAtCost'
        and b.entityID=@entity and b.instantPeriod=@instantPeriod
    --select '@entidadesSeguros'_A, @entidadesSeguros_A
    if @entidadesSeguros_A is null
        begin
            if @fallbackvalue is not null
                begin
                    set @entidadesSeguros=@fallbackvalue
                    --select '@entidadesSeguros', @entidadesSeguros
                end
            else
                begin
                    set @evaluar=0
                    set @assertion_error=1
                end
            end
        end
else
    begin
        set @entidadesSeguros=cast(@entidadesSeguros_A as int);
        --select '@entidadesSeguros...:', @entidadesSeguros
    end
end

```

```

set @otrasEntidades_A=null;
set @otrasEntidades=null;
set @fallbackvalue=0;
select @otrasEntidades_A=a.Value
    from TXBRL_Lab6.dbo.Facts a inner join TXBRL_Lab6.dbo.Context b on
a.contextRef=b.ID
    inner join TXBRL_Lab6.dbo.Context_Dim_AttrDim c on a.contextRef=c.contextID
where c.dimension ='es-be-d-FR-dist:DistribucionDimension'
    and c.attrDim like '%OtrasEntidades%'
    and a.baseID='ifrs-gp:InvestmentsInSubsidiariesAtCost'
    and b.entityID=@entity and b.instantPeriod=@instantPeriod
--select '@otrasEntidades'_A, @otrasEntidades_A
if @otrasEntidades_A is null
begin
    if @fallbackvalue is not null
    begin
        set @otrasEntidades=@fallbackvalue
        --select '@otrasEntidades', @otrasEntidades
    end
    else
    begin
        set @evaluar=0
        set @assertion_error=1
    end
end
else
begin
    set @otrasEntidades=cast(@otrasEntidades_A as int);
    --select '@otrasEntidades...:', @otrasEntidades
end

set @ajustesEliminaciones_A=null;
set @ajustesEliminaciones=null;
set @fallbackvalue=0;
select @ajustesEliminaciones_A=a.Value
    from TXBRL_Lab6.dbo.Facts a inner join TXBRL_Lab6.dbo.Context b on
a.contextRef=b.ID
    inner join TXBRL_Lab6.dbo.Context_Dim_AttrDim c on a.contextRef=c.contextID
where c.dimension ='es-be-d-FR-dist:DistribucionDimension'
    and c.attrDim like '%AjustesEliminaciones%'
    and a.baseID='ifrs-gp:InvestmentsInSubsidiariesAtCost'

```

```

        and b.entityID=@entity and b.instantPeriod=@instantPeriod
--select '@ajustesEliminaciones'_A, @ajustesEliminaciones_A
if @ajustesEliminaciones_A is null
begin
    if @fallbackvalue is not null
    begin
        set @ajustesEliminaciones=@fallbackvalue
        --select '@ajustesEliminaciones', @ajustesEliminaciones
    end
    else
    begin
        set @evaluar=0
        set @assertion_error=1
    end
    end
else
begin
    set @ajustesEliminaciones=cast(@ajustesEliminaciones_A as int);
    --select '@ajustesEliminaciones...:', @ajustesEliminaciones
end

set @entidadesGrupoTotal_A=null;
set @entidadesGrupoTotal=null;
set @fallbackvalue=NULL
select @entidadesGrupoTotal_A=a.Value
    from TXBRL_Lab6.dbo.Facts a inner join TXBRL_Lab6.dbo.Context b on
a.contextRef=b.ID
        inner join TXBRL_Lab6.dbo.Context_Dim_AttrDim c on a.contextRef=c.contextID
        where c.dimension ='es-be-d-FR-dist:DistribucionDimension'
        and c.attrDim like '%TotalSectorial%'
        and a.baseID='ifrs-gp:InvestmentsInSubsidiariesAtCost'
        and b.entityID=@entity and b.instantPeriod=@instantPeriod
--select '@entidadesGrupoTotal'_A, @entidadesGrupoTotal_A
if @entidadesGrupoTotal_A is null
begin
    if @fallbackvalue is not null
    begin
        set @entidadesGrupoTotal=@fallbackvalue
        --select '@entidadesGrupoTotal', @entidadesGrupoTotal
    end
    else
    begin
        set @evaluar=0

```

```

        set @assertion_error=1
    end
end
else
    begin
        set @entidadesGrupoTotal=cast(@entidadesGrupoTotal_A as int);
        --select '@entidadesGrupoTotal....:', @entidadesGrupoTotal
    end

--select '@evaluar.....:', @evaluar
if @evaluar =0
    begin
        if @assertion_error=1
            set @val_6610_sh_2_1_Message=0
        end
    else
        begin
            if not((((@entidadesCredito + @entidadesSeguros + @otrasEntidades +
@ajustesEliminaciones)=@entidadesGrupoTotal) and (@entidadesGrupoTotal=0)))
                begin
                    set @val_6610_sh_2_1_Message=0
                    --select '@val_6610_sh_2_1_Message.....:', @val_6610_sh_2_1_Message
                end
            end
        end

-----
        fetch next from implicitFilter
        into @entity, @instantPeriod;

end

close implicitFilter;
deallocate implicitFilter;

--if @val_6610_sh_2_1_Message=1
--    begin
--        select 'Asercion val_6610_sh_2_1 correcta';
--    end
--else
--    begin
--        select 'Asercion val_6610_sh_2_1 incorrecta';
--    end

```

**Código 6.6 Procedimiento de validación de la aserción *val\_6610\_sh\_2\_1***

Este conjunto se resume en la tabla de la *Figura 6.6*. El modelo en estrella se valida comparando las dimensiones, los atributos de dimensión y las restricciones en el MDM y el XBRLDM (*Tabla 5.1* del capítulo anterior). Finalmente, se realiza otra validación, los datos se comparan entre el documento instancia XBRL y los hechos en la base de datos. La primera columna de esta tabla, *Figura 6.6*, muestra el número de aserción testeado. La segunda, el tipo de implementación, que se identifica numéricamente con hasta cinco posibles procedimientos almacenados (plantillas en el algoritmo). En la siguiente columna se identifica el número, hay siempre dos, y el tipo de filtros implícitos. En la POC, ambos filtros implícitos son los conceptos básicos (BC) *Entity*, y *Period of Time*, por ello el número es dos. La cuarta columna es para identificar los filtros explícitos en toda la aserción. Como en las columnas previas, la cuarta consta de dos subcolumnas para indicar el tipo de filtro, aquí BC o dimensión (D), y el número de dimensiones y/o BC. La quinta columna indica las variables y está subdivida en cuatro columnas correspondientes a (1) número de variables, (2) tipo de variable (aquí BC o D), (3) número de filtros (aquí BC o D) por el número de variables y (4) número de no nulos *fallbackValues*. Y al final, tres columnas de la tabla, que representan respectivamente, si hay constantes, si hay test, y si hay precondiciones.

### TEST OF VALIDATION (REPORT 6610)

NumCheck	Type	ImplFilter		ExplFilt in Assert		Variables				Const	Test	Precon
		T	Num Filt	T	Num Filt	Num	T	Num FiltV	Fall back			
1	1	BC	2	BC	1	5	D	1 X 5	4	N	Y	N
2	2	BC	2	BC	2	5	D	1 X 5	5	Y	Y	N
3	3	BC	2	D	1	2	BC	1 X 5	2	N	Y	N
4	3	BC	2	D	1	2	BC	1 X 2	2	N	Y	N
5	3	BC	2	D	1	2	BC	1 X 2	2	N	Y	N
6	3	BC	2	D	1	2	BC	1 X 2	2	N	Y	N
7	3	BC	2	D	1	2	BC	1 X 2	2	N	Y	N
8	4	BC	2	D	1	1	BC	99 X 1	0	N	N	Y
9	3	BC	2	D	1	2	BC	1 X 2	2	N	Y	N
10	4	BC	2	D	2	1	BC	2 X 1	0	N	N	Y
11	3	BC	2	D	1	1	BC	1 X 1	0	N	Y	N
12	5	BC	2	---	---	2	BC	1 X 2	0	N	Y	N

Figura 6.6 Test de validación para el informe 6610

Por ejemplo, la fila 1 significa que es la primera muestra del informe seleccionado 6610 (utilizado en esta tesis como ejemplo al final de la sección 5.4, Figuras 5.6 y 5.7). La segunda celda indica que la plantilla utilizada es la primera (la cual es un procedimiento almacenado SQL). Las celdas tercera, cuarta, quinta, y sexta son los filtros utilizados (dos y una dimensiones de conceptos básicos para cada clase de filtro, respectivamente). Las celdas séptimas, octava, novena y décima significan cinco variables, las cuales son dimensiones y no son conceptos básicos, cada variable incluye una dimensión, y todas esas variables son inicializadas si el valor es nulo, excepto la última (Figura 5.6). La tres últimas celdas indican que no hay un valor constante, la validación se hace a través de una formula, y que no hay precondiciones. Y de esta manera se pueden explicar las demás filas de la tabla de la Figura 6.6.

Esta POC se realizó escogiendo el mayor número de ejemplos diferentes en FINREP 2008. No obstante, no se validaron todas las aserciones de una forma automática, sino que solo se escogió una muestra de aproximadamente el 10%, ya que el resto

de las fórmulas era del mismo tipo. En las POCs posteriores de *León (2012)* y *Cid (2013)*, se utiliza un modelo en estrella puro, sin los contextos, como el mostrado en la *Figura 4.4*, pero con las taxonomías de *FINREP 2014* y *Solvency II*. En estas pruebas se utilizó el 100% del juego de pruebas facilitado en Eurofiling, la EBA y la EIOPA, no obstante el autor de este trabajo considera que en absoluto son suficientes pues el conjunto proporcionado por las instituciones no contempla al completo el juego de fórmulas definidas en el *Capítulo 5*.

Sería deseable para un futuro cercano, contar con un conjunto de pruebas completo a nivel de fórmulas y/o aserciones. Pese a ello, podemos concluir que los experimentos han determinado la validez del modelo propuesto.

# CAPÍTULO 7

---

Conclusión y trabajo  
futuro



## CAPÍTULO 7. Conclusión y trabajo futuro

---

### 7.1. Conclusiones

Esta tesis realiza, por un lado, un análisis de todas las etapas de desarrollo de la creación de los metadatos de informes económicos financieros, usando la especificación *eXtensible Business Reporting Language* (XBRL), y por otro, se analiza el modelo de fórmulas, que se propone en esta especificación, debido a la importancia que tiene para la validación de la información transmitida con este estándar, entre entidades financieras. De este análisis se extraen diversas carencias, como la imposibilidad de interoperabilidad en el estado actual, una falta de desarrollo completo y automático de la generación semántica de informes financieros, o la imposibilidad de realización de juegos de pruebas. Este trabajo propone solventar estas carencias mediante un proceso general de desarrollo desde el mundo real a su implementación física en base de datos. Para ello, se hace uso del paradigma *Model Driven Architecture* (MDA), que es una metodología propuesta por el *Object Management Group* (OMG) y ampliamente utilizada en proyectos de desarrollo de software. Esta tesis presenta el diseño del ciclo de vida del desarrollo software, desde los usuarios expertos, el *Computation Independent Model* (CIM), al código; se escoge el *Multidimensional Data Model* (MDM) en el *Platform Independent Model* (PIM), ya que este modelo se ajusta al desarrollo de los metadatos Europeos, y se escoge el modelo relacional (*Relational On-Line Analytical Processing*, ROLAP), en el *Platform Specific Model* (PSM), para su implementación en un *Relational Database Management System* (RDBMS). Por medio del MDM se formalizan y auditan las definiciones y reglas, y la semántica del *XBRL Data Model* (XBRLDM). El MDM se acompaña de una formalización y representación matemática de las formalizaciones y reglas. Además se propone la automatización de esta transformación.

El objetivo de esta tesis es clarificar el modelo XBRL, además de realizar la conversión entre los modelos XBRL y MDM. Como objetivo adicional, la transformación del modelo conceptual al modelo lógico y al físico, se acompaña de la formalización y representación matemática de las definiciones y reglas de las fórmulas.

El autor de esta tesis considera que todo este trabajo permitirá a las compañías, instituciones financieras y autoridades reguladoras comprender y mejorar la especificación XBRL y sus capacidades.

Parte de este trabajo está basado en el desarrollo europeo de los conceptos económico/financieros, debido a que el autor ha trabajado en este marco europeo y ha participado en el grupo de *Eurofiling (2015)* y en el *Comité Européen de Normalisation Workshop XBRL (CEN/WS XBRL, CEN, 2013)*. Este trabajo conlleva un conjunto de desarrollos y Pruebas de Concepto (POC), que implementa esta investigación, las cuales han sido presentadas en fóruns especializados para su validación (*Santos y Castro, 2010, 2011b, 2012a; Openfiling, 2014; Santos y Nieto, 2014, 2015*).

La implementación de los metadatos en una base de datos, en vez de crear las taxonomías usando la especificación XBRL, está ganando cierto interés entre los reguladores. Esto es debido, en parte, al alto coste de la infraestructura de la especificación XBRL, teniendo en cuenta las restricciones económicas de la mayoría de las instituciones financieras, supervisores o reguladores, y compañías en general. Además, cuando los informes XBRL (documento instancias XBRL) son muy grandes, éstos llegan a ser pesados o incluso imposibles de implementar por medio de la tecnología *eXtensible Markup Language (XML)* (*Santos y Castro, 2011a*). La implementación en base de datos hace que este problema sea resoluble, especialmente en la validación a través de fórmulas.

Resumiendo, se puede afirmar que:

- La propuesta principal de esta tesis es el desarrollo del ciclo de vida completo del modelo de metadatos haciendo uso de una arquitectura robusta como MDA, no existente hasta la fecha. De esta forma se facilita la extensión de nuevos elementos y/o modificaciones haciendo uso de esta metodología.
- Estos informes, que llegan a ser de gran tamaño, pueden ser implementados directamente en bases de datos. En principio la implementación de estos informes en un gestor de base de datos, puede significar evitar la limitación del procesamiento de grandes informes en XBRL, facilitando la posibilidad de monitorizar su rendimiento y su posterior análisis de tuning.

- Al establecerse un ciclo de vida del diseño del modelo de datos se nos asegura menor número de errores en el diseño, como se ha comprobado en las pruebas de concepto y validación, y se da la posibilidad de hacer casos de conjuntos de pruebas, para analizar las anomalías y otras cuestiones semánticas.
- Por otra parte se ha probado con esta tesis la interoperabilidad de esta tecnología, ya en el propio diseño. La tesis muestra como este modelo puede ser implementado en base de datos de diferentes gestores y da pie, incluso, a mapeos con otras plataformas.
- Al utilizar el MDM y la tecnología ROLAP, se está ayudando a simplificar su complejidad, sin ser menos robusta, y a tener una infraestructura si no más barata, si compartida por otras aplicaciones.

Este trabajo ha sido refrendado por la revista *Information Systems* (JCR, Q2), en el artículo “*XBRL Formula Specification in the Multidimensional Data Model*” (Santos, et. al. 2016).

## 7.2. Trabajos Futuros

Este trabajo debe ser extendido, no sólo en Europa y a sus taxonomías, sino también a las taxonomías americana (USA), canadiense o japonesa. Además, se espera en un futuro que esta propuesta haga que la especificación XBRL cubra otros campos económicos, como por ejemplo el intercambio de informes de los estados de órdenes de compra o cálculos de preventa, venta y disponibilidad de productos.

Otro modelo lógico utilizado en Europa (European Banking Authority, EBA; European Central Bank, ECB; European Insurance and Occupational Pensions Authority, EIOPA, etc.) es el *Data Point Model* (DPM). Este modelo está muy cercano a las aplicaciones finales y se orienta exclusivamente al desarrollo utilizando la especificación XBRL (*Romanelli, 2007; Ruíz et al., 2012; DPM, 2014*). La incorporación del DPM a este ciclo de vida se analizará en trabajos futuros. Por otra parte, si este trabajo analiza los metadatos de los informes desde el punto de vista del esquema, en futuros trabajos, se tendrá que analizar el diseño del ciclo de vida de los propios informes, es decir, desde el punto de vista de los datos.

En la actualidad, los departamentos de Tecnología de la Información de los Organismos Supervisores y Reguladores, los cuales tienen grandes taxonomías, tienen un importante reto, ya que crean las taxonomías y sus validaciones sin casos de test públicos. Y esto es debido a que los supervisores están continuamente creando modelos más grandes y complejos.

Esta tesis ha validado de forma automática e implementado las fórmulas a través de procedimientos almacenados en bases de datos, tanto fuera o dentro de una base de datos. Un importante factor a considerar es el rendimiento de las fórmulas XBRL. Mientras en XML, es lo más parecido a una caja negra, en un RDBMS es bien conocido y analizado extensivamente. Así, un área a explorar en una futura investigación es el análisis y comparación del rendimiento de estas fórmulas tanto en XML como en base de datos.

El capítulo 5 presenta la base del diseño del ciclo de vida del desarrollo del software, desde los usuarios expertos (el CIM), al código. Sin embargo en este punto no hay diseño automático desde el universo del discurso al modelo físico. Por esta razón, los departamentos de Tecnología de la Información tienen problemas en la validación de la implementación por los usuarios expertos. Si bien es cierto que el diseño automático actualmente es inviable, se debería estudiar la posibilidad de semi-automatizar este proceso.

En la validación de las fórmulas, capítulo 6 sección 6.3, cuando se utiliza un modelo en estrella puro, sin los contextos, como el mostrado en la *Figura 4.4* (León, 2012; Cid, 2013), los casos de prueba son 4 para *FINREP 2014* y un borrador con un solo caso para *Solvency II*. No es hasta principios de este año cuando tanto la *EBA* como *EIOPA* sacan al público taxonomías con ejemplos de pruebas. Por lo que queda abierta la posibilidad de utilizar más conjuntos de pruebas para la validación de la propuesta. La diferencia entre la codificación de los algoritmos diseñados por el autor de esta tesis y aquellos generados por los alumnos tutelados, es que la segunda codificación es más sencilla. Esta tesis deja como trabajos futuros:

- Comparar la complejidad de ambos tipos de algoritmos, los basados en un modelo en estrella con contexto o sin contexto. Y el rendimiento de ambos.
- Comprobar el rendimiento de las validaciones de las fórmulas a través de procedimientos almacenados y compararlas con el rendimiento utilizando las validaciones XML.

El autor de esta tesis se queda con un reto mayor, para futuros trabajos de investigación: si se ha tratado que las fórmulas ya estaban desarrolladas en XBRLDM, el objetivo siguiente es partir del desarrollo de las fórmulas que generan los usuarios expertos. Este es un tema importante, ya que muchas organizaciones no pueden presentar casos de pruebas públicos, y en algún caso estas son erróneas, por la falta de un ciclo de desarrollo completo.

A pesar de la complejidad de los dominios y estándares financieros, y quizá debido precisamente ello, los expertos animan a la creación de proyectos propios y a la divulgación de trabajos relacionados con el sector, que clarifiquen y acerquen al mundo real todo este entramado de definiciones, taxonomías y datos. La utilización de técnicas de procesamiento del lenguaje natural, podría ayudar a acercar este mundo a usuarios finales describiendo semánticamente el contenido de los informes económico-financieros (*Mendez-Nunez y Trivino, 2010*).

Para finalizar, el trabajo futuro que se abre a partir de esta tesis, se puede resumir en seis líneas:

- Extender el ciclo de vida del desarrollo software de este tipo de informes, no sólo al modelo europeo, si no al resto, que no es puramente dimensional.
- Ampliar el uso de esta especificación a otros entornos que no son la Supervisión y la Regulación.
- Incorporar a este trabajo de investigación el *Data Point Model* (DPM), y su posterior implementación en XBRL.
- A través de este desarrollo del ciclo de vida, facilitar la creación de juegos de prueba públicos.
- Generar no sólo plantillas para procedimientos almacenados en SQL, si no también en otros lenguajes como COBOL, .NET, Java, etc.
- Análisis de técnicas de procesamiento del lenguaje natural para aumentar el contenido semántico de los informes.



## Bibliografía

---

### Libros

AICPA/ CICA (1999) Web Trust and Criteria for Business-to-Consumer Electronic Commerce, version 2.0. American Institute of Certified Public Accountants Canadian Institute of Chartered Accountants, 15 de octubre de 1999.

Boletín Oficial del Estado (BOE) (2007) Plan General contable (2007). Fecha 20 de Noviembre de 2007, anexo 278.

Cid C (2013) Mapa de carreteras del Modelo de Datos Multidimensional (MDM), haciendo uso de herramientas ROLAP (Relational Online Analytical Processing), de FINREP 2012 y Solvency II (01-07-2012) con validaciones, Prueba de Concepto. Proyecto final de carrera de la Escuela Politécnica de la Universidad Carlos III de Madrid, España. Fecha: 14 de noviembre de 2013. Tutores: Ignacio Santos y Elena Castro.

Felden C (2007) Multidimensional XBRL (New Dimensions of Business Reporting and XBRL). DUV Deutscher Universitäts-Verlag, páginas 191-209, 18 de noviembre de 2007.

Inmon W H (2005) Building the Data Warehouse. 4ª Edición. John Wiley & Sons.

Jarke M, Lenzerini M, Vassiliou Y, Vassiliadis P (2003) Fundamentals of Data Warehouses, 2ª Ed. Springer.

Kernan K (2009) The story of our new language personalities, cultures, and politics combine to create a common, global language for business. American Institute of certified Public Accountants. New York, NY 10036-8775.

Kimball R (1996-2004) The Data Warehouse Toolkit series. John Wiley & Sons 1996-2004.

León Y (2012) Automatización de la transformación del modelo de datos de la especificación XBRL en Base de datos. Trabajo Fin de Grado de la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, España. Fecha: 4 de

octubre de 2012. Tutores: Ignacio Santos y Elena Castro. URI: <http://hdl.handle.net/10016/16843>.

Martín Quetglás G (2006). Curso de XBRL. Pearson Educación. ISBN 9788483223116.

Morrison M (1999). XML Unleashed. Editor Sams Publishing. ISBN 10:0672315149.

Nieto, A (2015). *Validación de Informes Económicos / Contables / Financieros Semánticos y su implementación en Base de Datos, de una forma automática (Validation of Semantic Economic / Accounting / Financial Reports and its Implementation in Database, in an automated way)*. Proyecto Final de Carrera en Ingeniería técnica de informática de la Universidad Carlos III de Madrid, España. Fecha: 13 de octubre de 2015. Tutores: Ignacio Santos y Elena Castro.

Rusty Harold E (2004). XML 1.1 Bible 3<sup>rd</sup> Edition. Publish by Wiley Publishing. Inc. ISBN: 0-7645-4986-3.

Sperberg-McQueen C M, Burnard L (2002). Chapter 2: A Gentle Introduction to SGML of Guidelines for Electronic Text Encoding and Interchange. Publish: University of Virginia Press, Charlottesville, 2002. ISBN 10: 095233013X.

The Larosière Group (2009) The High-Level Group on Financial Supervision in the EU. Fecha: 25 de febrero de 2009. The Larosière Group: Jacques de Larosière (Chairman), Leszek Balcerowicz, Otmar Issing, Rainer Masera, Callun Mc Carthy, Lars Nyberg, José Pérez, Onno Ruding, Bruselas, Bélgica.

Valencia J (2011) Modelo de meta-metadatos XBRL. Proyecto Final de Carrera en Ingeniería técnica de informática de la Universidad Carlos III de Madrid. Tutores: Ignacio Santos y la Dra. Elena Castro. 27 de septiembre de 2011. Accesible en: <http://hdl.handle.net/10016/13157>

## Revistas

Agrawal D, El Abbadi A, Singh A, Yurek T (1997) Efficient view maintenance at data warehouses. Proceeding SIGMOD '97, Proceedings of the 1997 ACM SIGMOD

international conference on Management of data. Volumen 26 número 2, junio de 1997, páginas 417-427.

Aljumaily H, Cuadra D, Martínez P (2014) Applying black-box testing to UML/OCL database models. *Software Quality Journal*. Junio 2014, Volumen 22, número 2, páginas 153-184.

Bennett M (2013) The financial industry business ontology: Best practice for big data. Fecha: 1 de julio de 2013. *Journal of Banking Regulation*, vol. 14, números 3-4 (Julio/Noviembre 2013). Special issue: The future of financial regulatory data *J Bank Regul* 14:25555-268. Doi:10.1057/jbr.293,13

Boixo I, Flores F (2005) New Technical and Normative Challenges for XBRL: Multidimensional in the COREP Taxonomy. Fecha: 18 de Julio de 2005. *The International Journal of Digital Accounting Research*, Vol. 5, Nº 9, 2005, pp 79-104. ISSN. 1577-8517.

Callaghan H, Savage A, Sugumaran V (2002) Augmenting XBRL Using UML Improving Financial Analysis. Fecha: 1 de septiembre de 2002. *The Review of Business Information Systems*. Quarter 4, vol 6, nº 4.

Cuadra D, Aljumaily H, Castro E, Velasco M (2011) An OCL-Based approach to derive constraint test cases for database applications. *International Journal of Software Engineering and Knowledge Engineering* 21 (2011) 5, pp 621-645. ISSN: 0218-1940. DOI: 10.1142/80218194011005426.

Debreceny R, Farawell S, Piechocki M, Felden C, Graning A (2010) Does it Add up? Early evidence on the data quality of XBRL filings to the SEC. Fecha 1 de junio de 2010. *Journal of Accounting and Public Policy*. Volumen 29, número 3 de junio 2010, páginas 296-306.

Debreceny R, Gray G L (2001) The production and use of semantically rich accounting reports on the Internet: XML and XBRL. *International Journal of Accounting Information Systems*, volumen 2, número 1, enero de 2001, páginas 47-74.

Feng M, Dong G, Li J, Tan YP, Wong L (2010) Pattern Space Maintenance for Data Updates and Interactive Mining. *Computational Intelligence*, 26(3):282--317, agosto de 2010.

Gogolla M, Büttner F, Richters M (2007) USE: A UML-Based Specification Environment for Validating UML and OCL. *Science of Computer Programming*, 69:27.

Gräning M, Büttner F, Richters M, (2007) USE: A UML-Based Specification Environment for Validating UML and OCL. *Science of Computer Programming*, 69:27-34.

Graning A, Felden C, Piechocki M (2011) Status Quo and potential of XBRL for business and Information Systems Engineering. Fecha 12 de julio de 2011. *Business & Information Systems Engineering*, vol 3, número 4, páginas: 231-239.

Hodge F D, Kennedy J I, Maines L A (2004) Does search-facilitating technology improve the transparency of financial reporting? *Accounting Review*, 79 (3), páginas 687-703.

Jimei L, Yuzhou H, Meijie D (2013) XBRL in the Chinese Financial Ecosystem. *IT Professional Magazine*, (Dec 2013): 36.

Lara R, Cantador I, Castells P (2006) XBRL taxonomies and OWL ontologies for investment funds. *Advances in Conceptual Modeling-Theory-Theory and Practice. Lecture Notes in Computer Science*, volume 4231, 20066, pp 271-280, enero de 2006.

Lee H L, Padmanabhan V, Whang S (1997) The bullwhip effect in supply chains. *Sloan Management Review*, 38(3)-93-102, año 1997.

Lin F, Sheng O R, Wu S (2005) An Integrated framework for eChain bank accounting systems. Fecha 1 de marzo de 2005. *Industrial Management & Data Systems*, Vol. 105 (3/4).

Loukis E N, Charalabidis (2013). An empirical investigation of information interoperability business in European Firms. Fecha 1 de mayo de 2013. *Computers in Industry*, Vol. 64, número 4, mayo de 2013, páginas 412-420.

Nickerson R C, Varshney U, Muntermann (2013). A Method for taxonomy development and its application in information systems. Fecha 1 de mayo de 2013. *European Journal of Information Systems*, número 22, 336-359. Doi:10.1057/ejis.2012.26; publish on-line 19 de junio de 2012.

Prasanna N K, Anjaneyulu A, Shasi M (2012) Semantic Web and Economic and Financial Information Management. *Journal of Theoretical and Applied Information Technology*, volumen 18, número 2.

Santos I, Castro E, Velasco M (2016) XBRL *Formula Specification in the Multidimensional Data Model*. En *Information Systems*, volumen 57, abril de 2016, páginas 20-37.

Silveira C, Abreu R, David F (2007) From an Object-Oriented Approach to the Financial Reporting: An Open Architecture. *The International Journal of Digital Accounting Research*, volumen 7, número 13, 2007, páginas 1-25. ISSN: 1577-8517.

Spies M (2010) An ontology modelling perspective on business reporting. *Information Systems*, volumen 35, número 4, junio de 2010, páginas 404-416.

Wagenhofer A (2003) Economic consequences of Internet financial reporting. *Schmalenbach Business Review (SBR)*, 55(4): 262-279.

Williams S P, Scifleet P A, Hardy C A (2006) On line business reporting: an information management perspective. *International Journal of Information Management*, 26(2):91-101.

Woodroof J, Searcy D. W (2001) Continuous audit Model development and implementation within a debt covenant compliance domain. *International Journal of Accounting Information Systems*, volumen 2, número 3, septiembre de 2001, páginas 169-191.

## Congresos

Buder J, Koschtial C, Felden C (2009) Formalization of REA ontology. Fecha 6 de Agosto de 2009. *Proceeding of the fifteenth Americas conference on information systems*, San Francisco, California. Del 6 al 9 de agosto de 2009.

Callaghan J H, Savage A, Sugumaran V (2002) Augmenting XBRL Using UML Improving Financial Analysis. *European Conference on Information Systems*. 2002, trimestre: 4-volumen 6, número 4, Modeling XBRL Using UML: Improving Semantics for Financial Analysis.

Callaghan J H, Nehmer R, Sugumaran (2006) Modeling XBRL-Based Applications with UML: Developing Valanced-Scorecard Management Appraisal Systems.

*Proceedings of 12th Americas Conference on Information Systems AMCIS 2006*, Acapulco, Mexico. Del 4 al 6 de agosto de 2013.

Declerck T, Krieger H U (2003) Traslating XBRL into description logic. An approach using protégé, sesame and OWL. *9<sup>th</sup> International Conference on Business Information Systems in cooperation with CAN SIGMIS*, Klagenfurt Austria. Del 31 de mayo al 2 de junio de 2006.

Di Giovanni F, Piazza D (2009) Processing and managing statistical data: a National Central Bank experience. Fecha 15 de septiembre de 2009. *International Statistical Conference*, Prague, Czech Republic, del 14 al 15 de septiembre de 2009. <http://www.czso.cz>.

Díaz A (2012) Opening & Welcome (XBRL Week in Madrid), 11 de mayo de 2012. [http://www.eurofiling.info/15th\\_worksop/presentations/AnselmoDiaz-welcome.pdf](http://www.eurofiling.info/15th_worksop/presentations/AnselmoDiaz-welcome.pdf).

Golfarelli M, Rizzi, Urdoljak B, (2001). Data Warehouse design from XML Sources.Fecha 3 de septiembre de 2001. *Proceedings of 4<sup>th</sup> ACM International Workshop on Data Warehousing on OLAP*, páginas 40-47.

JiMei J, Hui Zou Z, MeiJie D U (2012) Analizing Semantic Heterogeneity in XBRL Taxonomies: an Ontology Perspective. *Proceeding ICMECG12, Proceeding of the 2012 International Conference on Management of e-Commerce and e-Government*.

Mendez-Nunez, S., Trivino, G. (2010) Combining Semantic Web Technologies and Computational Theory of Perceptions for Text Generation in Financial Analysis. Published in: *Fuzzy Systems (FUZZ)*, 2010 IEEE International Conference. Fecha de la conferencia: 18-23 de julio de 2010, página(s): 1 – 8, ISSN: 1098-7584.

Morilla V (2008) XBRL Formula Tutorial, 29 de septiembre de 2008. *IX European Banking Supervisors. XBRL Workshop*, Paris, France.

Piechocki M.; Felden C., Gräning A (2007) Multidimensional XBRL Reporting. Fecha: 25 de marzo de 2007. *Proceedings of the Fifteen European Conference on Information Systems*.

Romanelli M (2007) Matrix Schemas for COREP & FINREP Templates November 6<sup>th</sup>, 2007. *VIII CEBS-XBRL Workshop*. Amsterdam, del 6 al 7 de noviembre de 2007.

Santos I, Castro E (2010) XBRL, UML and Databases: State of Art. XIII European Banking Supervisors XBRL Workshop. Lonxembourg. Del 24 al 25 de noviembre de 2010. Orgnized by the XBRL Operational Network of the Committee of European Banking Supervisors, and hosted by the Commission de Surveillance du Secteur Financier of Luxembourg.

Santos I, Castro E (2011a) XBRL and the Multidimensional Data Model. *Proceedings of the 7<sup>th</sup> International Conference on Web Information Systems and Technologies, WEBIST 2011*, páginas 161-164, Noordwijkerhout, The Netherlands, del 6 al 9 de mayo de 2011.

Santos I, Castro E (2011b) Proof of Concept of mapping a XBRL reports versus a RDBMS. 5 de septiembre de 2011. *XBRL Openfiling 1<sup>st</sup>, General Assembly, organized by XBRL Operational Network of the European Banking Authority*, and hosted by the Banca D'Italia, Rome, Italy.

Santos I, Castro E (2011c) XBRL Interoperability through a Multidimensional Data Model. *IADIS International Conference on Internet Technologies & Society (ITS 2011)*. Shanghai, China, del 8 al 10 de diciembre de 2011.

Santos I, Castro E (2012a). Validation of a XBRL Document Instance in a RDBMS, Proof of Concept. Fecha 1 de junio de 2012. *15<sup>th</sup> Eurofiling Workshop, organized by XBRL initiative along with WBRL Europe* and hosted by the Bank of Spain, Madrid, Spain.

Santos I (2013) Data Point Model (DPM) versus Multidimensional Data Model (MDM). Contribution for DPM Chapter in CEN WS XBRL Plenary Session, Dublin, 19 de abril de 2013. Hosted by the Central Bank of Ireland. [www.openfiling.info/academy](http://www.openfiling.info/academy).

Santos I. y Nieto A (2014). Formal Validation of Data Point Model. XBRL week in Brussels. National Bank of Belgium. 26 de noviembre de 2014. <http://eurofiling.info/201411/index.shtml>.

Santos I, Nieto A, (2015) Validation of the DPM and database implementation. 15<sup>th</sup> XBRL EU Day & 21<sup>st</sup> Eurofiling Workshop, XBRL week in Madrid, hosted by Business Register and Bank of Spain, 3 de junio de 2015. <http://www.eurofiling.info/201506/index.shtml>.

Schmehl, K. y Ochocki, B (2009) Data Model and Matrix Schemas. <http://www.eurofiling.info>. XI European Banking Supervisor XBRL Workshop. 16 de noviembre de 2009, Vienna.

Snijders P (2005) XBRL and Complex Data Mapping. Fecha 11 de febrero de 2005. Conference for Software Developers, Madrid, España,

Sugumaran V, Callaghan J, Sarage A (2002) Modeling XBRL Using UML: Improving Semantics for Financial Analysis. *AMMCIS 2002 Proceeding (AMCIS). Association for Information Systems. AIS Electronic /AISel) Americas Conference on Information Systems.*

Weller, A (2015) EBA Taxonomies or “Benny the tomcat serve to demonstrate the evolution in Regulatory reporting in the financial sector”. 15<sup>th</sup> XBRL EU, XBRL week in Madrid, hosted by Business Register and Bank of Spain. 2 de junio de 2015. <http://www.eurofiling.info/201506/index.shtml>.

Zhu H, Fu L (2009) Quality of Data Standards: Empirical finding from XBRL. Fecha 15 de diciembre de 2009. 30<sup>th</sup> International Conference on Information System (ICIS'09), December 15th-18th Phoenix, AZ, USA.

## Referencias en Internet

Arelle (2015) Open source XBRL platform, <http://arelle.org/>.

Banco de España (2010) Banco de España's Data Point Model for collecting BSI/MIR Statistics Using XBRL. Fecha 16 de Julio de 2010. Directorate General Banking Regulation, Bank of Spain. Regulations ECB/2008/32 and ECB/2001/18 amended by ECB/2009/7.

Banco de España (2011) Descripción de la adaptación nacional de la taxonomía FINREP. 26 de noviembre de 2011. Edit: Bank of Spain. [http://www.bde.es/es/fr/documentacion/taxonomias/es-be-finrep/6-2008/2008-11-26/es-be-finrep\\_tecnico.pdf](http://www.bde.es/es/fr/documentacion/taxonomias/es-be-finrep/6-2008/2008-11-26/es-be-finrep_tecnico.pdf).

Basel II (2004) Bank for International Settlements. <http://www.bis.org/publ/bcbs107.htm>.

Basel III (2010) Bank for International Settlements. Fecha 12 de Septiembre de 2010. <http://www.bis.org/publ/bcbs189.pdf>, [www.oecd.org/finance/financiaal-markets/45314422.pdf](http://www.oecd.org/finance/financiaal-markets/45314422.pdf), [www.basel-iii-acord.com](http://www.basel-iii-acord.com).

BR-AG (2015) Business Reporting. <http://www.br-ag.eu/>.

BSI-MIR (2010) Balance Sheet Items (BSI) and Monetary, Financial Institutions (MFI). Interest Rates (MRI). Fecha 27 de enero de 2010. [www.eurofiling.info/bsi-mirTaxonomies/Taxonomy.Shtml](http://www.eurofiling.info/bsi-mirTaxonomies/Taxonomy.Shtml).

CEN (2013) Workshop Agreement CWA. « <http://cen.eurofiling.info/> ». Fecha: 11 de diciembre de 2013.

CEN (2014) Comité Européen de Normalisation/European Committee for Standarization. <http://www.cen.eu>.

CWA (2014) CEN Workshop Agreement. <http://cen.eurofiling.info>.

EBA (2011) EBA BS 180 (CP50 ITS on reporting-Annex III to V FINREP templates and instructions). Fecha: 30 de noviembre de 2011.

EBA (2013) Update on the technical standards on supervisory reporting requirements. EBA. <http://www.eba.europa.eu/-/update-on-the-technical-standards-on-supervisory-reporting-requirements>.

EBA (2014) European Banking Authority Data Point Model and Taxonomies for Implementing Technical Standard (ITS) on Supervisory Reporting. <http://www.eba.europa.eu/>.

EIOPA (2014) (European Insurance and Occupational Pensions Authority) Solvency II Reporting Format (2014). <https://eiopa.europa.eu/publications/solvency-ii-reporting-format/index.html>.

Engel P, Shuetrim G, y Kannon D. V (2006) XBRL Formula 1.0 Specification. [www.xbrl.org](http://www.xbrl.org).

Engel P, Hamscher W, Shuetrim G, Kannon D V, Wallis H (2008) Extensible Business Reporting Language (XBRL) 2.1. XBRL International. <http://www.xbrl.org/Specification>.

Eurofiling (2011) Proof-of-concept simple, Eurofiling data modeling and taxonomy architecture. Fecha 2 de junio de 2011. Copyright © 2005-2011 European Banking Authority using European Union Public Licence.

Eurofiling (2012) Eurofiling: New, 2012 FINREP taxonomy. <http://eurofiling.info/Finreptaxonomy/taxonomy2012.shtml>.

Eurofiling (2015). [www.eurofiling.info/index.shtml](http://www.eurofiling.info/index.shtml).

EXGEN (2012) "Data Point Modeler, Excel XBRL Generator (Excel add in). Fecha 15 de febrero de 2012. En [www.openfiling.info/exgen/](http://www.openfiling.info/exgen/).

Fischer H (2011) XBRL Formula Overview 1.0, Public Working Draft 21 December 2011. Fecha 21 de diciembre de 2011. XBRL international. Public Working Draft 21 de diciembre de 2011.

Hamscher W, y Kannon D. V (2000) Extensible Business Reporting Language (XBRL) Specification. Fecha 31 de julio de 2000 por XBRL International. [www.xbrl.org](http://www.xbrl.org).

Hernández-Ros I y Wallis, H (2006) XBRL Dimensions 1.0. XBRL International <http://www.xbrl.org/Specification/XDT-CR3-2006-04-26Af>.

Hernández-Ros, I (2009) XBRL Infoset 0.3. Fecha 4 de febrero de 2009. XBRL International Inc. Public Working Draft 4 de febrero de 2009.

IFRS (2015) International Financial Reporting. [www.ifrs.org/home.htm](http://www.ifrs.org/home.htm).

Inteco (2009) Manual de usuario API XBRL LENLOC PENLOC. Fecha 1 de noviembre de 2009. Autor Inteco (Instituto Nacional de Tecnología de la Comunicación de España). Laboratorio Nacional de Calidad de Software/Proyecto SBRL LENLOC PENLOC. <http://www.inteco.es>.

ISO 20022 (2015) Standard for Financial Services Messaging. <https://www.iso20022.org/>.

MDA (2015) OMG Model Driven Architecture. [www.omg.org/mda](http://www.omg.org/mda).

OMG (2015) Object Management Group, Inc. <http://www.omg.org/mda/>.

Openfiling (2015) Academy. "http://www.openfiling.info/academy/".

- Ruíz A, Ochocki B, Brandt D, De Fiedura R, Morilla,, Sever W.van, Boixo I (2012) Data Point Modeler: EXGEN. Excel XBRL Generator (Excel add in). February 15<sup>th</sup>, 2012. <http://www.openfiling.info/exgen/>.
- Santos I, Hommes R, Heinze K (2013) Improving transparency in financial and business reporting - Harmonisation topics – Part 5: Mapping between DPM and MDM. [http://cen.eurofiling.info/wp-content/uploads/data/CWA\\_XBRL\\_WI001-5-E.pdf](http://cen.eurofiling.info/wp-content/uploads/data/CWA_XBRL_WI001-5-E.pdf).
- SDMX (2015) Statistical Data and Metadata Exchange. <http://sdmx.org/>.
- SEC (2015) U. S. Securities and Exchange Commission. [www.sec.gov](http://www.sec.gov).
- SEPA (2014) Simple Euro Payment Area. <http://www.sepaesp.es/sepa/es/>.
- Si Alhir S (2015) Understanding the Model Driven Architecture (MDA). <http://www.methodsandtools.com/archive/archive.php?id=5>.
- T2 (2013) Target 2. <http://www.ecb.europa.eu/paym/t2/html/index.en.html>.
- T2S (2014) Target 2 Securities. <http://www.ecb.europa.eu/paym/t2s/html/index.html>.  
Y <https://www.ecb.europa.eu/paym/t2s/about/html/index.en.html>.
- W3schools (2015). Company Refsnes Data <http://www.W3schools.com>.
- Weller A (2015) EBA Taxonomies or “Benny the tomcat serve to demonstrate the evolution in Regulatory reporting in the financial sector”, 15<sup>th</sup> XBRL EU, XBRL week in Madrid, <http://www.eurofiling.info/201506/index.html>.
- XML.com (2015. O’Reilly “XMLO.com”, using W3C XML Schema. O’Reilly Media, Inc. [www.xml.com](http://www.xml.com).
- XBRL España (2006). Libro Blanco, Grupo de Trabajo de Tecnología 2006. [www.xbrl.es/formacion/formacion.html](http://www.xbrl.es/formacion/formacion.html).
- XBRL Formula (2011) Overview 1.0.Public Working Draft, 21 de diciembre de 2011. XBRL International Public Working Draft.
- XBRL International and Novartis International AG (2002) Samples XBRL 1.0, 15 de noviembre de 2002. <http://www.xbrl.org/taxonomy/int/fr/ias/ci/pfs/2002-11-15/Samples.htm>.

XBRL International (2009) Formula 1.0 Recommendation 22 de junio de 2009.  
<http://specifications.xbrl.org/work-product-index-formula-formula-1.0.html>.

XBRL International (2009a) Implicit Filters 1.0. 22 de junio de 2009.  
<http://www.xbrl.org/Specification/implicitFilters/REC-2009-06-22/implicitFilters-REC-2009-06-22.html>.

XBRL International (2009b) Concept Filters 1.0. 22 de junio de 2009.  
<http://www.xbrl.org/Specification/conceptFilters/REC-2009-06-22/conceptFilters-REC-2009-06-22.html>.

XBRL International (2009c) Variables 1.0. 22 de junio de 2009.  
<http://www.xbrl.org/Specification/variables/REC-2009-06-22/variables-REC-2009-06-22.html>.

## Acrónimos

---

NOMBRE	DESCRIPCIÓN	REFERENCIA
<b>AICPA</b>	American Institute of Certified Public Accountants	<a href="http://www.aicpa.org">www.aicpa.org</a>
<b>API</b>	Application Programming Interface	
<b>APRA</b>	Australian Prudential Regulation Authority	<a href="http://www.apra.gov.au/Pages/default.aspx">http://www.apra.gov.au/Pages/default.aspx</a>
<b>BdE</b>	Bank of Spain	<a href="http://www.bde.es">www.bde.es</a>
<b>BIS</b>	Bank for International Settlements	<a href="http://www.bis.org">www.bis.org</a>
<b>BR-AG</b>	Business Reporting	<a href="http://www.br-ag.eu/about">www.br-ag.eu/about</a>
<b>BSI</b>	Balance Sheet Items (taxonomy)	
<b>CDM</b>	Conceptual Data Model	
<b>CEN</b>	Comité Européen de Normalisation/European Committee for Standardization	<a href="http://www.cen.eu">http://www.cen.eu</a>
<b>CICA</b>	Canadian Institute of Chartered Accountants	<a href="http://www.cica.ca">www.cica.ca</a>
<b>CIM</b>	Computation Independent Model	MDA
<b>CNMV</b>	Securities Market Commission (Spain)	<a href="http://www.cnmv.es">www.cnmv.es</a>
<b>CWM</b>	Common Warehouse Metamodel	<a href="http://www.omg.org/spec/CWM/">http://www.omg.org/spec/CWM/</a>
<b>DOM</b>	Document Object Model	Parser XML
<b>COREP</b>	Common Reporting (taxonomy)	<a href="http://www.eurofiling.info/corepTaxonomy/taxonomy.shtml">http://www.eurofiling.info/corepTaxonomy/taxonomy.shtml</a>
<b>CRD</b>	Capital Requirements Directive	<a href="http://ec.europa.eu/internal_market/bank/index_en.htm">http://ec.europa.eu/internal_market/bank/index_en.htm</a>

<b>NOMBRE</b>	<b>DESCRIPCIÓN</b>	<b>REFERENCIA</b>
<b>CWA</b>	CEN Workshop Agreement	<a href="http://www.eurofiling.info/cen/cwa/">www.eurofiling.info/cen/cwa/</a>
<b>DBMS</b>	Database Management System	
<b>DM</b>	Data Mart	
<b>DPM</b>	Data Point Model	
<b>DTD</b>	Document Type Definition	
<b>DTS</b>	Discoverable Taxonomy Set	Set of taxonomies (XML files) that are necessary for processing a XBRL document.
<b>DW</b>	Data Warehouse	
<b>EBA</b>	European Banking Authority	<a href="http://www.eba.europa.eu">www.eba.europa.eu</a>
<b>ECB</b>	European Central Bank	
<b>EDGAR</b>	Electronic Data Gathering, Analysis and Retrieval system	<a href="http://www.sec.gov/edgar.shtml">www.sec.gov/edgar.shtml</a>
<b>EDITRAN / EDI</b>	Electronic data interchange Transactions / Electronic Data Interchange	<a href="http://www.editran.info">www.editran.info</a>
<b>EDOC</b>	Enterprise Distributed Object Computing	<a href="https://edoc2015.unisa.edu.au/">https://edoc2015.unisa.edu.au/</a> , <a href="http://ieeexplore.ieee.org/xpl/conhome.jsp?punumber=1000274">http://ieeexplore.ieee.org/xpl/conhome.jsp?punumber=1000274</a> .
<b>EIOPA</b>	European Insurance and Occupational Pensions Authority	<a href="http://www.eiopa.europa.eu">www.eiopa.europa.eu</a>
<b>EMU</b>	Economic and Monetary Union	<a href="http://www.ecb.int/ecb/history/emu/html/index.en.html">www.ecb.int/ecb/history/emu/html/index.en.html</a>
<b>ESCB</b>	European System of Central Banks	<a href="http://www.ecb.europa.eu/ecb/orga/escb/html/index.en.html">http://www.ecb.europa.eu/ecb/orga/escb/html/index.en.html</a>
<b>E/RM</b>	Entity/Relationship Model	
<b>EU</b>	European Union	

<b>NOMBRE</b>	<b>DESCRIPCIÓN</b>	<b>REFERENCIA</b>
<b>EUROSYSTEM</b>	Euro System	It is the monetary authority of the Eurozone.
<b>EUROZONE</b>	Euro Area	States of the European Union with the euro (€) as currency.
<b>EXGEN</b>	Data Point Modeler: EXGEN. Excel XBRL Generator (excel add in)	<a href="http://www.openfiling.info">http://www.openfiling.info</a>
<b>FED</b>	Board of Governors of the Federal Reserve System	<a href="http://www.federalreserve.gov/">http://www.federalreserve.gov/</a>
<b>FIN</b>	Financial message	Service of SWIFT
<b>FINREP</b>	Financial Report (taxonomy)	<a href="http://www.eurofiling.info/finrepTaxonomy/taxonomy.shtml">http://www.eurofiling.info/finrepTaxonomy/taxonomy.shtml</a>
<b>FIX</b>	Financial Information eXchange	
<b>GML</b>	Generalized Markup Language	
<b>HOLAP</b>	Hybrid OLAP	
<b>HTML</b>	HyperText Markup Language	
<b>HTTP</b>	HyperText Transfer Protocol	
<b>HTTP-S/HTTPS</b>	Hypertext Transfer Protocol Secure	
<b>IAS/IFRS</b>	International Accounting Standards/International Financial Reporting Standard	<a href="http://www.ifrs.org/home.htm">www.ifrs.org/home.htm</a>
<b>IBM</b>	International Business Machine	<a href="http://www.ibm.com">www.ibm.com</a>
<b>IFRS</b>	International Financial Reporting	<a href="http://www.ifrs.org/home.htm">www.ifrs.org/home.htm</a>
<b>INTERACT</b>	Interavtive	Protocolo para el automatizado e interactivo intercambio de mensajes síncronos o asíncronos, y utilizado por SWIFT.
<b>IPSEC</b>	Internet Protocol Security	Usado por SWIFT.

<b>NOMBRE</b>	<b>DESCRIPCIÓN</b>	<b>REFERENCIA</b>
<b>IS</b>	Information Systems	
<b>ISDA</b>	International Swaps and Derivatives Association	<a href="http://www2.isda.org/">http://www2.isda.org/</a>
<b>ISO</b>	International Organization for Standardization/ Organización Internacional para la Estandarización	<a href="http://www.iso.org">http://www.iso.org</a>
<b>IT</b>	Information Technology	
<b>ITS</b>	Internet Technologies & Society	<a href="http://its-conf.org/">http://its-conf.org/</a>
<b>ITU</b>	International Telecommunication Union/ Unión Internacional de Telecomunicaciones	<a href="http://www.itu.int">www.itu.int</a>
<b>MDA</b>	Model Driven Architecture	<a href="http://www.omg.org/mda">www.omg.org/mda</a>
<b>MDM</b>	Multidimensional Data Model	
<b>MIR</b>	Monetary Financial Institutions Interest Rates (taxonomy)	
<b>MOF</b>	Meta-Object Facility	<a href="http://www.omg.org/mof/">http://www.omg.org/mof/</a>
<b>MOLAP</b>	Multidimensional OLAP	
<b>MS</b>	Matrix Schema	
<b>NASDAQ</b>	National Association of Securities Dealers Automated Quotations	<a href="http://www.nasdaq.com/">http://www.nasdaq.com/</a>
<b>NCBs</b>	National Central Banks	<a href="http://www.ecb.europa.eu">http://www.ecb.europa.eu</a>
<b>NSIs</b>	National Statistical Institutes	<a href="http://epp.eurostat.ec.europa.eu">http://epp.eurostat.ec.europa.eu</a>
<b>OLAP</b>	On-Line Analytical Processing	
<b>OMG</b>	Object Management Group	<a href="http://www.omg.org">www.omg.org</a>

<b>NOMBRE</b>	<b>DESCRIPCIÓN</b>	<b>REFERENCIA</b>
<b>OWL</b>	Web Ontology Language	<a href="http://www.w3.org/TR/ow/-ref">www.w3.org/TR/ow/-ref</a>
<b>PIM</b>	Platform Independent Model	MDA
<b>POC</b>	Prueba de Concepto	
<b>PSM</b>	Platform Specific Model	MDA
<b>RDBMS</b>	Relational DBMS	
<b>RM</b>	Relational Model	
<b>ROLAP</b>	Relational On-Line Analytical Processing	
<b>SAX</b>	Simple API for XML	Parser XML
<b>SDMX</b>	Statistical Data and Metadata eXchange	<a href="http://www.sdmx.org">www.sdmx.org</a>
<b>SEC</b>	U.S. Securities and Exchange Commission	<a href="http://www.sec.gov">www.sec.gov</a>
<b>SEPA</b>	Single Euro Payments Area	<a href="http://www.ecb.europa.eu/paym/sepa/html/index.en.html">http://www.ecb.europa.eu/paym/sepa/html/index.en.html</a>
<b>SGML</b>	Standard Generalized Markup (ISO 8879-1986)	<a href="http://www.w3.org/Markkup/SGML">www.w3.org/Markkup/SGML</a>
<b>SPEM</b>	Process Engineering Metamodel	<a href="http://www.omg.org/spec/SPEM/2.0/">http://www.omg.org/spec/SPEM/2.0/</a>
<b>SQL</b>	Structured Query Language	Special propose programming language for relational database (RDBMS).
<b>SSE</b>	Shanghai Stock Exchange	<a href="http://english.sse.com.cn/">http://english.sse.com.cn/</a>
<b>SWIFT</b>	Society for Worldwide Interbank Financial Telecommunication/Sociedad para las Comunicaciones Financieras Interbancarias Internacionales	<a href="http://www.swift.com">www.swift.com</a>
<b>SZSE</b>	ShenZhen Stock Exchange	<a href="http://www.szse.cn/main/en/">http://www.szse.cn/main/en/</a>

<b>NOMBRE</b>	<b>DESCRIPCIÓN</b>	<b>REFERENCIA</b>
<b>T2</b>	TARGET2	<a href="http://www.ecb.europa.eu/paym/t2/html/index.en.html">http://www.ecb.europa.eu/paym/t2/html/index.en.html</a>
<b>T2S</b>	TARGET2 Securities	<a href="http://www.ecb.europa.eu/paym/t2s/html/index.en.html">http://www.ecb.europa.eu/paym/t2s/html/index.en.html</a>
<b>UD</b>	Universe of the Discourse	
<b>UML</b>	Unified Modelling Language	
<b>URI</b>	Universal Resource Identifier	
<b>URL</b>	Universal Resource Locator	
<b>US-GAAP</b>	United States-Generally Accepted Accounting Principles	
<b>WAN</b>	Wide Area Network/ Red de Área Extensa	
<b>XBRL</b>	eXtensible Business Reporting Language	<a href="http://www.xbrl.org">www.xbrl.org</a>
<b>XBRLDM</b>	XBRL Data Model	
<b>XDT</b>	XBRL Dimensional Taxonomies	XBRL Dimensional 1.0
<b>XLink</b>	XML Linking Language	<a href="http://www.w3.org/TR/xlink/">www.w3.org/TR/xlink/</a>
<b>XMI</b>	XML Metadata Interchange	Estándar para el intercambio de metadatos utilizando XML. <a href="http://www.omg.org/spec/XML/">http://www.omg.org/spec/XML/</a>
<b>XML</b>	eXtensible Markup Language	
<b>XPath</b>	XML Path Language	<a href="http://www.w3.org/TR/xpath20/">http://www.w3.org/TR/xpath20/</a>
<b>Xpointer</b>	XML Pointer Language	<a href="http://www.w3.org/TR/xptr/">www.w3.org/TR/xptr/</a>
<b>Xquery</b>	XML Query Language	<a href="http://www.w3.org/TR/xquery/">www.w3.org/TR/xquery/</a>
<b>W3C</b>	World Wide Web Consortium	<a href="http://www.w3.org">www.w3.org</a>

# Glosario

---

## A

### Aserción

Es una restricción general que hace referencia a un hecho o conjunto de hechos.

### Atributo Calculado

Es un agregado de atributos de dimensión y/o calculados.

### Atributo de dimensión

Son conceptos de una dimensión.

### Atributo por defecto

Cada dominio tiene un concepto por defecto.

## C

### Capital Requirements Directives (CRD)

Marco de la supervisión financiera de la Unión Europea, basada en las reglas de medición del capital y estándares de capital de Basilea II y Basilea III.

### Computation Independent Model (CIM)

Modelo de dominio del negocio. En este nivel, del MDA, se analiza el mundo real, incluyendo conceptos, datos y regla.

### Concepto

Es la definición de un concepto de negocio o ítem. Cada concepto está asociado a un atributo de periodo de tiempo (Instant, Period, y Forever).

### Concepto básico

Es un concepto especial que tiene asociado un tipo de dato, un tipo de período de tiempo y un tipo de balance.

### Constante

Es un valor o expresión que produce un resultado y cuya expresión es determinista.

## **COREP**

Se trata de un estándar emitido por la Autoridad Bancaria Europea (EBA), para los informes de la Directiva sobre Requisitos de Capital, que cubre el riesgo de crédito, riesgo de mercado, riesgo operativo, los fondos propios y ratios de solvencia.

## **D**

### **Dimensión**

Es un conjunto de conceptos de un dominio. Estos conceptos tienen una estructura en árbol.

### **Dimensión base**

Es un conjunto con solo conceptos básicos.

### **Dimensión explícita**

Es explícita si los atributos de dimensión están definidos.

### **Dimensión implícita**

Es implícita si los atributos de dimensión no están definidos.

### **Dimensión/atributos de dimensión y calculados**

Pueden definir un hecho o un conjunto de hechos.

### **Discoverable Taxonomy Set (DTS)**

Conjunto de taxonomías (archivos XML) que son necesarios para el procesamiento de un documento XBRL.

### **Document Object Model (DOM)**

Es una plataforma que permitirá a programas y scripts acceder y modificar el contenido, estilo y estructura de un documento.

### **Dominio**

Es un grupo de conceptos que pertenecen a un campo o espacio de conocimiento, o actividad. En este modelo un dominio puede contener conceptos básicos y no básicos, pero no ambos.

## E

### Eurofiling

La comunidad de Eurofiling es una empresa abierta conjunta de la *Autoridad Bancaria Europea* (EBA) y la *European Insurance and Occupational Pensions Authority* (EIOPA) en colaboración con *XBRL Europe*, así como depositarios como Bancos, proveedores de soluciones, universidades y personas individuales. <http://eurofiling.info/portal/>.

## F

### FallbackValue

Define un valor por defecto para un hecho o un conjunto de hechos, salvo que el valor de este atributo sea nulo.

### Filtro

Es un conjunto de dimensiones a las cuales la aserción está confinada.

### Filtro explícito

Es una restricción semántica, al cual todos los hechos, grupo de hechos o hechos de la aserción están circunscritos.

### Filtro implícito

Es una restricción inherente a todos los hechos circunscritos, por la aserción (por ejemplo, la aserción se restringe a los hechos circunscritos por el conjunto de pares inherentes dimensión-atributo de dimensión).

### FINREP

Cubre la información financiera consolidada y sub-consolidada a efectos de supervisión en base a las Normas Internacionales de Contabilidad (IAS) y el estándar de Informes Financieros Internacional (IFRS) y es emitido por la EBA.

## G

### Grupo de dimensiones

Grupo de dimensiones de un dominio.

### **Grupo de filtros**

Define un conjunto de filtros explícitos para el cual todas las aserciones están circunscritas.

## **H**

### **Hecho**

Un hecho es un valor que representa una medida particular provista por la entidad que hace el informe.

### **Hecho permitido**

Restricción de usuario.

### **Hecho prohibido**

Restricción de usuario.

## **M**

### **Model Driven Architecture (MDA)**

En ingeniería de software ofrece un buen marco para la generación automática de código para el desarrollo de aplicaciones. Se centra en el uso de modelos como enfoques para cubrir el ciclo de vida de desarrollo de software. Los problemas de heterogeneidad y la interoperabilidad entre los sistemas con diferentes plataformas de implementación se resuelven mediante el uso de este enfoque. La MDA estratificó el diseño en tres fases o niveles, para permitir el desarrollo más fácil.

### **Modelo de Datos Conceptual**

Este modelo muestra la semántica del mundo real, mostrando un conjunto de aserciones, a través de entidades y relaciones.

### **Modelo de Datos Multidimensional (MDM)**

Es un modelo sencillo que combina objetos, dimensiones (jerarquías), las medidas y los atributos para representar problemas de negocio reales de trabajo. Además, este modelo es el corazón de procesamiento analítico en línea (OLAP), el cual exige consultas complejas que pueden ser resueltos por el MDM en tiempo real.

## **P**

### **Platform Independent Model (PIM)**

Este se centra en la lógica de negocios de alto nivel en el MDA, sin tener en cuenta las características de la tecnología de implementación del sistema.

### **Precondition o atributo de la aserción**

Define los hechos que pueden ser evaluados.

### **Platform Specific Model (PSM)**

Representa el detalle de la utilización de una plataforma específica para un sistema en el MDA.

## **R**

### **Referencias**

Referencias a directivas o leyes de los conceptos.

### **Restricción jerárquica**

Los conceptos en una dimensión tienen una estructura en árbol. Se utiliza en las validaciones entre una hoja raíz y sus hojas, es decir para los atributos calculados.

## **S**

### **SAX (Simple API for XML)**

Es un algoritmo en tiempo de ejecución orientado a eventos para analizar documentos XML, con una interfaz API desarrollada por la lista de correo de XML-DEV.

### **Solvency II**

Es una directiva de la Unión Europea (UE), que codifica y armoniza las regulaciones de seguros de la UE y que se emitió por la Autoridad Europea de Seguros y Pensiones de Jubilación (EIOPA).

## V

### **Variable**

Es un elemento atómico de una expresión en una aserción el cual conlleva un hecho o un grupo de hechos.