



Universidad
Carlos III de Madrid
www.uc3m.es

DOCTORAL THESIS

Framework for multidimensional definition of the data model, structure, taxonomies and rules of the XBRL specification

AUTHOR:

Ignacio-J. Santos Forner

DIRECTOR:

Dra. Elena Castro Galán

CO-DIRECTOR:

Dr. Manuel Velasco de Diego

COMPUTER SCIENCE DEPARTMENT

Leganés, December 2015



Universidad
Carlos III de Madrid
www.uc3m.es

DOCTORAL THESIS

Framework for multidimensional definition of the data model, structure, taxonomies and rules of the XBRL specification

Author: Ignacio-J. Santos Forner

Director: Dra. Elena Castro Galán

Co-director: Dr. Manuel Velasco de Diego

Signature of Tribunal's decision:

Signature		
Chairman		
Tribunal' member		
Secretary		

Mark:

Leganés,

2015.

Acknowledgements

In Regulation and Supervision, an important agreement is *Basel II* (recommendations for banking and regulatory law). This is based on three pillars that are, in summary:

- Minimum capital requirements (obtaining data from the supervised entity).
- Supervision and review of data from capital requirements (exploiting these data).
- Market discipline, i.e. making the obtained and supervised data public.

Using the analogy of the three pillars of Basel, in this doctoral thesis there are four essential people (two of them represented in the first pillar), without whom I could never have written this thesis. These people are:

- My tutors, Manuel Velasco and Elena Castro, especially Elena Castro, who has made a superhuman effort to understand the semantic domain of economic reports, which is very close to the world of finance, but also a huge effort to understand me, because I'm not a young student who recently finished university studies.
- Ignacio Boixo with XBRL Spain and the Bank of Spain, who allowed Elena and me all the necessary means of support and infrastructure to develop this thesis.
- My secretary, Aurora Forner, with infinite patience, working tirelessly, writing all the documents in *Word* (in Spanish and English), creating all the drawings in *PowerPoint*, classifying the documents, mail, etc. She has done excellent and brilliant work at the computer. At this point, I will not forget my brother, Félix Santos, nor my father, Ignacio Santos, who worked a lot for his children.

There are many more people who have supported me, and I hope not to forget anyone. Examples of these contributions are: Víctor Morilla from the Bank of Spain, Moira Lorenzo of Atos and Santiago Carrasco of Indra. I would also like to highlight María José Gallego, colleague at the Bank of Spain, for her huge work on the editing and layout of this dissertation, effort on which she has spent more than one sleepless night.

During the development of this thesis four final graduate works have been conducted so it would be unthinkable not to acknowledge the efforts they have made in such a technical and specialized subject. These four students have been, in chronological order:

- Jorge Valencia Martínez, *Modelo de Meta-Metadatos XBRL (Model of XBRL Meta-metadata)*, accessible at <http://hdl.handle.net/10016/13157> (Valencia, 2011).
- Yolanda León Román, for her final work entitled: *Automatización del Modelo de Datos de la Especificación XBRL a Base de Datos (Automating the Data Model of the XBRL specification in a database)*, accessible at <http://hdl.handle.net/10016/16843> (León, 2012).
- Camelia Cid Canelada, *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 (Road map of the MDM (implemented in a design ROLAP) of FINREP 2012 and Solvency II (01-07-2012) with validations, Proof of Concept.)*. In progress (Cid, 2013).
- Abel Nieto Cano, *Validación de los informes semánticos económicos financieros (Validation of the semantic economic financial accounting reports)* (Santos and Nieto, 2014, 2015; Nieto, 2015).

Last but not least, I cannot conclude this section without recording my deepest gratitude to Dr. Dolores Cuadra and Dr. Harith Al-Jumaily, for their involvement and contribution. Without your support I had not reached so far. Thanks.

Index

Acknowledgements	i
Index of Algorithms.....	v
Index of Codes	vii
Index of Definitions.....	ix
Index of Figures	xi
Index of Rules	xv
Index of Tables.....	xvii
Summary.....	1
CHAPTER 1. Introduction.....	3
1.1. Motivation	3
1.2. Aims.....	4
1.3. Document Structure	7
CHAPTER 2. State of the Art	11
2.1. Introduction	11
2.2. Evolution	14
2.3. Current and future state	17
2.4. Design of semantic reports.....	20
2.5. Model Driven Architecture	21
CHAPTER 3. Model of meta-metadata of the specification.....	25
3.1. Introduction	25
3.2. Models and XML objects	29
3.3. Model of the XBRL meta-metadata	31
3.4. XML and XBRL objects used in the definition of the meta-metadata	33
CHAPTER 4. Fundamentals and Basic Concepts of the proposal	55

4.1. Universe of the Discourse and the proposal	55
4.2. The Computation Independent Model (the CIM)	57
4.3. The Platform Independent Model (the PIM): Rules and Definitions	60
4.4. Semantic Questions about the Design of Metadata in the PIM	72
4.5. Analysis in the Platform Specific Model (PSM)	73
CHAPTER 5. Fundamentals and Basic Concepts of the proposal	79
5.1. The Universe of the Discourse	79
5.2. XBRL Data Model and calculations	81
5.3. Introduction to XBRL formula specification	82
5.4. Definitions and semantics of XBRL formula specification	86
5.5. Semantic questions of the XBRL formula specification	97
5.6. Implementation and proof-of-concept (POC), in the PSM	100
CHAPTER 6. Validation of the proposal	109
6.1. Introduction to the validation	109
6.2. Validation of the development of the metadata of the semantic reports	109
6.3. Validation of calculations using XBRL formula specification	123
CHAPTER 7. Conclusion and future work	139
7.1. Conclusions	139
7.2. Future works	141
Bibliography	145
Acronyms	157
Glossary	165

Index of Algorithms

Algorithm 4.1 Transformation from the CIM in the PIM	70
Algorithm 4.2 Implementation of the metadata model	75
Algorithm 5.1 Creation of SQL templates	104
Algorithm 6.1 Extraction of the metadata model	110

Index of Codes

Code 5.1 Template I (SQL I), a summary of the code.....	106
Code 6.1. Creation of the tables of the star model (<i>Nieto, 2015</i>)	116
Code 6.2. Cycles in the hierarchy of concepts are searched for (<i>Nieto, 2015</i>).....	117
Code 6.3. This trigger creates the tables of dimensions and loads the dimension attributes into the dimensions (<i>Nieto, 2015</i>)	118
Code 6.4 SQL script that creates the tables in the star model	126
Code 6.5. Formula <i>val_6610-sh-2</i> obtained from Arelle (<i>Figure 5.6</i>)	127
Code 6.6 Procedure of validation of the assertion <i>val_6610_sh_2_1</i>	133

Index of Definitions

Definition 4.1	58
Definition 4.2	59
Definition 4.3	62
Definition 4.4	62
Definition 4.5	62
Definition 4.6	63
Definition 4.7	63
Definition 4.8	64
Definition 4.9	64
Definition 4.10	64
Definition 4.11	65
Definition 4.12	65
Definition 4.13	66
Definition 4.14	66
Definition 4.15	67
Definition 4.16	67
Definition 4.17	68
Definition 4.18	68
Definition 5.1	87

Definition 5.2	87
Definition 5.3	88
Definition 5.4	88
Definition 5.5	88
Definition 5.6	88
Definition 5.7	89
Definition 5.8	89
Definition 5.9	90
Definition 5.10	91
Definition 5.11	93

Index of Figures

Figure 2.1 Example of transmission of data of an ATM	11
Figure 2.2 Example of semantic accounting report.....	14
Figure 2.3 Tour of a financial report.....	17
Figure 3.1 Structure of the XBRL specification	26
Figure 3.2 Structure of an <i>XBRL Instance Document</i>	27
Figure 3.3 The data model of XBRL	28
Figure 3.4 Structure XML versus XBRL.....	32
Figure 3.5 Type XBRL <i>contextScenarioType</i>	38
Figure 3.6 Element XBRL <i>linkbase</i>	40
Figure 3.7 Element XBRL <i>schemaRef</i>	41
Figure 3.8 Element XBRL <i>context</i>	42
Figure 3.9 Example of the element XBRL <i>segment</i>	43
Figure 3.10 Element XBRL <i>dimension Item</i>	45
Figure 3.11 Element XBRL <i>hypercubeItem</i>	46
Figure 3.12 Element XBRL <i>xbrl</i>	47
Figure 3.13 Element XBRL <i>explicitMember</i>	50
Figure 3.14 Element XBRL <i>typeMember</i>	51
Figure 4.1 Design of semantic reports using the MDA paradigm	56
Figure 4.2 Example of report with semantic.....	58

Figure 4.3 Sample of template	59
Figure 4.4 Star model in the PIM.....	69
Figure 4.5 Example of fact table with calculated attributes	71
Figure 4.6 Summary of the artefacts in UML of the data model in the PIM	72
Figure 5.1 General Structure of the transformation XBRL vs. RDBMS	80
Figure 5.2 Example of hierarchical report.....	83
Figure 5.3 Graphic design of the report of the example of <i>Figure 5.2</i>	84
Figure 5.4 Hierarchical structure of the report of <i>Figure 5.2</i>	85
Figure 5.5 Automatic conversion of formulas between XBRL instance documents and Databases using MDA	86
Figure 5.6 Example of the assertion <i>val_6610sh-2</i>	95
Figure 5.7 Example of the assertion <i>val_6610sh-2</i> presented graphically	96
Figure 5.8 Example with null fact.....	98
Figure 5.9 Multivalued facts.....	100
Figure 5.10 Obtaining the validation API	102
Figure 5.11 Validation process	103
Figure 5.12 Assertion process of validation-process-generating algorithm	103
Figure 6.1 FINREP 2012, statistical data in the MDM.....	112
Figure 6.2 Validation of the metadata.....	113
Figure 6.3 Flow of IS_DIMENSION data. Execution of the fact table creation SQL procedure (<i>Nieto, 2015</i>).....	114
Figure 6.4. Diagram of the database in the MDM of FINREP 2014 (<i>Nieto, 2015</i>) .	119
Figure 6.5. Diagram with the context in the star model.	124

Figure 6.6 Validation test for 6610 report..... 134

Index of Rules

Rule 4.1.....	60
Rule 4.2.....	64
Rule 5.1.....	89
Rule 5.2.....	90
Rule 5.3.....	90
Rule 5.4.....	90
Rule 5.5.....	90
Rule 5.6.....	91
Rule 5.7.....	91
Rule 5.8.....	91
Rule 5.9.....	92
Rule 5.10.....	92
Rule 5.11.....	93

Index of Tables

Table 4.1 Definitions and rules in the MDM (the PIM).....	61
Table 4.2 Mapping from the PIM to the PSM.....	74
Table 5.1 From the CIM to the PIM	93
Table 6.1 Validation tests belonging to the UML star model	121
Table 6.2 Validation tests belonging to the relational model (RM)	122

Summary

Since 2000 there has been a significant development in the number of applications based on XML and Data Warehouses. As these types of applications need to interact with different technologies and other applications, **interoperability** is an important factor. In parallel, in the same period there has been an evolution of the economic data needed by companies and financial institutions, increasing the number of entities that need more information, but also requiring this information to be reliable and on time. That economic or financial information is reliable does not refer to encoding or transmission errors, since there were already enough reliable procedures for this in 2000, but that these reports have to be semantically correct. It is in this decade that the need for economists and accountants to obtain their reports with semantic content has been observed. For this reason some specifications for the transmission of economic information have arisen, such as the *eXtensible Business Reporting Language* (XBRL) and the *Statistical Data and Metadata eXchange* (SDMX). The reports of both specifications have semantic content and are dedicated to sending and processing financial and / or statistical information. However, both specifications were created by expert economists or statisticians, and their only purpose was to solve their problems without formalizing their definitions and conceptual rules, because there is a Conceptual Model and the Logical Model created is very close to the Physical Model, that is to say, to its implementation.

The aim of this thesis is to formalize the specification of economic financial reports using the XBRL specification, analysing and auditing that specification. In addition, the design of these reports is studied in depth by making use of the methodology of the *Model Driven Architecture* (MDA), with the ultimate goal of analysing its interoperability. Finally, some proofs of concept are shown, with different validations that are the proposal of the thesis.

The thesis begins by analysing the necessity of this specification (XBRL), its historical origin, and the general state of the art that has led to the use of it as a *de facto standard*. In addition, this research work studies its evolution and utilization, both historically, as it is today, and its expected future.

Once the state of the art of economic financial semantic reports has been analysed, the next step is to analyse the XBRL specification. After giving an overview of the XBRL language or specification, the thesis begins to detail it and finally shows the set of meta-metadata used in XBRL, using UML.

This thesis proposes designing both the metadata of the reports and the reports themselves. This research work also proposes to study the different implementations of these reports and their design in totality. MDA provides a good consolidated framework for automatic code generation in the life cycle of software development, in addition to solving the problems of heterogeneity and interoperability between systems with different platforms.

One of the main challenges of this research is the formalization of the specification in the conceptual model. Different types of design are also discussed, including those already making intensive use of dimensions (European design) and making less use of dimensions (mainly US and Canadian design). This work is mainly based on the reports of the European Supervision - Regulation, thus making intensive use of dimensions. This is due to the fact that the author of this thesis belongs to various groups that deal with harmonised European supervision reports and is a civil servant of the Bank of Spain, and this institution is part of the *European System of Central Banks* (ESCB), and this in turn through the *European Central Bank* (ECB) and the *European National Central Banks* (NCBs) participates in the *Eurosystem* (monetary Authority in the Eurozone (*Eurosystems*)). Therefore the conceptual model proposed in this research is the *Multidimensional Data Model* (MDM). The proposal is based on reports defined for the European Supervision/Regulation. These reports are the data requested of financial institutions or supervised companies, being the *Universe of the Discourse* (UD), also called the *Computation Independent Model* (CIM), using the methodology of the MDA.

Once the definitions are fulfilled the next step is analysing the processes of validation of this proposal, showing algorithms for the MDM and the process of generating assertions, or formulas in the *XBRL Data Model* (XBRLDM) data. In addition, its implementation is displayed in the logical and physical models.

The entire proposal is supported by different *Proofs of Concept* (POCs) submitted in specialized forums of the European financial and economic supervision. These tests are based on three European taxonomies: *Financial Report 2008 (FINREP)*, 2012

and *Solvency II* (Eurofiling 2015). *FINREP* taxonomy is sponsored by the *European Banking Authority* (EBA). *Solvency II* codifies and harmonizes the regulation of insurance in the *European Union* (EU) and is sponsored by the *European Insurance and Occupational Pensions Authority* (EIOPA). In these POCs all the proposals of this thesis are shown and analysed. In addition, tables show the different tests that show the validation of this proposal at the various levels of the MDA methodology.

In addition, the interoperability of economic financial reports, and their need for mapping XBRL, spreadsheets, Access, Database Management System (DBMS), XML and others are analysed.

Finally, research work that has to be done in the immediate future as a result of this thesis is proposed.

CHAPTER 1

Introduction

CHAPTER 1. Introduction

1.1. Motivation

Semantic economic and financial reports were born due to the need for regulators and supervisors of information from companies and financial institutions for a reliable, real-time and easily treatable format for other applications. These reports were designed by economists, accountants, or financial experts, according to their needs, and computer engineers took care to solve their strict requirements.

From these needs arose what would become the XBRL specification (*Hamscher and Kannon 2000; Engel et al 2008*), which has become a *de facto* standard. When it was concluded that this specification did not meet all the needs of expert users' calculations, the specification was extended (*Hernández and Wallis 2006; Engel et al., 2006*). As other needs of Supervisors or Regulators were addressed, the specification continued to grow. At present, it is actively used by the *Board of Governors of the Federal Reserve System (FED)*, the *Securities and Exchange Commission (SEC)*, the *ShenZhen Stock Exchange (SZSE)*, the *Shanghai Stock Exchange (SSE)* (*JiMei et al., 2012; Jimei et al., 2013*), the *European Central Bank (ECB)*, the *European Banking Authority (EBA)*, the *European Insurance and Occupational Pensions Authority (EIOPA)*, the *Deutsche Börse*, the *Deutsche Bundesbank*, *Companies House and HM Revenue & Customs (UK)* and the *Australian Prudential Regulation Authority (APRA)*, among many other institutions and agencies. In the USA, Canada, Europe, China, etc. all financial entities and companies quoted on the stock market have to report compulsorily to the supervisory and regulatory authority using the XBRL specification. Financial statements are governed by strict requirements, such as the *International Financial Reporting Standard (IFRS, 2015)* or *Generally Accepted Accounting Principles (GAAP)*. XBRL is used instead of XML, because the reports have specific semantics managed by IFRS and/or the GAAP, and these reports can be very large.

All data from these reports are stored in a database manager (DBMS) and it is therefore important that the mapping is easy, optimal and automatic. However, it is equally important that the design of these reports is specified according to

standardised data design models. In this latter respect, the European Union through the *Eurofiling* (2015) project is making efforts that will be mentioned later in this paper.

On the other hand, when the reports are very large (*XBRL reports* or *XBRL instance documents*), it is almost impossible to implement them using existing XML technology, especially for the validation of the documents.

At present, the *Information Technology* (IT) departments of the supervisory/regulatory bodies have huge difficulties because taxonomies are created with a lot of concepts (*Weller, 2015*) and user rules or restrictions (validations) and many have design errors and no public test cases. This is due to the lack of a complete software development life cycle, a lack of interoperability (*Santos and Castro 2010*) and in turn the fact that supervisors and/or regulators are continually creating larger and more complex models.

Although use of the specification is spreading, various problems do arise that restrain its universal use. It is a specification defined by economists and financiers, implemented in XML and is some respects complex, in scope of knowledge required and infrastructure.

1.2. Aims

Summing up the problems mentioned above:

- These reports are designed almost exclusively by economists, accountants, or financial according to their needs.
- The specification is extended according to expert user's requirements for new elements.
- The specification has become larger and more complicated.
- The specification is currently very actively used.
- Financial statements are governed by strict requirements.
- XBRL is used instead of XML, since reports have specific semantics of the IFRS or the GAAP.

- These reports can be large, since taxonomies are created with many concepts and user rules or constraints.
- When the reports are very large, this generates problems in implementation (especially in performance at run-time) using existing XML technology, especially in the validation of the documents.
- Many reports present design errors.
- No public test cases.
- Lack of interoperability.
- Supervisors and / or regulators are continually creating larger and more complex models.
- Complexity of the specification in knowledge as well as in infrastructure.

This thesis will attempt to solve the problems mentioned by means of analysis of the design of such independent platform reports, using the *Model Driven Architecture* (MDA, 2015) paradigm. The MDA stratifies the design into three phases or levels to allow for easier development. The levels of the MDA are: The *Universe of the Discourse* (UD), which is the set of accounting definitions, European directives, circulars, etc. that are analysed. This UD using the MDA paradigm is the CIM (*Computation Independent Model*). From the CIM we will advance to the PIM (*Platform Independent Model*). The PIM shall define all the concepts and rules of users in a sufficiently large and formal model for the transformation to the PSM (*Platform Specific Model*) in order to ensure that it is as accurate and correct as possible, given that this transformation is as automatic as is possible. Several independent platforms such as the DPM (*Data Point Model*) used by the IT departments of some regulatory and supervisory agencies, and the more universal SQL standard, will be analysed. Finally, the encoding rules are displayed, *Code* in the MDA paradigm. In addition, it is necessary to analyse various semantic issues that are important to keep in mind, because they can be intrinsic to the model or generate errors, inheriting these difficulties from PIM to PSM. This study will avoid errors in the business logic or in execution. In short, the application of this methodology ensures that the reports are well-constructed according to the structure and semantics previously defined, and therefore avoids making erroneous reports, ensuring the reliability, confidentiality and consistency of the report.

Furthermore, this study will analyse the interoperability of the models and even physical objects generated, so that from an XBRL taxonomy an implementation in MDM and in a RDBMS can be obtained, using the MDA paradigm.

Moreover, it must show the development life cycle of the whole semantic metadata reports, as well as the reports themselves.

In addition, all the research involved in this thesis will be accompanied by a set of conferences and magazine articles, both academic, as related to the domain expert users of XBRL reports. This thesis will be accompanied by a set of proofs of concept that must validate this approach.

This research will seek to clarify and simplify the complex XBRL specification, created by expert economists and accountants. In turn, the research will put a target on the reduction of infrastructure and development costs for these semantic reports, troubleshoot performance, and extend the field so that it can be used, for example, in the calculation of a presale, sale, and availability of the products of a company.

To accomplish these objectives, it is necessary to analyse the domain with expert users, and to use a paradigm wide enough for the development of life cycle software, such as the DMA paradigm, and to use sufficiently accurate models for the transformation to the MDM. The MDM must be defined with formal definitions, so that its processing, depending on the platform, can be as accurate as possible.

In short, for the achievement of this work, the following partial objectives should be covered:

- Create a software development life cycle for semantic economic / financial reporting, formalizing the concepts and user constraints or rules in the PIM, and the transformation rules at different levels of the MDA paradigm. All this makes use of the robustness provided by the multidimensional modelling technology.
- When creating a software development life cycle is intended to solve the problems of new extensions and / or software modifications.
- Automatic implementation in SQL, since the final destination of the data is a database, to allow its use by agencies and companies.

- Defining transformation rules to resolve many of the problems of interoperability presented to users of such reports.
- As these reports are automatically downloaded into a database (RDBMS), they can easily be used by business intelligence tools, and overall data analysis.
- Analyzing semantic issues or anomalies can resolve errors in design and build test suites.

1.3. Document Structure

After this brief description:

Chapter 2 will address the state of the art of semantic economic financial accounting reports, what their evolution has been, as well as their current and future state.

Chapter 3 will discuss the model of meta-metadata within the XBRL specification.

Chapters 4 and 5 will discuss the proposal:

- **Chapter 4** will also analyse the specification by itself for the design of these reports, but without any treatment of the formulas specification.
- **Chapter 5** will focus on the design and the process of the automation of calculations on these reports.

Chapter 6 will attempt to validate this proposal.

Chapter 7 will show the conclusions and future work of this proposal.

CHAPTER 2

State of the Art

CHAPTER 2. State of the Art

2.1. Introduction

The XBRL specification allows the standardization of accounting and financial information broadcasted electronically. Up to the 2000s this type of information was done on paper or in electronic form, but without semantic content. If information was issued electronically, it was through messages of varying size. In general, these messages were economic transactions. The format of these messages could be tabular or separated by a delimiter, for example by a ";". *Figure 2.1* shows an example of data from an ATM of a financial institution to a central computer of the bank (the example shown is a simplification, it is not real, it is only an approximation).

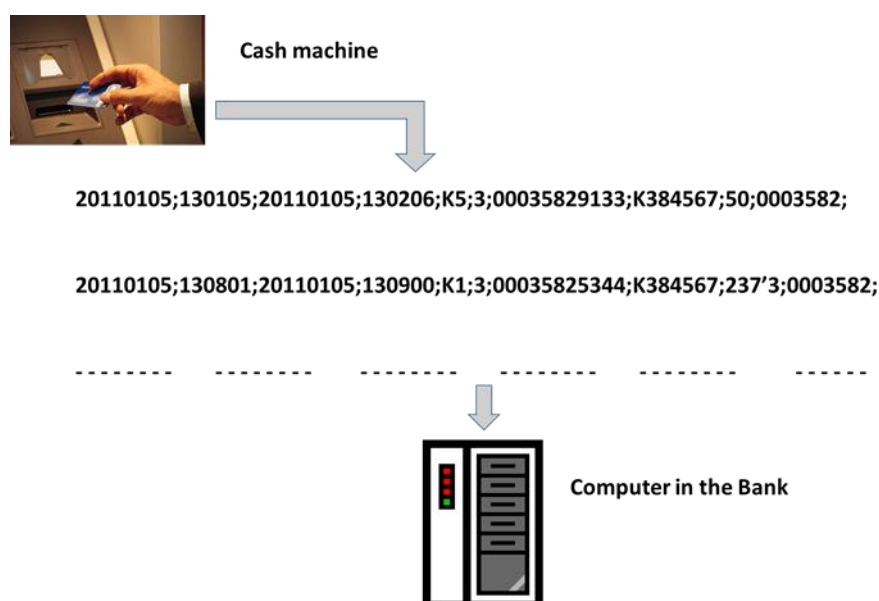


Figure 2.1 Example of transmission of data of an ATM

This example shows two transactions, in the first a customer withdraws €50 from his current account, the number of the current account is 0035829133, the K5 code means to withdraw money. After this code is the result of code operation 3 (which may mean transaction accepted). In this transaction the date and time of its beginning and end are indicated. In addition, there is a cashier code (K384567) and the entity indicated 0003582. The order of the second transaction indicates the obtaining of the

balance of the current account code K1 with a success code 3 indicating that the account has €237.3.

In a company, organization or agency, there is always an exchange of accounting or economic reports. Since the late 1990s, this exchange of reports (which are defined as messages in *Information Systems*, or IS) has started to increase. Companies needed to know the status of their orders as soon as possible, and to perform a calculation of the presale, sale and future product availability (Lee *et al.*, 1997; Wagenhofer, 2003). Until then, the exchange of messages was done via flat files, and with pre-Internet communication protocols. This information was a set of records, where each record or line consisted of a set of characters, separated by *commas*, *semicolons* or in *columns*. This set of transmitted data does not have any more semantics than the application itself that uses the message. The application will give itself for valid received messages, except syntax problems, such as a strange code appearing in the account code, which will prompt the message relay application to cancel it.

In 1996 the protocol *Electronic Data Interchange* (EDI) was formulated. EDI implies a sequence of messages amongst computers. This protocol is very widely used in purchase orders, invoices, shipping notices, and so on. These messages simulate the exchange of documents on sheets of paper.

Another protocol that is very widely used worldwide is SWIFT (*Society for Worldwide Interbank Financial Telecommunication*; <http://www.swift.com>). This company started working in 1973 with the sponsorship of 293 financial institutions. This platform initially used the X.25 protocol (Standard of the *International Telecommunication Union*, ITU, using in the *Wide Area Network*, WAN) and a messaging service called FIN (*Financial message*). FIN is based on ISO 20022 (2015) (*Standard for Financial Services Messaging*). This standard consists of a repository of metadata, where the messages are described, as well as the business processes and the maintenance of the repository. This standard is very widely used in agencies of financial services such as *FIX Protocol Limited* (*Financial Information eXchange*, FIX), and used among others for NASDAQ (*National Association of Securities Dealers Automated Quotations*) on the American Stock Exchange. Also, this protocol is used in ISDA (*International Swaps and Derivatives Association*), which is an organization of derivatives trading; this organization invoices billions of dollars on NASDAQ, in addition to SWIFT or VISA (*Visa Inc.*). Since 2003, SWIFT has used the protocols IPsec (*Internet Protocol*

Security) and *Interfact* (a protocol to exchange automated and interactive messages, synchronous or asynchronous). Consequently, with this new version, FIN works with applications-applications (A2A) and users-applications (U2A).

Currently, it is possible to find short messages with little semantic information or long messages or reports that need more semantics (*Hodge et al. 2004; Lin et al. 2005; Kernan 2009; Debreceeny et al. 2010; Prasanna et al. 2012*). A typical semantic report might be the authorization of a loan, in which certain guarantees are necessary, such as financial assets, guarantees, quoted securities and the authorization of the granting of the loan. This is equivalent to a set of messages or reports related with business semantics. Examples of these include SEPA (*Simple Euro Payment Area, 2014*), T2 (*Target 2, 2013*) or T2S (*Target 2 Securities, 2014*), used for payments and transfers, large payments and securities transactions respectively.

According to *Woodroof and Searcy (2001)* a system of economic-accounting should have integrity, security, availability and maintainability. Moreover, it should have good business practices, integrity in the transactions, protection of the information and certainty in issuance of the report (AICPA / CICA, 1999). If these reports are not semantic, they cannot be directly automated in the internal or external processes of the company through IS (*Wagenhofer 2003; Williams et al., 2006*).

Consequently, the sending and receiving of financial information is perfectly resolved. Furthermore, this information can be generated, received and processed directly by the applications involved.

However, from the early 2000's, economists found they did not have enough with the technology developed for the transmission of economic messages, they needed semantics in these reports. In addition, when reports are too large, they often have a strong semantic content, and by the 2000s no solution for this had been created by Information Systems (IS). For example *Figure 2.2* shows a report with the settlement accounts of a local authority to the tax agency (*Inteco 2009*).

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
TOTAL	20.000,00	4.000,00

Figure 2.2 Example of semantic accounting report

In this report the data are interrelated, and therefore may be established some semantics. For example the total for the obligations should be the sum of the items 1 to 7, if not, is that the total is incorrect. However an accounting report can become much more complicated and with a complex semantics. Since concepts can be related to each other, depending on each other. This thesis analyse the semantics of the economic / financial reports based on XBRL specification.

2.2. Evolution

Since the year 2001, stock market regulators began to demand the reporting of much more business information and reduced the amount of time in which this reporting was to occur. While this swap of information may be done in paper, FAX or e-mail, all such reports are nevertheless becoming more complex, requiring more in-depth analysis and processing. Although for years there have been several solutions available for transmitting information, none are adequate from the viewpoint of business economics. Given these requirements, the information included in such statements must be semantically well-built, free of syntax errors and must be prepared for immediate or subsequent processing and storage.

The *eXtensible Business Reporting Language* (XBRL), an XML-based standard, is a specification used to exchange financial/economic information. The world's main economic/financial institutions and agencies, as well as many companies and state or local agencies, actively use this specification. The use of this specification is increasingly important. In the USA, Canada, Europe, China, etc. all financial entities and companies quoted on the stock market have to report compulsorily to the supervisory and regulatory authority using the XBRL specification. XBRL is actively used by the Board of *Governors of the Federal Reserve System* (FED), the *Securities and Exchange Commission* (SEC), the *ShenZhen Stock Exchange* (SZSE), the *Shanghai Stock Exchange* (SSE) (JiMei et al., 2012; Jimei et al., 2013), the *European Central Bank* (ECB), the *European Banking Authority* (EBA, 2014), the *European Insurance and Occupational Pensions Authority* (EIOPA, 2014), the *Deutsche Börse*, the *Deutsche Bundesbank*, *Companies House* and *HM Revenue & Customs* (UK) and the *Australian Prudential Regulation Authority* (APRA), among many other institutions and agencies.

Financial statements are governed by strict requirements, such as the *International Financial Reporting Standard* (IFRS 2015) or in Spain, the *General Chart of Accounts Plan* (Spanish Official Gazette, 2007). Thus, financial statements of credit institutions, for example, are specific statements defined by one or more taxonomies, including their structures and semantics. As these financial statements vary with time by decree (national level) or circular (Bank of Spain) or directives (European Union), involve problems of versioning and localization. The various laws or policies change over time and therefore reports must adapt to these new regulations, and therefore must maintain a versioned. The location means that depending on where or who order this report, the accounting rules vary, so for example the countries of *European Monetary Union* (EMU), or a country. Then the design problems of these taxonomies (set of concepts and rules) rise significantly when they are designed together. For just give an example, while the accounting requirements of capital flows, until very recently were based on Basel II (2004) as a result of the current financial crisis, these regulations have been modified in Basel III (2010) (The Larosière Group (2009)). Besides, the large amount of requirements and data imply a high cost in resources from IS in many companies, financial and economic institutions.

Following the bankruptcy of *Enron Corporation* in December 2001, the stock market regulators begin to demand more information on the reports sent to the supervisor in the US (*Securities and Exchange Commission*, SEC 2015). But in turn, they wanted

to reduce the processing time of these reports. The SEC has a system called EDGAR (*Electronic Data Gathering, Analysis and Retrieval System*, <http://www.sec.gov/edgar.shtml>) to obtain information in electronic format and recovery analysis, though these files collected, which were financial statements, were in any format, PDF, or scanned. At presents, EDGAR collects reports in XBRL. Therefore, it is required that this information is free of errors, secure in its delivery and easy to process. In addition, companies and financial institutions are increasingly in need of exchanging economic / financial information. While this exchange of information was done in paper, fax or email, the process of incorporation into a database was slow and therefore the analysis was slower. Already in 2000 there were enough IT solutions for the transmission of information, but to date, none was appropriate, from an economic and accounting perspective, because this information has not contain semantic content, which was what was required.

In April 1998, the accountant and auditor, Charles Hoffman, proposed the automation of the exchange of financial information, developing a prototype for accounting and auditing, known as the base of XBRL (*Hamscher and Kannon, 2000*). XBRL is an XML-based standard for semantic financial reporting (*Engel et al., 2008*). The reports XML or XBRL instance documents are generated from several sources and validated at source, ensuring its syntactical validation. In *Figure 2.3*, the report is filled in by the user, before being sent, it is validated and if the validation is successful, the report is sent for processing in destination.

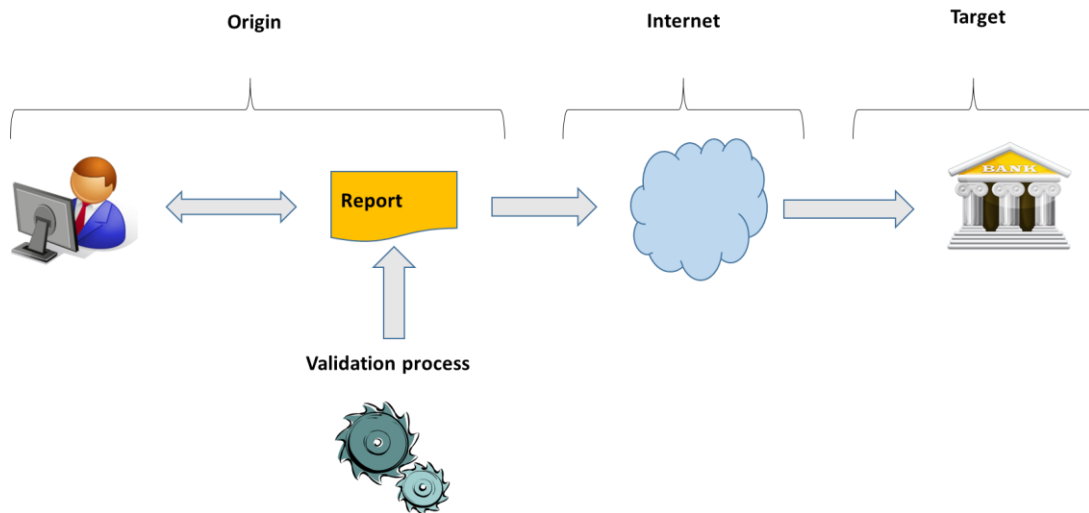


Figure 2.3 Tour of a financial report

As the XBRL specification represents financial information, which is multidimensional, the logical location for storage is a *Data Warehouse (DW)* (Golfarelli et al., 2001) (Boixo and Flores, 2005). Given the requirements of financial institutions, Regulators and Supervisors, in general, this information must be semantically well-built, free of syntax errors (Debreceeny et al. 2010; Zhu and Fu 2009; Gräning et al. 2011; JiMei et al 2012; Jimei et al 2013 Nickerson et al. 2013). In addition, these reports must be prepared for immediate IS process and its subsequent storage in a DW. Adding further complexity to this task, the analysis of the information needed for a particular purpose or general business often requires additional information contrasted with the report received (Callaghan et al 2002; Di Giovanni, and Piazza, 2009; Loukis, and Charalabidis 2013). Experts users, supervisors, regulators, companies generally need to analyse the information received from various sources; it is therefore important that the reports are interoperable (Bennett 2013). In this latter respect when working with different specifications or standards will be necessary to analyse the semantic negotiation between different processes (Buder et al. 2009).

2.3. Current and future state

The current use of this specification is widespread within the sector of the Supervision and the Regulation in almost all world. However, little by little they are adding other

sectors, although generally related to institutions and public agencies. This is due to the high cost of implementation, both in human resources, and software, as it requires experts in the specification and software. This thesis examines the development and implementation of semantic financial economic reports, for it will analyse the XBRL specification, formalizing their definitions, rules and constraints, studying their interoperability, in order to extend the use of the specification, lowering these costs, by making and widespread, because it is used a general and known technology.

Another specification for semantic reports is the *Statistical Data and Metadata eXchange* (SDMX 2015). This is often used by the *European Central Bank* (ECB) and the *Bank for International Settlements* (BIS), among other agencies and institutions. This standard is mainly used for the exchange of statistical data and metadata among institutions and for the dissemination of statistical information. It is a specification developed by international institutions, *National Central Banks* (NCBs) and *National Statistical Institutes* (NSIs), and not specifically adapted for other sectors. So, its semantic analysis is out of scope of this thesis.

Computer engineers of European banking supervision saw the need for a design model for European taxonomies. Since 2010, BR-AG and the *Bank of Spain*, using reverse engineering started using the *Data Point Model* (DPM) (*Eurofiling* 2015). This model started using to the taxonomies of *Balance Sheet Items and Interest Rates Monetary Financial Institutions* (BSI-MIR 2010; Bank of Spain 2010) and they were implemented by the Polish financial software company, BR-AG. After, with this model was developed COREP (it focuses on the consolidated, sub-consolidated and solo reporting of capital requirements and capital and reserves based on EU directives) and FINREP (consolidated and sub-consolidated financial reporting for supervisory purposes based on IAS (*International Accounting Standards*)/IFRS) taxonomies (*Eurofiling* 2012). When this taxonomy is obtained, this was mapped to an *Excel* spreadsheet, but in DPM format and then was analysed by business users (the Regulators) to validate the taxonomy. Two years later a generator of *Excel* templates, called *Data Point Modeller* (ExGen 2012), from the XBRL taxonomies was developed. Then, using the same methodology, the DPM was developed two taxonomies for respectively *Cayman Islands Monetary Authority* and *Bermuda Monetary Authority*.

In 2013 the *Comité Européen de Normalisation/European Committee for Standardization* (CEN 2014) has been working on the standardization of design data model reports financial accounting in Europe. In the CEN was created a group that

has prepared a series of *CEN Workshop Agreements* (CWA 2014) which promulgated deliverables for a base of standardization in Europe. This group is divided into three subgroups: *Standard CWA1 harmonization topics*, *CWA2 Metadata container and compliance tools* and *CWA3 Standard Regulatory roll-out package for adoption*. The group CWA1 had as work to create the metadata model for the design of European economic financial reports. The CWA's first goal was to obtain a comprehensive dictionary of common concepts to all reports (*reporting*). The CWA1 group worked to create this set of concepts, definitions and types to reduce the budget in the construction of reports (*reportings*) to the Regulators in financial institutions and also increase the consistency and quality of the data (*Debreceeny et al. 2010*). The author of this thesis has belonged to this group, especially in the subgroup CWA1, contributing to this subgroup with the work: the mapping of DPM model ROLAP (CWA 2014; Santos et al 2013; Santos 2013). The goal of this methodology is to identify the *data point* (the fact) in the MDM (Díaz 2012). A tool for the development of DPM is the *DPM Architect* that is been developing by the Bank of Spain. This tool is developed for supervisors that they can develop taxonomies with DPM. Its aim is to build the XBRL taxonomies using only the XBRL specification and also provide a graphical tool for creating formulas (Santos and Castro 2011a, 2012a; Santos et al 2016; Morilla 2008). Banca d'Italia has developed the predecessor of the DPM, the *Matrix Schema* (MS) (Romanelli 2007), which contributed to the initiative *Eurofiling*. MS is based on a spreadsheet where rows show *dimensions* and *domain-members* (dimension and calculated attributes in MDM) and columns show the *primary items* (base dimension, will be seen below). The intersections of rows and columns show the facts, after in the XBRL data model (XBRLDM) hypercubes are constructed. Also, in this logic model hierarchies are shown (added in the MDM), but not its rules.

At the time of writing this thesis, the type of reports that use this specification have a so far unresolved problem which occurs when the reports are very large. The problem is that as the *XBRL instance documents* are actually *XML documents*, in order to be validated they need to be in memory and memory always has a limit. These limitations are caused by the SAX (*Simple API for XML*) and DOM (*Document Object Model*) technologies. This fact implies that in large institutions that are required to report to the Supervisor, if they do not constrain the size of the reports, they may not be treatable or cause the performance of the validating machine to fall exponentially. To overcome this problem we can try to divide the report in the stream so that the facts and context are required to be in sequential positions, so that it cannot release

memory. However this *in-stream* technology requires that the taxonomy designer group the facts.

2.4. Design of semantic reports

The need for inclusion of XBRL reports semantics has been widely analyzed by different authors. Callaghan et al. (*Callaghan et al., 2002; Callaghan et al., 2006*) show the first ideas about semantic reports in business, using UML and the XBRL specification. Sugumaran et al. (2002) present an evolution of Callaghan et al. and they show the vision of business for this type of report. Callaghan et al. (2006) depict the global idea of the business with an example. Nickerson et al. (2013) show the design for economic expert users; however, it is the basis of the design of taxonomies in general. Although, this thesis does not study the topic of ontologies, these are presented in Declerck and Krieger (2003), who analyse a translation schema for the base ontology of XBRL into OWL (*Web Ontology Language*) that was extended to German accounting principles. Silveira et al. (2007) present the business data and they classify these data. Lara et al. (2006) show the basis of an XBRL ontology using OWL. Spies (2010) presents project MUSING (*MUlti-industry, semantic-based next generation business INtelliGence*) and in it makes an extraction analysis of the ontology of XBRL, but it does not propose its automatic implementation. JiMei et al. (2012) describe the XBRL structure, and show the target of obtaining ontology of the XBRL data. In summary, in the financial world of institutions, agencies and multinationals, the semantic reports are very widely used. However, the software development life cycle of this type of report has not been analysed in a detailed way, until now.

The design of these reports is extremely general and the designs of taxonomies (set of concepts and rules for semantic reports) can vary greatly. In Europe the design makes intensive use of dimensions (*Boixo and Flores, 2005; Felden, 2007*). This use of dimensions makes the design process easier, since if the number of dimensions in the conceptual model is high, it is semantically richer, and the mapping to a database is easier. In addition, it is possible to reduce the number of dimensions through views in the logical model (the PSM) according to implementation criteria. At present, these reports use the XBRL specification, through taxonomies. This proposal will help to

design these taxonomies or semantic reports in XBRL or other technologies (XML, spreadsheets, databases, etc.), using the MDA paradigm.

There are previous work on mapping the XBRLDM to the MDM published as *Snijders (2005)*, *Santos and Castro (2011c, 2012)*, *Santos and Nieto (2014, 2015)*, *Santos et al. (2013)* and in *Openfiling (2015)*. The present contribution further develops this mapping of MDM rules with a mathematical formalization (the PIM). Moreover, an alternative and potentially better-performing report (or XBRL instance document) validation strategy based on databases rather than XML tools is presented. Finally, a proof-of-concept (POC) verifying the new research, the mapping of XBRLDM to an RDBMS and vice versa, and originally presented (*Santos and Castro, 2011b, 2011a*) is also included in the current study. Additionally, the mapping presented may serve well not only in the implementation of the proposals, but also as an open-source tool for companies and financial institutions with more limited resources.

2.5. Model Driven Architecture

OMG's Model Driven Architecture (MDA, 2015) provides an open approach to the challenge of business and technology change. Bases on OMG, the MDA separates business and application logic from the technology. Platform independent models of an application or integrated system is based on UML. The system development life cycle is the process where requirements are within a domain, systems that address the requirements may be considered solutions residing within an environment, and this involves understanding a problem, solving the problem, and implementing the solution. Every system development life cycle process involves the following types of life cycle activities (*Si Alhir, 2015*): Requirements gathering activities, Analysis activities, Design activities, Implementation activities, Testing activities and Deployment activities. The MDA model is related to multiple standards, including the *Unified Modeling Language (UML)*, the *Meta-Object Facility (MOF)*, *XML Metadata Interchange (XMI)*, *Enterprise Distributed Object Computing (EDOC)*, the *Software Process Engineering Metamodel (SPEM)*, and the *Common Warehouse Metamodel (CWM)*. One of the main aims of the MDA is to separate design from architecture. The MDA is an open and vendor-neutral architectural framework, associated OMG standards within the systems development life cycle across various domains and technologies. The MDA broadly supports different types of application domains and

technology platforms. The MDA enables transforming or converting platform independent models (the PIM) to produce platform specific models (the PSM) using mappings. Within the system development life cycle process, the MDA applies the PIM and the PSM to sustain and leverage investments in requirements, technologies, and the life cycle that bridges the gap between them as they independently change. Such an approach generally leads to long-term flexibility of implementations, integration, maintenance, testing and simulation as well as portability, interoperability and reusability. There are some papers that try resolving the developing of XBRL reports using UML as *Mendez-Nunez and Trivino (2010)* and *Callaghan et al. (2002 and 2006)*. This thesis proposes the developing of the semantic reports using MDA.

CHAPTER 3

Model of meta-metadata
of the specification

CHAPTER 3. Model of meta-metadata of the specification

3.1. Introduction

In this chapter the XBRL specification is introduced and analysed in detail. It begins by analysing the meta-metamodel.

A meta-metamodel is understood as the set of definitions and rules of a specification. As noted, XBRL is an XML-based specification, and therefore there will be many references to it, however, in this thesis, starting from the specification, a set of definitions and rules based on the MDM will be proposed.

Taxonomy is a science that deals with the principles, methods and purposes of classification (Definition of the *Royal Academy of the Spanish Language*, 22nd edition). In the XBRL specification, a taxonomy is the set of documents where the concepts that will later be represented in financial economic reports (*reports*) are defined, along with how these concepts are related. Therefore this set of concepts is presented in a structured way using the data. Hence, XBRL enables the information modelling and semantic meaning commonly required in financial economic reports. It uses XML syntax and related technologies such as *XML Schema*, *XLink*, *XPath* and *Namespaces* to provide semantic meaning (*Hernández-Ros, 2009*).

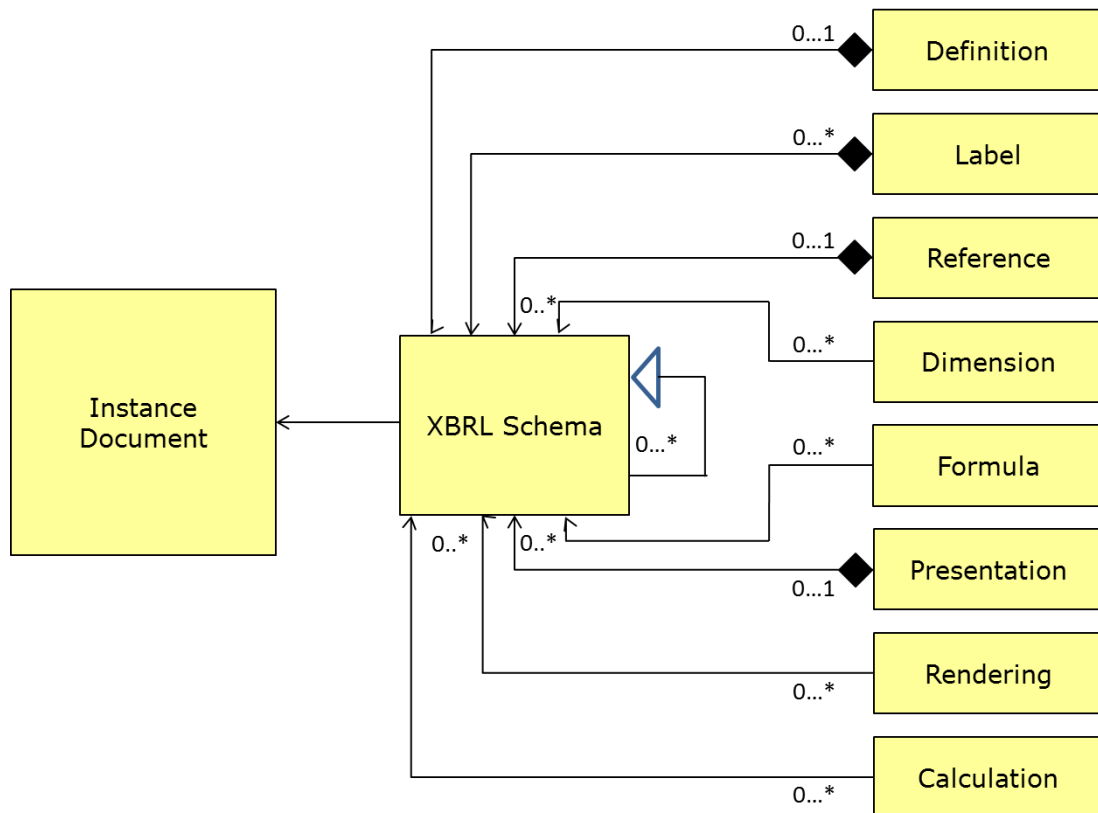


Figure 3.1 Structure of the XBRL specification

A report that uses this specification, as *Figure 3.1* shows, consists of an *XBRL instance* (*XML instance document*), which is the report itself or *report*, also called *XBRL instance document* that consists of a set of *XML Schemas* or *XBRL Schemas*, called *Discoverable Taxonomy Set* (DTS), which specify the concepts, rules and constraints (*Engel et al., 2008; Santos and Castro, 2011a, 2011c*). Each *XBRL Schema* consists of a set of roles or *linkbases* in the *XBRL Data Model* (XBRLDM). These roles are shown in *Figure 3.2* as a set of constructors of an *XBRL instance document*.

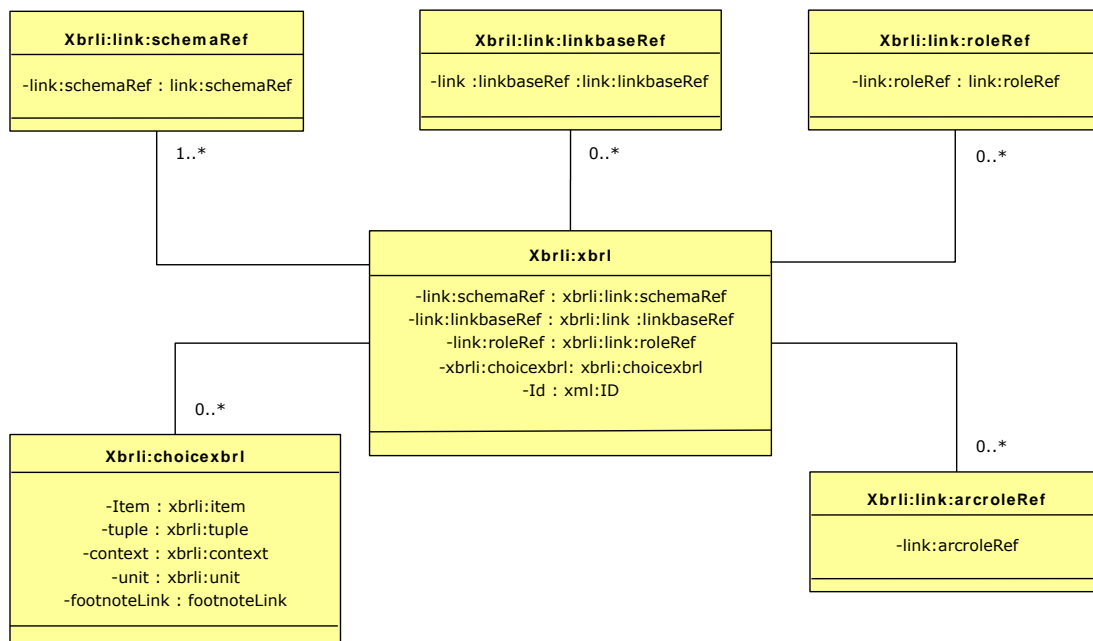


Figure 3.2 Structure of an XBRL Instance Document

The roles are defined as:

- **Definition.** It allows the establishment of relationships between concepts, and they are fixed by the business logic.
- **Label.** This *linkbase* is used to give names to concepts defined in different languages.
- **Reference.** It allows the inclusion of references to legal regulation, no text is included, only its reference.
- **Dimension.** It allows valid combinations of dimensions.
- **Formula.** It supports complex expressions based on *XPath*, which can be applied to *instance documents* to validate your information.
- **Presentation.** It establishes hierarchical relationships between elements (concepts) defined in a schema. It is used as a guide for the presentation, because the concepts have a hierarchy. This hierarchy entails an order which is established by the business logic.
- **Rendering.** Firstly, it is embedded in the HTML specification, and secondly, it is a Specified Transformations registry, so that a report would be more understandable, but without giving incorrect data. It isn't in this moment a standard.

- **Calculation.** It provides simple relationships of calculation between different elements (concepts). Only simple operations are possible, nor formulas or expressions.

Each XBRL or XML *Schema* may have 0 or 1 or N *roles* depending on the role type (Santos and Castro, 2011a, 2011c). In this specification a *Primary Item* in the XBRL data model defines a fact schema, without dimensions, in the MDM (Martín Quetglás, 2006; Hernández-Ros and Wallis, 2006; Engel, et al., 2008; Schmehl and Ochocki, 2009). For example, a *primary item* can be *Asset*, but we don't know if this concept, this asset belongs to *Equities*, a *Risk*, and so on.

In Figure 3.3, the layers of the specification in the data model of XBRL (XBRL Data Model, XBRLDM) are shown:

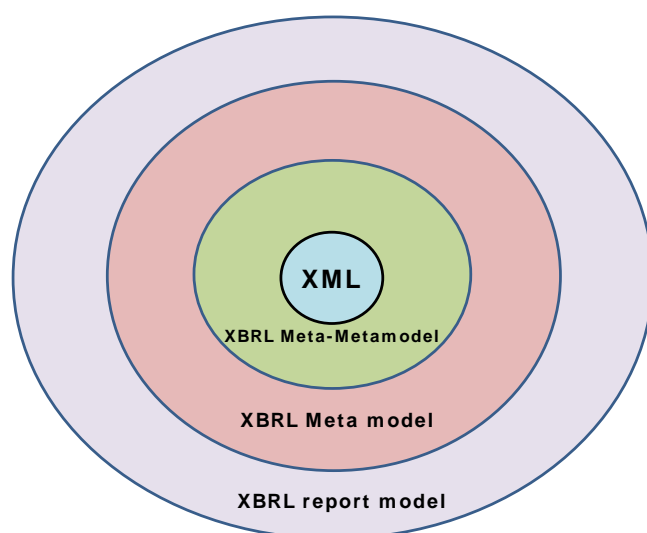


Figure 3.3 The data model of XBRL

XBRL is based on XML and therefore in Figure 3.3, the XML layer is in the centre of the figure. On it the set of definitions and rules that make up the specification can be built. In the next layer is the metamodel, which is the set of taxonomies that define a report. The outermost layer is the report itself.

The following sections analyse the set of necessary XML elements, based on the XBRLDM.

3.2. Models and XML objects

An XML document is used to store information (*Morrison, 1999*). In the decade of the 70s, it was intended to structure documents in an organized way to facilitate exchange and manipulation. Sponsored by IBM, the *Generalized Markup Language* (GML) was created. GML was used to produce books, reports and other documents. Also in collaboration with IBM the SGML (*Standard Generalized Markup Language*) standard was defined and adopted for information exchange (ISO-8879). One of the main uses of this standard was to format and keep IBM's legal documents. SGML is based on the marking-up and labelling of documents, but labels are not defined. The structure of a document is based on the DTD (*Document Type Definition*) (*Rusty Harold, 2004*). SGML is a metalanguage with a set of rules to define tags. An SGML document consists of three parts (*Sperberg-McQueen and Burnard, 2002*): the declaration of the SGML, the declaration of the document (DTD), and the instance itself.

The declaration indicates the types of characters, delimiters and document characteristics. The DTD defines the structure of the document. It contains naming rules, content descriptions and item usage details. The DTD interprets a document and serves as validator. Finally, the instance is the report itself or document that the business user requires.

An evolution of the SGML specification is the *HyperText Markup Language* (HTML) specification. In February 1998 the XML 1.0 specification (*eXtensible Markup Language*) formally appeared, as an evolution of HTML and supported by SGML. XML was defined as a system for defining, validating and sharing document formats on the Web. HTML focuses on the presentation of data and XML data models (*w3schools, 2015*).

The DTD provides a grammar of an XML vocabulary and it determines the structure of XML documents. A DTD consists of elements, attributes, entities, and notations. The DTD may be embedded in the document or referenced in an external source.

A document is **validated** when it meets the following requirements (*Morrison, 1999*):

- The document must be well formed.

- The name of the root element must be the same as the name of the document type declaration.
- The document must have a DTD in which the elements, attributes and entities are declared, it is used in the document as internal, external or both.
- The document must meet the grammar established by the DTD.

In XML an element provides its own structure that allows the definition of a logical structure in the document. In the following example there is an XML document. This document begins with a statement that identifies the version used, and the 16-character code. Then a root element will appear containing the rest of the objects in the document.

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

Example 3.1 Example of an XML document

In the first line of the example indicates the version of the document, the code used and if it has an associated a DTD.

The attributes have characteristics associated of the element and they do not contain complex information as shows the *Example 3.2*:

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

Example 3.2 Example of attributes in a XML document

Attributes can be any set of values from a list, or default values, and they can have data types.

DTD has certain limitations, such as that the syntax is not easily understandable and there is a lack of support *XML Schema* namespace, based on W3C, which defines an

XML vocabulary (*XML.com, 2015*). *XML Schema* defines the structure of *XML documents* and valid types of data for each element and attribute.

The *XML Schema* structure is modular, allowing reuse of defined modules. Furthermore, a relationship hierarchy is established between these modules. There are two types of objects in an XML schema: simple and complex types. Simple types are those that have no child elements or attributes. The predefined XML types are *double*, *string*, *boolean*, etc. The enumerated or list type and unions are also simple types. The complex types are those that have child elements and/or attributes.

New specifications associated with XML have appeared, including XPath, XQuery, XPointer, and XLink¹, sponsored by W3C, *World Wide Web Consortium*.

3.3. Model of the XBRL meta-metadata

As shown in this chapter, especially in *Figures 3.1* and *3.2*, a report using the XBRL specification, called an *XBRL instance document*, is formed by a series of *linkbases* or *roles* (from the point of view of data modelling) that consist of a taxonomy, and the instance itself. However, it is possible to say that an *XBRL instance document* consists of the XBRL specifications based on the XML specification and a set of *XML Schemas*, as *Figure 3.4* shows.

¹ See *Acronyms*.

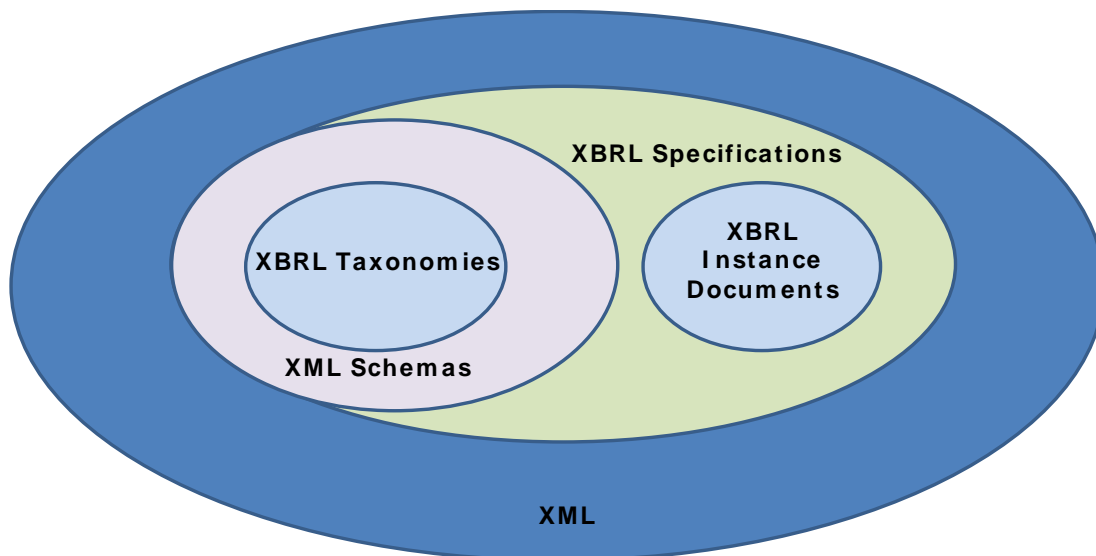


Figure 3.4 Structure XML versus XBRL

Accordingly, an XBRL report consists of different levels (*XBRL Spain, 2006*):

- Contents of the report, consisting of a set of accounting events that conform to current accounting standards.
- Definition of the concepts and the external and internal relations of the concepts themselves (taxonomies).

If the architecture of the XBRL specification is analysed, it can be defined in **four** layers:

- **Conceptual layer.** Concepts, syntax and semantics are defined.
- **Relations layer.** In this layer the relationships between concepts are detailed.
- **Taxonomies layer.** Called DTS (*Discoverable Taxonomy Set*), which is the union of taxonomies and document *linkbases* to interpret a instance document (report). A DTS is a set of *XBRL Schemas* or *XML Schemas* and its associated XML files or roles that form one taxonomy and validate an *XBRL instance document*.
- **Extensions Layer.** They are extensions, upgrades and modifications of the taxonomy.

3.4. XML and XBRL objects used in the definition of the meta-metadata

Next, the XML and XBRL objects forming the XBRL specification are analysed, although not in detail, because they are analysed in *Valencia (2011)*, and in *Openfiling (2015)*.

The **objects XML** used to define the meta-meta model **XBRL** are:

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

Next, the XBRL objects are shown, on which is built the specification. These objects are structured in attributes, groups of attributes, simple types, complex types and elements.

XBRL Objects “Attributes”

- *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*
- *xbrli:balance*
- *xbrli:periodType*

XBRL Objects “Groups of Attributes”

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

XBRL Objects “Simple Types”

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

- *xbrli:shares*
- *xbrldt:contextElementType*

XBRL Objects “Complex Types”

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

- *xbrli:exBinaryItemType*
- *xbrli:integerItemType*
- *xbrli:intItemType*
- *xbrli:longItemType*
- *xbrli:measuresItemType*
- *xbrli:monetaryItemType*
- *xbrli:negativeIntegerItemType*
- *xbrli:nonPositiveIntegerItemType*
- *xbrli:normalizedStringItemType*
- *XBRL xbrli:positiveIntegerItemType*
- *XBRL xbrli:pureItemType*
- *xbrli:QNameItemType*
- *XBRL xbrli:sharesItemType*
- *XBRL xbrli:shortItemType*
- *XBRL stringItemType*
- *XBRL xbrli:timeItemType*
- *xbrli:tokenItemType*
- *xbrli:unsignedByteItemType*
- *xbrli:unsignedIntItemType*
- *xbrli:unsignedLongItemType*
- *xbrli:unsignedShortItemType*
- *xbrli:contextEntityType*
- *xbrli: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*
- *xbri:contextScenarioType*

This type (**contextScenarioType**) describes additional information, equivalent to a set of dimensions in the Multidimensional Data Model (Figure 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>
```

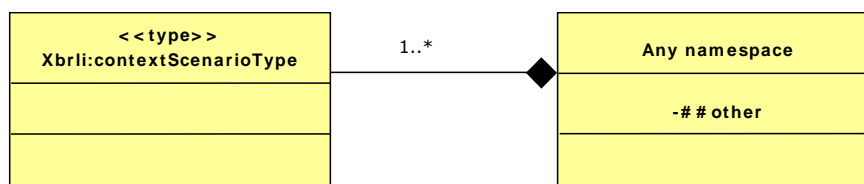


Figure 3.5 Type XBRL *contextScenarioType*

Example 3.3 shows a scenario that consists of a namespace, which define the dimensions forming the scenario and therefore the context.


```

<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>

```

Example 3.3 Example of scenario

- *link:linkbase*

The *link:linkbase* complements the rule XBRL in the specification on the XLink (Figure 3.6), they were explained before (presentation, calculation, formula, 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>
    <attribute name="id" type="ID" use="optional"/>
    <anyAttribute namespace="http://www.w3.org/XML/1998/namespace"

```

```

    processContents="lax"/>
  </complexType>
</element>

```

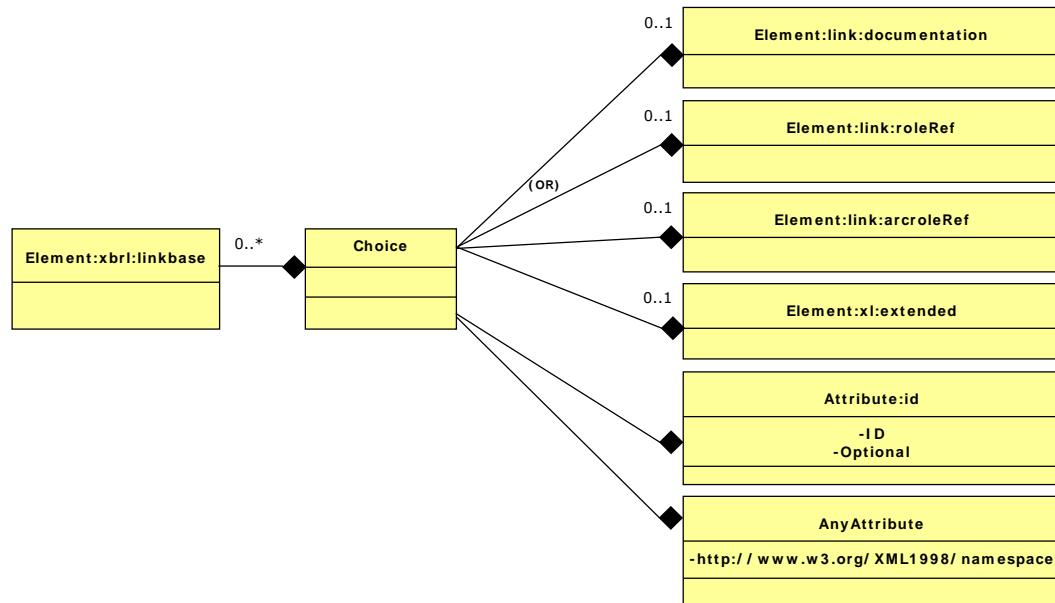


Figure 3.6 Element XBRL *linkbase*

XBRL Objects “Elements”

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

- *link:schemaRef*

This element is used for the XBRL instance documents references to taxonomies schemas and not rewrite the element (*Figure 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>
```

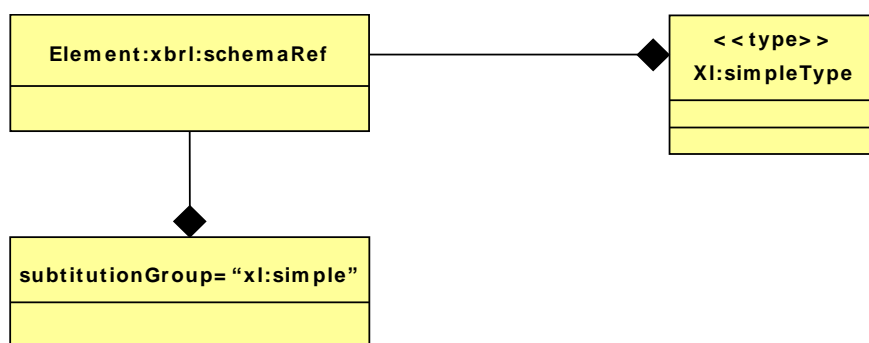


Figure 3.7 Element XBRL *schemaRef*

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

Context is the set of dimensions and dimension attributes of a fact. It consists of 0-1 segment (pair <dimension/attribute dimension>) and from 0 to n scenarios, pairs <dimension, dimension attribute>, and with the organization and time dimension. In *Figure 3.8* the structure of the definition is shown. *Example 3.4* shows a context with its scenario and two dimensions, for entity ES9000, in a instant period 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>

```

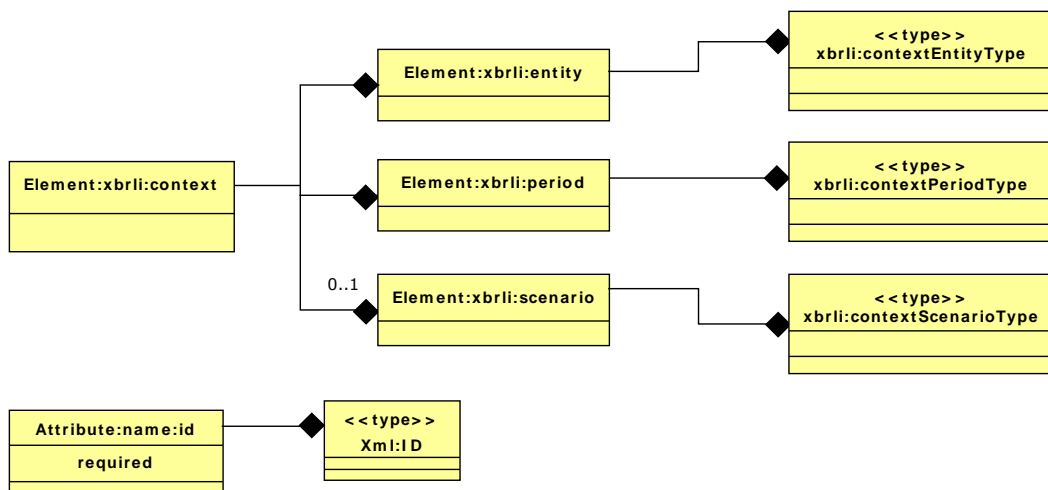


Figure 3.8 Element XBRL context

```

<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>
      <xbrli:instant>2008-09-30</xbrli:instant>
    </xbrli:period>
  </xbrli:context>

```

```

</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>

```

Example 3.4 Example of the element XBRL context

- *xbrli:denominator*
- *xbrli:numerator*
- *xbrli:divide*
- *xbrli:item*
- *xbrli:measure*
- *xbrli:segment*

The element *segment* defines a business segment, and in the multidimensional data model (MDM) defines a pair <dimension, dimension attribute> (Figure 3.9). Example 3.5 shows a *segment* containing the pair <Agrupación, AgrupaciónIndividual>.

```

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

```

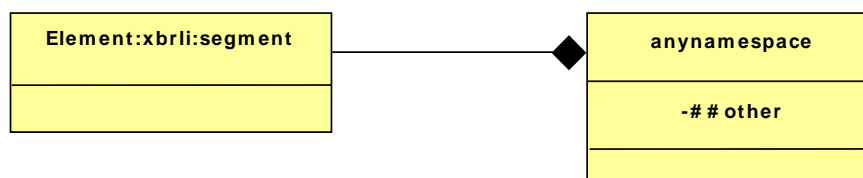


Figure 3.9 Example of the element XBRL *segment*

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

Example 3.5 Example of the element 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*

The element “dimensionItem” that is shown in the next figure (*Figure 3.10*) collects the multidimensional concepts. This element will be analysed below.

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

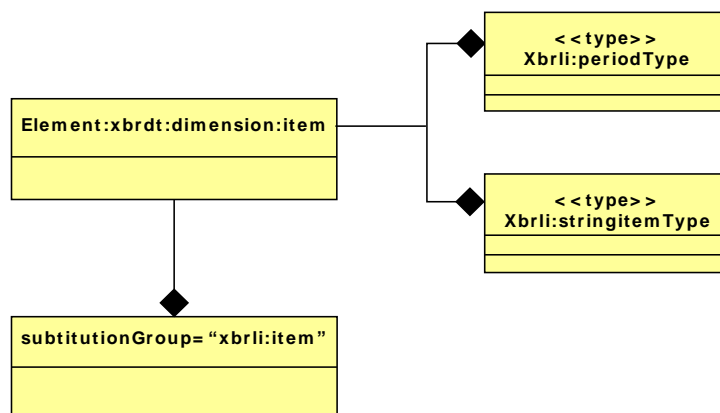


Figure 3.10 Element XBRL *dimension Item*

- *xbrldt:hypercubeItem*

It defines a cube with 1 or n facts (*Figure 3.11*). This element will be analysed below.

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

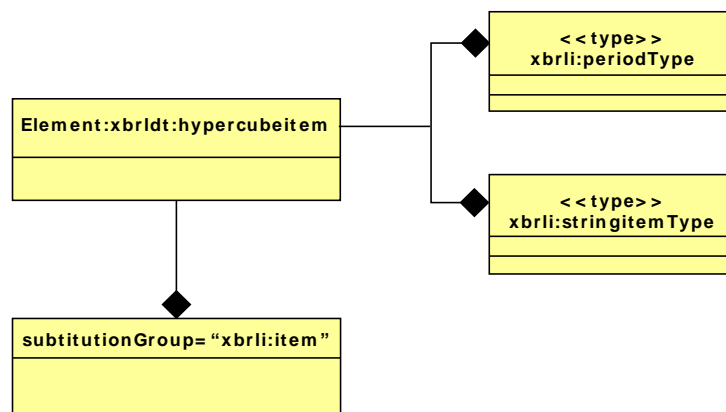


Figure 3.11 Element XBRL *hypercubeItem*

- *xbrli:xbrl*

This element is the root of all instance documents (*Figure 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>

```

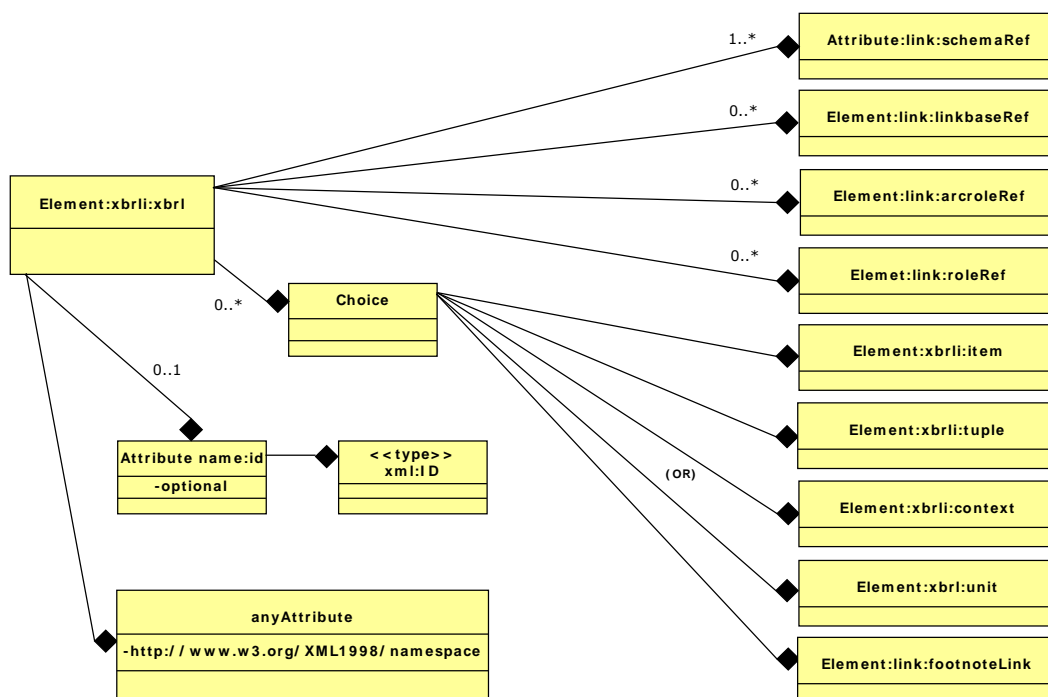


Figure 3.12 Element XBRL xbrl

Example 3.6 shows a case of a XBRL instance document. In this example is shown the definition of the segment as the dimension "es-be-ce-dim:AgrupacionIndividual". In addition, the defined period has its starting date in 2008-07-01 and end date on 2008-09-30.

```
<xbli:xbli>
  <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"/>
  <xbli:context id="Context_Instant"></xbli:context>
  <xbli:context id="IS1"></xbli:context>
  <xbli:context id="Context_Instant_GrupoConsolidableEntidadesCredito">
  </xbli:context>
  <xbli:context id="Context_Instant_EntidadesSeguros"></xbli:context>
  <xbli:context id="pblo_FINREP"></xbli:context>
  <xbli:unit id="EURO"></xbli: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>
</xbli:xbli>
```

Example 3.6 Example of an XBRL instance document

This example shows an item of the *Example 3.6*.

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

Example 3.7 Detail of an item in an XBRL instance document

The *Example 3.8* shows an example of the XBRL instance document that does use tuples.

```
<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>
```

Example 3.8 Example of the use of tuples in an XBRL instance document

- *xbrldi:explicitMember*

Dimension attributes of a dimension (Figure 3.13), the dimension attributes explicitly defined in the taxonomy (these attributes can also be calculated in the 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>
```

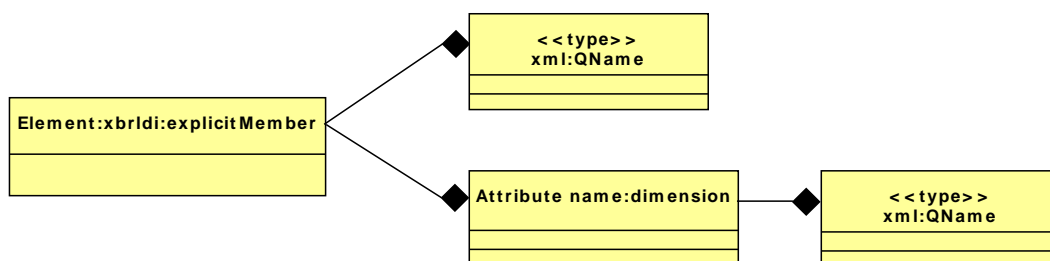


Figure 3.13 Element XBRL *explicitMember*

The Example 3.9 shows the dimension *Agrupación* and dimension attribute *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>

```

Example 3.9 Example of the element XBRL *explicitMember*

- *typedMember*

Finally, as in the following chapters will be analysed, a typed dimension is a dimension that dimension attributes are not known at the time of defining the taxonomy (in run-time), but its type, if it is known. Implicit dimension (*Figure 3.14*) is also called.

```

<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>

```

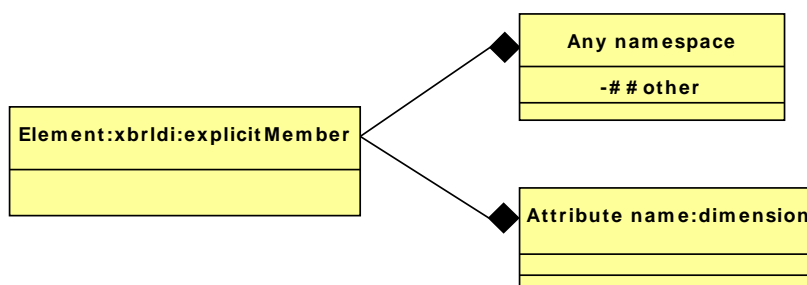


Figure 3.14 Element XBRL *typedMember*

CHAPTER 4

Fundamentals and Basic
Concepts of the proposal

CHAPTER 4. Fundamentals and Basic Concepts of the proposal

4.1. Universe of the Discourse and the proposal

This chapter analyses the software development life cycle of the metadata in accounting semantic reports, using the *Model Driven Architecture* (MDA, 2015) paradigm. From 2008 especially in Europe, the *National Central Banks* (NCBs), *European Central Bank* (ECB), *European Banking Authority* (EBA), and Supervisors and Regulators in general have been looking for a data model for the development of this type of reports (*Eurofiling*, 2015). The *Data Point Model* (DPM) appeared as a logical design model (only applicable to the XBRL specification) with a stable version in 2010 (*Eurofiling*, 2011), for the *Balance Sheet Items* and *Monetary Financial Institutions Interest Rates* taxonomies (BSI-MIR (2010), Bank of Spain (2010)) and they were implemented by the Polish company BR-AG (2015). Two years later, an Excel template generator (Data point modeller, EXGEN (*Ruíz et al.*, 2012)) was developed for XBRL taxonomies. The *Comité Européen de normalisation/European Committee for Standardization* and *Eurofiling* (2015) (CEN, 2013) was working on the standardization of the data model design in Europe, named the *Data Point Model* (DPM) (*Díaz*, 2012). A tool to model DPM, called *DPM Architect for XBRL*, is under development at the Bank of Spain. This tool supplied to some supervisors who wish to develop taxonomies with DPM. The target is to construct taxonomies using the XBRL specification exclusively. Banca d'Italia has developed the predecessor of the DPM, the *Matrix Schema* (MS) (*Romanelli*, 2007), which contributed to the *Eurofiling* initiative (2015). The MS is based on a spreadsheet. As commented in other sections, this thesis does not deal with the DPM, because the work will be wider in scope, however the author of this thesis has worked on it elsewhere (*Santos*, 2013; *Santos et al.*, 2013).

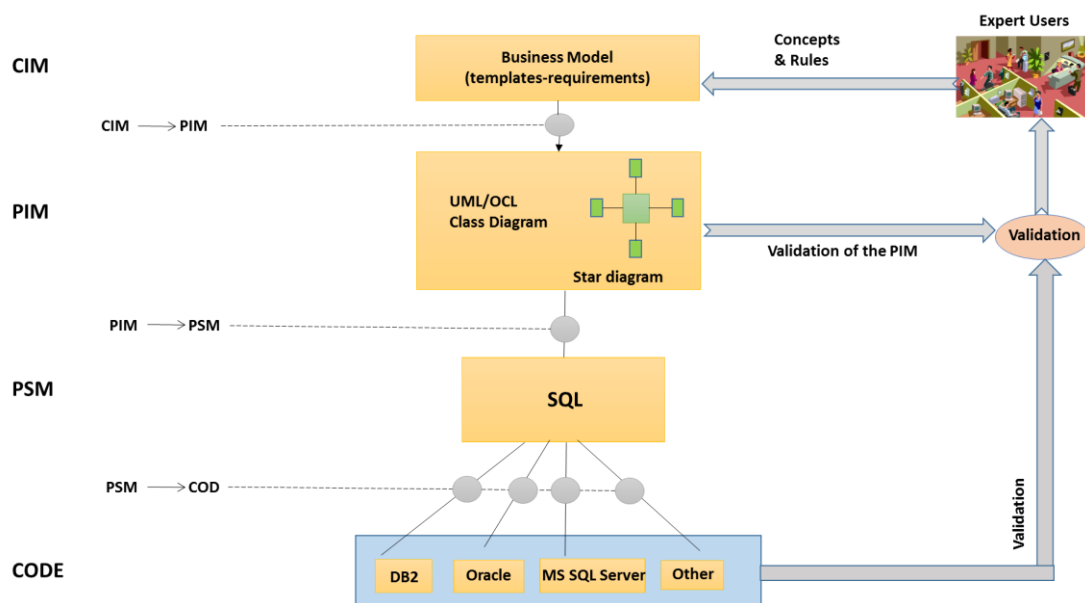


Figure 4.1 Design of semantic reports using the MDA paradigm

Figure 4.1 displays one part of the proposal of this thesis, the software development life cycle, which will be developed in this chapter. Accountants, financiers or economists need to obtain a set of data. These expert users, with the help of IS, builds a set of templates through one or more spreadsheets. Therefore, the real world consists of a set of accounting rules, laws, directives, etc., defined in a set of required data in a report (the *Computation Independent Model*, CIM), through templates. According to the MDA paradigm, the *Platform Independent Model* (the PIM) is obtained from the CIM. In the PIM the set of definitions, user rules, and a mapping from the CIM to the PIM is analysed. The model used by the author is MDM. MDM is a straightforward model that combines objects, dimensions (hierarchies), measures and attributes for representing real work business problems (Kimball, 1996-2004; Inmon, 2005; Jarke et al., 2003). In addition, this model is the heart of On-Line Analytical Processing (OLAP), which requires complex queries that can be solved by the MDM in real time.

The specific platform chosen is SQL (*Platform Specific Model*, PSM). The PSM consists of a set of definitions, rules and transformations. Because the MDM is used in the PIM and SQL in the PSM, the mapping is almost immediate (using *Relational Online Analytical Processing* (ROLAP) technology). Finally, the transformation from

the PSM to the code is shown. In the PIM and the PSM a set of automatic validations are provided.

The design of these reports is extremely general and the designs of taxonomies (set of concepts and rules for semantic reports) can vary greatly. In Europe the design makes intensive use of dimensions (*Boixo and Flores, 2005; Felden, 2007*). This use of dimensions makes the design process easier, since if the number of dimensions in the conceptual model is high, it is semantically richer, and the mapping to a database is easier. In addition, it is possible to reduce the number of dimensions through views in the logical model (the PSM) according to implementation criteria. At present, these reports use the XBRL specification, through taxonomies. This proposal helps to design these taxonomies or semantic reports in XBRL or other technologies (XML, SDMX, iXBRL, spreadsheets, databases, etc.), using the MDA paradigm.

This chapter is divided into five sections, in order to explain the software development life cycle of these reports, using the MDA. Firstly section 2 analyses the CIM (*Computational Independent Model*). Next section 3 analyses the PIM (*Platform Independent Model*), its rules and definitions. Section 4 studies semantic questions about the design of the metadata in the PIM. Finally, section 5 shows the PSM (*Platform Specific Model*), where the code of this proposal is shown. The validation will be analysed in *Chapter 6* of this thesis.

4.2. The Computation Independent Model (the CIM)

This section will start to analyse the real world. An economist-accountant wants only to obtain a set of data. In certain cases, these specialists design a report as in *Figure 2.2*. However, in most cases, they want to collect data independently of its presentation. A generalized method is to generate one or more spreadsheets or templates with the data that are needed. In this way, the presentation of the data is separated from its definition. According to the business logic, the user will create one or more spreadsheets, each sheet having a group of cells with the data that economists or accountants want to obtain in the reports. *Figure 4.2* shows a simplified example with a similar report to *Figure 2.2*. In this example there is a set of concepts that are interconnected and a set of values assigned to these concepts. In addition, the values must comply with certain rules and constraints, with other values and with

other concepts of this report or set of reports. This figure depicts the Financial Assets in a period in a country, with specific rules, such as that the real estate loans of the bank must be equal to the sum of the real estate loans with the bank itself and other banks.

Germany Euros

Financial assets, period: 1st 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

Figure 4.2 Example of report with semantic

Firstly, the following definition of an economic/financial semantic report is proposed:

Definition 4.1: *An economic/financial report is semantic if it is composed of a set of interconnected concepts, and values assigned to these concepts or group of concepts. Also, the values must comply with certain rules and/or constraints among other values and concepts.*

However, in most cases, the expert users want to collect data independently of its presentation. A generalized method is to generate one or more spreadsheets or templates with the data that are needed. In this way, the presentation of the data is separated from its definition. According to the business logic, the user will create one or more spreadsheets, each sheet having a group of cells with the data that economists or accountants want to obtain in the reports. *Figure 4.3* shows a simplified example with only three cells, based on the report of *Figure 4.2*, these are F(5, 1, 1), F(5, 1, 2) and F(5, 1, 3). These cells belong to the column values of *Figure 4.2*, F(5.1.1) to row 1, F(5,1,2) to row 2 and F(5, 1, 3) could be row 5, if the *Banks* concepts were the same as *To bank itself and other Banks*.

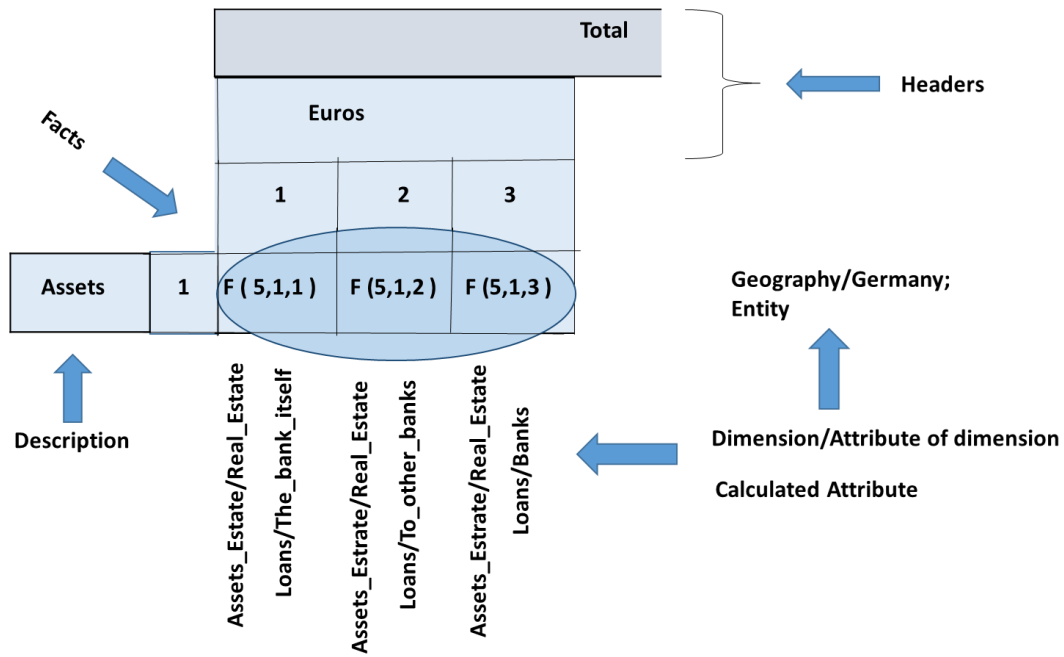


Figure 4.3 Sample of template

From these templates the IT analyst, together with the business user, extracts the metadata. In this template, the business users show the data they need to gather. The analyst may find a set of Excel sheets with a large number of cells unconnected with each other and with a high degree of redundancy. The European Regulation in 2012 needed 4500 cells, in 2015 the number of cells is 45000 (Weller, 2015). Each template has a different meaning for the business user. The template consists of a set of cells where each cell is a fact to be gathered, this being determined by a set of dimensions and dimension attributes, among other things. For example, the cell or fact $F(5, 1, 1)$ is real estate assets, with a loan from the bank, for an entity, in euros. In this figure, if the fields are crossed out, they are considered not allowed by the business user. Facts (cells for the expert user) are listed in a triplet, which is the numbers of the template, row, and column. Normally, facts are referenced with sequential numbers. A fact can be in more than one template, so that they can be defined as:

Definition 4.2: Let φ be a Fact. Let i be a number, where $i \in N$. Then $\varphi \leftrightarrow i$.

On the other hand, a fact can determinate more than one triplet (template, row, column), because a fact can be in more than one template. Therefore, it is possible to define the following rule:

Rule 4.1: *Let φ be a Fact. Let T_i be a template, $\forall i=1, \dots, n$. Let R_j be a row, where $R_j \in T_i$, $\forall j=1, \dots, m$. Let C_k be a column, where $C_k \in T_i$, $\forall k=1, \dots, o$. Then $\varphi \rightarrow (T_i, R_j, C_k)$.*

The proof of concepts (hereinafter POCs) of this paper are based in the reports that must be sent from financial institutions to European Supervisors (*Openfiling*, 2015). These POCs use the draft of the taxonomy FINREP 2012 (*EBA*, 2011; *Eurofiling*, 2012, 2015), published on the internet, with intensive use of dimensions.

4.3. The Platform Independent Model (the PIM): Rules and Definitions

This section analyses the PIM of this model. It uses UML for showing all necessary definitions and rules of this platform. This thesis proposes to use the star design of the MDM, with a minimum set of 16 definitions or rules as constructors of the semantic economic/financial report. *Table 4.1* summarises the set of definitions. After this table, the definitions and rules with examples in an extensive way are explained. Column 1 defines the name of the concept in the MDM and column 2, its description. However, these definitions are based on the XBRL Data Model (XBRLDM). Additionally, this section presents a new dimension to the proposal by offering a novel mathematical formalization (formal language in the MDM).

Table 4.1 Definitions and rules in the MDM (the PIM)

Name in the MDM	Description
Concept	The definition of a business concept or item. Each concept is associated with a time period type attribute (<i>Instant</i> , <i>Period</i> , and <i>Forever</i>).
Basic Concept	A special concept that has an associated data type, time period type, and balance type.
Domain	A group of concepts belonging to a field or scope of knowledge or activity. In this model a domain can contain basic concepts or non-basic concepts but not both.
Base Dimension	A domain with only basic concepts.
Dimension	A set of concepts of a domain. These concepts have a tree-like structure.
Dimension (explicit / implicit)	This is explicit if the attributes are defined. It is implicit if they are not defined. Dimension → Domain. Domain → → Dimension.
Dimension Group	Group of dimensions of a domain.
Calculated attribute	An aggregate of dimension attributes of a dimension, and/or calculated attributes.
Attribute of dimensions	Not an aggregate.
Attribute by default	Each domain has a concept by default.
Hierarchical Constraint	Concepts in a dimension have a tree-like structure. Validation is between a leaf and its leaves below, that is to say, it is used for the calculated attributes.
References	References to directives or laws of the concepts.
Dimension/Dimension attributes and Calculated attributes	Can define a fact or several facts.
Fact::= - <Dimension/Dimension attribute> - Basic concept - Calculated attribute	A fact is a value representing a particular measurement provided by the reporting entity.
Allowed fact	User constraint.
Forbidden fact	User constraint.

The first definition, according to the *Table 4.1*, is the definition of a business concept or item. Each concept has a time period type attribute. The time period type attribute can be: *Instant*, *Period*, and *Forever*. In the *Figure 4.2*, the concepts are {*Entity_Finan*, '*BNP Paribas*', '*ING Group*', '*Royal Bank of Scotland*', *Commerzbank*, '*Real estate*', '*No real estate*', '*Real estate and no real estate*', *Assets*, *Liabilities*, ...}.

A basic concept is a *primary item*, in the XBRLDM, is a special concept because it has three associated attributes: type, time period type of the basic concept and balance (Hernández-Ros and Wallis, 2006; Santos and Castro, 2011b, c). According to XBRLDM, if a data type of a basic concept is monetary, there is a new attribute, balance, which can take two values, *debit* or *credit*. If the data type is not monetary, then the value of the attribute *balance* will be null. In the example the basic concepts are {Assets (*Instant*, *monetary*, <*balance*>), Liabilities (*Instant*, *monetary*, <*balance*>), ...}.

All concepts of a domain have the same type of time period. A domain is formed of a set of concepts, and each concept belongs to a single domain. In this example the set of domains are *DEntity*, *DAssets_Estate*, *DLoans*, and *DGeography*. Where the domain *DEntity* consist of the next concepts {‘*Entity_Finan*’, ‘*BNP Paribas*’, ‘*ING Group*’, ‘*Royal Bank of Scotland*’, ‘*Commerzbank*’} and the domain *DAssets_Estate*= {‘*Real estate*’, ‘*No real estate*’, ‘*Real estate and no real estate*’}, and *DLoans* = {‘*Banks*’, ‘*The bank itself*’, ‘*To other banks*’, ‘*To non-financial institutions*’, ‘*The bank itself and other banks*’}, etc.

These definitions can be expressed formally as:

Definition 4.3: Let ξ_i be a concept, where $\xi_i \in \Gamma$, where Γ is the set of concepts, $\forall i=1, \dots, n$. Let Do_j be a domain, such that $Do_j \in Do$, where Do is the set of domains, $\forall j=1, \dots, m$. Then $Do_j = \bigcup_{i=1}^k \xi_i$.

Definition 4.4: According to the above definition, $\forall \xi_i \Rightarrow \xi_i \rightarrow Do_j$.

Definition 4.5: Let ξ_i be a concept, where $\xi_i \in \Gamma$, where Γ is the set of concepts, $\forall i=1, \dots, m$. Let q be the time period type, where $q = \{ \text{“Instant”} \mid \text{“period”} \mid \text{“forever”} \}$. Let Do_j be a domain, $\forall j=1, \dots, n$, where $Do_j \in Do$, where Do is the set of domains. Then $\xi_i \rightarrow q$. And if $\xi_i \rightarrow Do_j \Rightarrow Do_j \rightarrow q$.

A dimension is a set of concepts of a domain. In the MDM or the XBRL specification cannot have more of a dimension attribute of a dimension that refers to a fact. However, in the real world there can be more than one concept for a domain that makes references to a fact. The solution in the XBRLDM is to create as many dimensions of the same domain as possible, so that each fact has a dimension

attribute (*member-domain* in XBRLDM), without overlapping the dimension attributes of a dimension in a fact. Dimensions of a domain with attributes of dimension overlapped are created in the MDM. This means, a dimension determines univocally a domain. Then, in the example is possible to define in the domain *DLoans*, the dimension *Loans_1*={'The bank itself'+ 'To other banks'} and in the domain *DAssets_Estate*, the dimension *Assets_Estate_1*={'Real estate', 'No real estate'}.

A calculated attribute determines univocally a domain and a dimension. For example, in the domain *DLoans* the concept 'The bank itself and other banks' is a calculated attribute of *Loans_1*, where 'The bank itself and other banks'='The bank itself'+ 'To other banks'. Also, the concept 'Real estate and no real estate' of the domain *DAssets_Estate* is a calculated attribute of the dimension *Assets_Estate_1*, where 'Real estate and no real estate'='Real estate'+ 'No real estate'.

A dimension attribute determines a single concept of a domain, but dimension attributes determine from 1 to n dimensions. Therefore, it is possible to define:

Definition 4.6: Let D_i be a dimension, $\forall i=1, \dots, m$, where $D_i \in D$, and D is the set of dimensions. Let At_{i_j} be an attribute of dimension such that $D_i = \cup_{j=1}^k At_{i_j}$. Let Do_l be a domain, and let ξ_{l_o} be a concept, where $Do_l = \cup_{o=1}^{k'} \xi_{l_o}$. If $At_{i_j} \rightarrow \xi_{l_o} \Rightarrow \xi_{l_o} \rightarrow Do_l$ and $\xi_{l_o} \rightarrow \sum_{i=0}^n D_i \in Do_l$. $\forall l = 1, \dots, n$ and $\forall o = 1, \dots, n'$.

In XBRLDM a dimension consists of domain-member, and does not differentiate between dimension and calculated attributes. A calculated attribute is defined as:

Definition 4.7: Let $\xi_{i'_{j'}}$ be a concept, and let $Do_{i'}$ be a domain, $\forall i' = 1, \dots, m, \forall j' = 1, \dots, m'$. Where $\xi_{i'_{j'}} \in Do_{i'}$. Let At_{i_j} be an attribute of dimension, where $At_{i_j} \in D_i$, where D_i is a dimension $\forall i = 1, \dots, n$, and $\forall j = 1, \dots, n'$. If $\forall \xi_{i_j} \in Do_{i'}$, \nexists a $\xi_{i'_{j'}}$ such that $At_{i_j} \rightarrow \xi_{i'_{j'}}$, then $(\xi_{i'_{j'}}, D_{i'}) \rightarrow CAt_{i'_{j'}}$, $CAt_{i'_{j'}}$ is a calculated attribute.

The set of basic concepts are grouped in the base dimension. The base dimension determines a domain, and each of the basic concepts determines univocally the concepts of the domain. In general, the XBRLDM specification does not specify whether there are one or several base dimensions, but according to the guide of best practices of *Eurofiling* indicates only a base dimension by taxonomy or framework

exists (CEN, 2013). In the example the base dimension is $Base_Dimension = \{Assets (Instant, monetary, <balance>), Liabilities (Instant, monetary, <balance>), \dots\}$.

Therefore, it is possible to define:

Definition 4.8: Let ξ_j be a concept, $\forall j = 1, \dots, n$, and $\xi_j \in \Gamma$, where Γ is the set of concepts. Let ϱ be the period type, where $\varrho = \{“Instant” “period” “forever”\}$. Let o be a data type. Then $\xi_j \cup \varrho \cup o \rightarrow Bc$, where Bc is a basic concept and $\xi_i \rightarrow \varrho$, $\xi_i \rightarrow o$.

Definition 4.9: Let Bc_i be a basic concept, $\forall i = 1, \dots, m$. Let Bd be a base dimension, then $Bd = \bigcup_{i=1}^m Bc_i$.

XBRLDM defines a family of dimensions as a set of dimensions which are grouped by its semantics. In the MDM this will be named dimension group. A group of dimensions belongs to a domain, and each dimension of the group determines a domain, and this domain is the same as the group. Then, a dimension attribute can belong to several dimensions (really the concept) of the same dimension group.

Definition 4.10: Let D_i be a dimension, where $D_i \in D$. D is the set of dimensions, $\forall i = 1, \dots, n$. Let Do_k be a domain, where $D_i \rightarrow Do_k$, $\forall k = 1, \dots, m$. Let DG_j be a group of domains, where $DG_j \in DG$, DG is the set of dimension groups, $\forall j = 1, \dots, m$. Then $DG_j = \bigcup_{i=1}^n D_i$.

It is possible to deduce:

Rule 4.2: If $D_i \rightarrow Do_k$, $\forall i = 1, \dots, m$, $\forall k = 1, \dots, n$, and if $D_i \in DG_l$, $\forall l = 1, \dots, o \Rightarrow DG_l \rightarrow Do_k$, $\forall l = 1, \dots, o$.

On the other hand, in the XBRLDM, all defined domains must have a concept by default with semantic content (Hernández-Ros and Wallis, 2006; Eurofiling, 2011). Also in this data model, every dimension should have a concept by default of the domain to which the dimension belongs. Thus, in the MDM a concept by default in the domain must be defined, but the concept by default can be a calculated attribute or a dimension attribute, or both, normally will be a calculated attribute, as it is seen bellow.

Definition 4.11: Let Γ be the set of concepts defined in a taxonomy. Let ξ_i be a concept, and $\xi_i \in \Gamma_{def}$, $\forall i = 1, \dots, n$, where Γ_{def} is the set of concepts defined by default. Let Do_j be a domain, where $Do_j \in Do$, and Do is the set of domains, $\forall j = 1, \dots, m$, then $Do_j \rightarrow \xi_i$.

Definition 4.12: Let Do_i be a domain. Let D_{i_l} be a dimension, and let At_{i_l} be a dimension attribute, where $At_{i_l} \in Do_i$, and $D_{i_l} \rightarrow Do_{i_l}$. Let ξ_k be a concept, where $\xi_k \in \Gamma_{def}$ and $Do_i \rightarrow \xi_k$. Let CAt_l be a calculated attribute, then $At_{i_l} \rightarrow \xi_k$ or $CAt_l \rightarrow \xi_k$ and $\forall l = 1, \dots, O$, where $CAt_l \rightarrow (\xi_k, D_i)$.

In the XBRLDM a domain consists of dimensions and these of *domain-members*. In the MDM a domain consists of dimension attributes and the calculated attributes or measures of dimensions belong to a domain. These concepts are hierarchical (Hernández-Ros and Wallis, 2006; Schmehl and Ochocki, 2009; Santos and Castro, 2011a, c). In this data model, the hierarchies can be used for different validations of the concepts, and with a business perspective for IS. This means, that in the MDM the concepts (dimension attributes and measures) of a dimension are organized in an interconnected hierarchy tree. In the example the concept 'Real estate and no real estate' of the domain *DAssets_Estate* is a root of the concepts 'Real estate' and 'No real estate'. Each concept can have associated a comparison operation (the root) and an operation, "+", "-" (the leaves). Unlike the XBRLDM, the MDM uses calculated attributes to obtain a fact, but the XBRLDM does not calculate the facts, only their validations.

Therefore to obtain a mapping between the two models, a fact must carry out a certain rule of validation defined with respect to a calculated attribute. The validations from the XBRLDM hierarchies are used to take advantage of the Linkbase calculation (operation in the XBRL specification with only one dimension) (Engel et al., 2008; Santos and Castro, 2011a, c), however, the Eurofiling group in its guide of best practices recommends the use of the Linkbase of XBRL formulas (Morilla, 2008; XBRL International, 2009; Fischer, 2011). The formulas will be analysed in the next chapter.

The XBRLDM Dimension Taxonomy (XDT) defines two types of dimensions (Hernández-Ros and Wallis, 2006; Schmehl and Ochocki, 2009). The dimensions can be explicit and implicit. Explicit dimension attributes of dimension are defined in an

explicit way in the metadata model. A dimension is defined as implicit (typed in the XBRLDM) when its attributes of dimension are not explicitly defined in the metadata model, however they belong to a particular domain. In the MDM an implicit dimension their dimension attributes will be defined in run-time. If a dimension is implicit, there is no possibility of establishing hierarchies in the XBRLDM (*XBRL Dimensional Taxonomies*, XDT). This can be defined formally:

Definition 4.13: Let D_i be a dimension, $\forall i=1, \dots, n$, $D_i \in D$, where D is the set of dimensions. Let ξ_j be a concept, $\forall j = 1..m$. Let $Do_{i'}$ be a domain, where $Do_{i'} = \cup_{j=1}^m \xi_j$. Let At_j be a dimension attribute, where $D_i = \cup_{j=1}^{m'} At_j$. If $At_j \rightarrow \xi_j$ and $D_i \rightarrow Do_{i'}$, then D_i is explicit iff $D_i = \cup_{j=1}^{m'} \xi_j$, where $m' \leq m$.

Definition 4.14: Let ε_{o_i} be a data type, $\forall i=1, \dots, m$, $\varepsilon_{o_i} \in o$, where o is the set of data types of the specification or defined by the user. Let ξ_j be a concept, where $\xi_j \rightarrow Do_k$, $\forall k = 1, \dots, o$, and $Do_k \rightarrow \varepsilon_{o_i}$. $Do_k \in Do$, where Do is the set of domains. Let D_o be a dimension, $\forall o=1, \dots, m'$, and $D_o \in D$, where D is the set of dimensions. Where $D_o \leftrightarrow Do_k$, then D_o is an implicit dimension iff $\xi_j \in \varepsilon_{o_i}$ and $\xi_j \in Do_k$ and ξ_j is not defined.

Each concept is associated with 0 or an unknown number of references. The references are indications of legal texts. These references only indicate the reference to the law, directive, or circular and are not links to URL's or URI's (*Engel et al., 2008; Santos and Castro, 2011a, c*).

In the XBRL specification, tuples or arrays of data are allowed. However, the best practices guide developed by the *Eurofiling* group does not recommend them (*CEN, 2013; Eurofiling, 2015*). According to this group, the use of tuples complicates the use of the Linkbase of formulas (these will be analysed in the chapter 5), because it is impossible to indicate the element to validate in the array in a univocal way. In addition, this group shows that the extension of taxonomies over tuples is more complex, since the mechanism of adding concepts from tuples is technically more complex in XML. However, in the MDM an array is considered as another dimension.

In the XBRLDM, a fact must be defined as the same type of time period as the concepts involved (*Engel et al., 2008*) and therefore, this definition is mapped to MDM.

In the XBRLDM, a fact is defined as a set of pairs (*dimension / domain-member*) and a basic concept (*primary item*). In the MDM a table of facts consists of a set of facts, and these facts are determined by a set of pairs <dimension/dimension attributes>, including the base domain as an additional dimension, and/or with calculated attributes. For example in Figure 4.2 “BNP Paribas - The bank itself - Real Estate - 10,000.00” is chosen. This is equivalent to $F(5,1,1)$ in Figure 4.3. Then, the fact $F(5, 1, 1)$ is the union of <Entity, “BNP Paribas”>, <Assets_Estate_1, “Real estate”>, <Loans_1, “The bank itself”>, <Geography, Germany> and <Base dimension, Assets>.

Therefore, it is defined:

Definition 4.15: Let φ_i be a Fact, $\forall i=1, \dots, m$. Let D_j be a dimension, $\forall j=1, \dots, n$. Let At_{j_k} be an attribute of dimension, $\forall k=1, \dots, 0$. Let Bc_l be a basic concept $\forall l \in N$. Let $CAt_{j'_{k'}}$ be a calculated attribute, $\forall j' \in N$, and $\forall k' \in N$.
Then $Bc_l \cup (D_j, At_{j_k})_{j=1}^n \cup (CAt_{j'_{k'}})_{j'=0}^l \rightarrow \varphi_i$.

The *hypercubes* in the XBRLDM are constraints on the facts in the XDT (*XBRL Dimensional Taxonomies*), which indicates the valid combination of pairs <dimension, attributes of dimension>. A *hypercube* in MDM is a set of pairs <dimension, attributes of dimension> and calculated attributes defining one or more facts. In a *hypercube*, a calculated attribute can exist because this attribute is determined by a concept of a domain and the dimension, and is the result of operating with pairs <dimensions, attributes of dimension> and/or calculated attributes of a dimension.

Definition 4.16: Let φ be a Fact, $\forall i=1, \dots, m$. Let D_j be a dimension, $\forall j=1, \dots, n$. Let At_{j_k} be an attribute of dimension, $\forall k=1, \dots, 0$, where $At_{j_k} \in D_j$. Let $CAt_{j'_{k'}}$ be a calculated attribute, $\forall j' \in N$, and $\forall k' \in N$. Then a *hypercube* is $(D_j, At_{j_k})_{j=1}^{n'} \cup (CAt_{j'_{k'}})_{j'=0}^l \rightarrow \varphi$.

An allowed *hypercube* is defined as a *hypercube* associated with a basic concept that determines a fact. A *forbidden hypercube* is defined as a *hypercube* associated with a basic concept that cannot determine any fact, because the expert user considers to this fact as impossible or erroneous. In the XBRL specification is compulsory to define all the *hypercubes* of the facts shown in the report. As such it can be defined as:

Definition 4.17: Let φ_i be a Fact, $\forall i=1, \dots, m$. Let D_j be a dimension, $\forall j=1, \dots, n$. Let At_{j_k} be an attribute of dimension, $\forall k=1, \dots, o$, where $At_{j_k} \in D_j$. Let $CAt_{j'_{k'}}$ be a calculated attribute, where $\forall j' \in N, \forall k' \in N$. Let Bc_l be a basic concept, $\forall l=1, \dots, p$. Then a hypercube is allowed iff $Bc_l \cup (D_j, At_{j_k})_{j=1}^{n'} \cup (CAt_{j'_{k'}})_{j'=0}^l \rightarrow \varphi_i$.

A forbidden hypercube is defined as a hypercube associated with a basic concept that does not determine any fact.

Definition 4.18: Let φ_i be a Fact, $\forall i=1, \dots, m$. Let D_j be a dimension, $\forall j=1, \dots, n$. Let At_{j_k} be an attribute of dimension, $\forall k=1, \dots, o$, where $At_{j_k} \in D_j$. Let $CAt_{j'_{k'}}$ be a calculated attribute, $\forall j' \in N, \forall k' \in N$. Let Bc_l be a basic concept, $\forall l=1, \dots, p$. Then a hypercube is forbidden iff $Bc_l \cup (D_j, At_{j_k})_{j=1}^{n'} \cup (CAt_{j'_{k'}})_{j'=0}^l \rightarrow \emptyset$, where $\varphi_i \neq \emptyset$.

Figure 4.4 shows the star model of the MDM in the PIM. Each dimension has the following attributes: *Identifier* (or short name), *name* (or description of the concept), *time period type*, *references* (It can be null), *default* (if the concept is by default). The base dimension or dimension of basic concepts has the following attributes: *Identifier* (or short name), *name* (or description of the concept), *time period type*, *balance* (it can be null). The fact table has the following attributes: *Identifier* (primary key, pk), *reference to the dimension* (foreign key, fk), *unit* (Euros, Pounds, dollars,...), *Precision*, *time* and *fact*. The unit and the precision can be null, because the fact can be non-numeric. In the fact table, the set of external dimensions and the base dimension are associated with a *unique key* (uk).

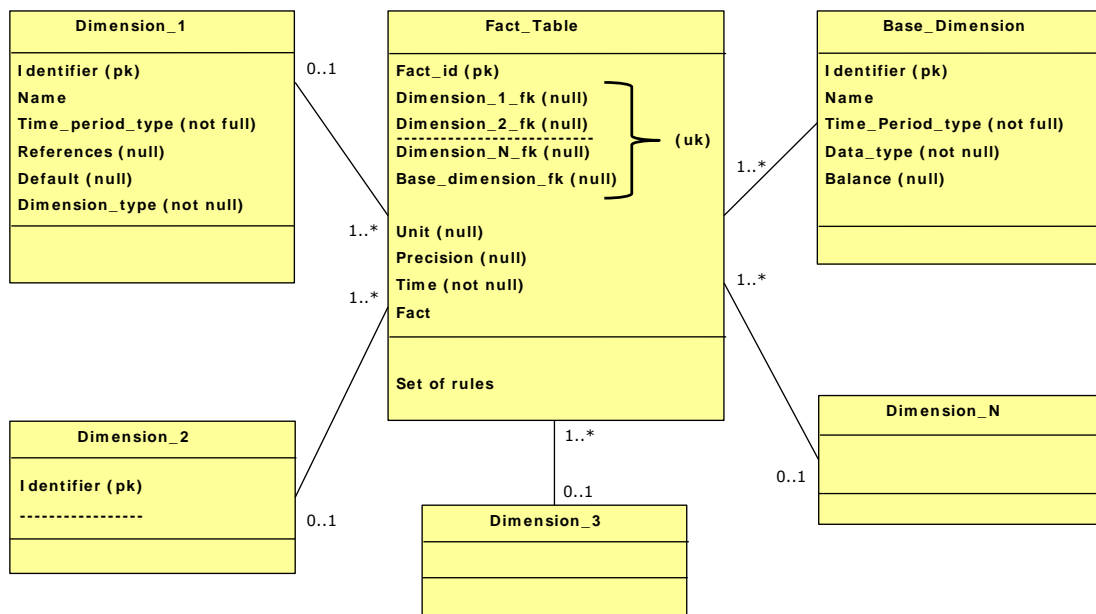


Figure 4.4 Star model in the PIM

The first algorithm depicts the extraction of the metadata from the CIM to the PIM:

```

start
  begin
    read template;
    repeat
      Gather concepts; Gather domains; read templates;
    until no more templates?;
    repeat
      Review domains; Review concepts; Gather basic concepts;
    until no redundancy?;
    repeat
      Establish root and leaves in tree-like structure of the concepts;
    until Is the tree-like structure correct?;
    repeat
      Gather dimensions; Gather attributes of dimensions;
    until Is it semantically correct?;
    repeat
      Gather calculated attributes;
    until Does the tree-like structure agree with the domain?;
    gather allowed facts; Gather forbidden facts;
    add new calculated user attributes (user constraints);
  end
end

```

Algorithm 4.1 Transformation from the CIM in the PIM

Concepts and domains are gathered from the templates in the spreadsheet and are then checked to avoid redundancy. The *basic concepts* are obtained in this phase. Then, the tree-like structure of the concepts in the domain is established, except in the domain of basic concepts. Now, it is necessary to verify that the trees are well built. Then, dimensions and attributes of dimension are collected. It is also necessary to verify if the semantics of the dimensions are correct (in a domain there are 1..* dimensions with attributes of dimension that are concepts in the same domain. A concept can belong to several dimensions of a domain). The next step is to obtain the calculated attributes and check that they respect the tree-like structure of the concepts in the domain. Then, the allowed and forbidden facts are obtained, and finally, the new calculated attributes defined by the user are acquired from different dimensions.

Up to this point all necessary elements for the implementation in a specific database are defined. *Figure 4.5* shows the fact table of *Figures 4.2* and *4.3*, the star model in the PIM, where 'To Other' is 'To other banks', 'RE' is 'Real Estate', and 'NRE' is 'No Real Estate'.

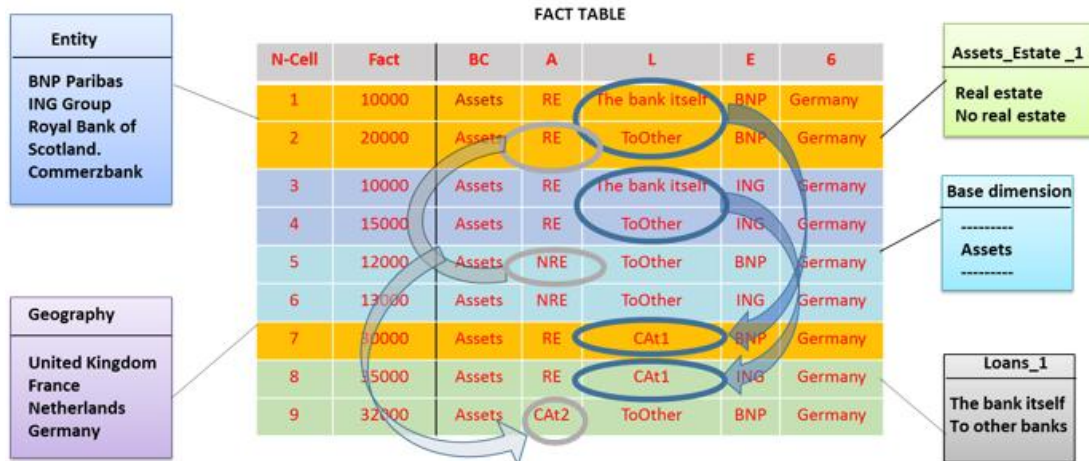


Figure 4.5 Example of fact table with calculated attributes

As it is explained above in this model there are two calculated attributes: $CAt_1 = ('To\ bank\ itself' + 'To\ other\ banks') |_{Loans_1}$, $CAt_2 = ('Real\ Estate' + 'No\ Real\ Estate') |_{Assets\ Estate_1}$. Finally, it is possible to analyse that Fact 7 is correct, and 8 is wrong. Moreover, it is possible to see an allowed hypercube as Fact 7 that is defined as $\{(BD, Assets), (Assets\ Estate_1, Real\ Estate), Cat1, (Entity, BNP), (Geography, Germany)\}$.

Figure 4.6 gives a summary of the set of objects defined in the MDM in UML. In this figure, it is possible to see that a domain consists of concepts, and a concept can be a basic concept or a non basic concept. A domain can have dimensions. An attribute of dimension can belong to different dimensions. A normal concept, which can be an attribute of dimension or a calculated attribute, determines a domain, but, it belongs to a dimension and a domain. A hypercube consists of pairs (dimension and dimension attribute) and/or calculated attributes. The concepts are either user type or standard type.

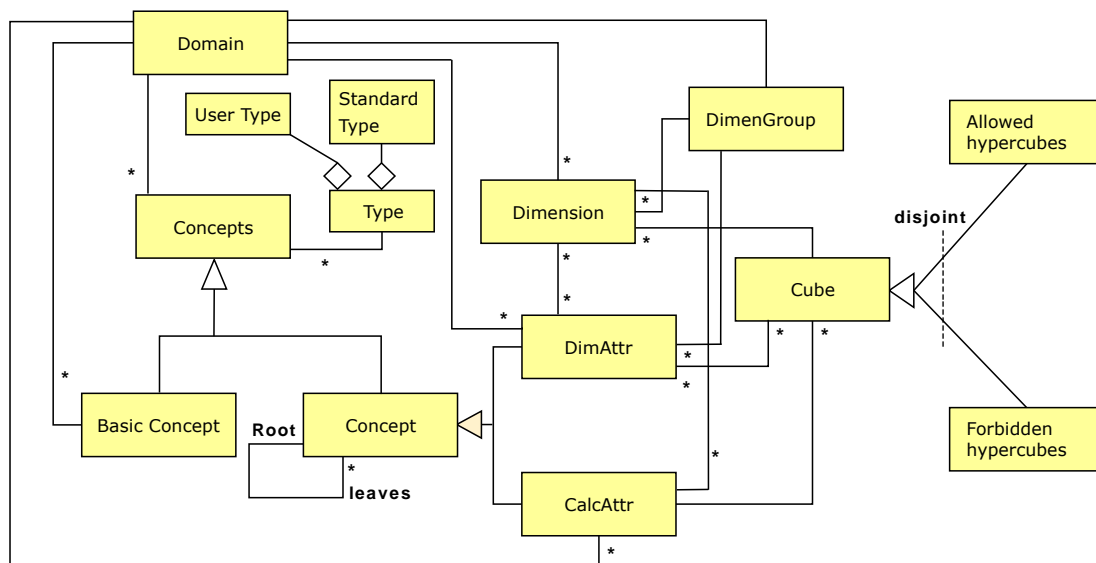


Figure 4.6 Summary of the artefacts in UML of the data model in the PIM

Until here, it is shown the structure, definitions and user constraints of the PIM. Now, in the next section, it is necessary to verify some constraints of the design in this platform.

4.4. Semantic Questions about the Design of Metadata in the PIM

In this section, certain questions associated with these semantic reports in this level, the PIM, are analysed. It is the aim of the new study and the thesis to obtain solutions for these questions. However, if these questions are not resolved in the MDM (the PIM), they will be inherited in the SQL (the PSM) and will produce errors at run-time or incorrect results. Firstly, the hierarchies are analysed. Secondly, the problems of the definition of the type of time period are considered. Thirdly, questions about implicit dimensions are discussed. Finally, the cases when several calculated attributes affect one fact are discussed.

Hierarchies are used in XBRLDM to give semantics to the metadata. A dimension in XBRLDM is composed of concepts (in the MDM, dimension attributes and calculated attributes or measures of the dimension), and these concepts are in a hierarchical tree structure, with only one root, that in general is the concept by default. In the first approach it is necessary to check if only one root exists and the structure does not

have cycles. It be test that concepts without leaves do not have any validation operations.

The type of time period of the attributes of dimension and of the calculated attributes of the same domain must have the same data type. But, in the same way, a basic concept also has a type of time period and must have the same data type. A fact is the union of a basic concept with the union of pairs *<attributes of dimension/dimension>* and the calculated attributes. How is it possible to ensure that both types of time period are the same? And, if they are not the same type, what is the correct time period? When the periods have different data types, one solution of the IS analyst is sometimes to create another dimension of time and to avoid problems in the transformation of the data in the presentation of these, or as input to other systems.

Although the attributes of dimension of the implicit dimensions are not validated in the XBRLDM, they should at least be verified if these attributes are to correspond with the specified type and depending of the result a message can be shown. The attribute of implicit dimension is filled out when the report is made, and they are not defined in the set of concepts of the taxonomy (in the metadata). Thus, for example, if the dimension is a financial entity, it must be tested in the report (*XML instance or XBRL instance document*), and the value should be a set of characters. From a semantic point of view, it could even check that the name of the entity has a structure regulated by a standard, or if it can be predefined, and in this case can be checked in *run-time*.

One difference between the MDM and the XBRLDM is on the topic of validation. In the MDM a fact can be obtained through a calculated attribute. However, in the XBRLDM a fact is not calculated, it is only validated. If the specification is mapped to MDM then a calculated attribute is used for validation. There is one more difference: a fact can be verified by one or more calculated attributes.

4.5. Analysis in the Platform Specific Model (PSM)

This section analyses the mapping from the PIM to PSM. *Table 4.2* shows the mapping from the PIM to PSM. In this platform some constructors are not transformed, as domains or concepts, because they are not used. Column 1 shows the concept in

the PIM. Column 2 displays the mapping in the PSM. The final column gives some comments.

Table 4.2 Mapping from the PIM to the PSM

PIM	PSM	Comments
Base dimension	Table	The PSM does not care if it is a base or normal dimension.
Basic Concept	Dimension attribute	Has an associated data type, a time period type and a balance type. The balance is a constraint.
Dimension (explicit/implicit)	Table	Explicit if the dimension attributes are defined, and implicit if they are not defined.
Calculated attribute	Calculated attribute	An aggregate of the leaves of the dimension concepts. This rule obtains calculated attributes.
Dimension attribute	Dimension attribute	Not an aggregate.
References	Dimension attributes	References of the concepts.
Facts	Fact table	Implicit in the design.
Fact::= - Dimension/(Dimension attribute) - Basic concept - Calculated attribute	Fact	Includes dimension attributes and calculated attributes (measures).
Allowed fact	Assertion	Constraint of the expert user.
Forbidden fact	Assertion	Constraint of the expert user.

This table shows the transformation of *Figure 4.4* (the MDM) into a relational model, using ROLAP technology. In the first phase the tables and dimension attributes are obtained. Next, assertions and rules are defined. A set of assertions are defined for validating the facts or measures that are allowed or forbidden. Moreover, calculated attributes are transformed into rules. *Algorithm 4.2* depicts the obtaining of the calculated.

```

start
  read dimensions;
  repeat
    read concepts of the dimension of the domain;
    repeat
      if the concept has no associated comparisons
        then the concept is a dimension attribute;
      else if the concept has no leaves
        then the concept is a dimension attribute;
      else begin
        the concept is a calculated attribute;
        read comparison operation;
        read leaf concepts and associated operations;
      end
    read concepts of the dimension of the domain;
  until no more concepts of the dimension;
  read dimensions;
until no more dimensions;
end

```

Algorithm 4.2 Implementation of the metadata model

Until here all necessary elements for the implementation in a specific database are defined. The POCs of the author of this thesis is shown in *Openfiling (2015)*, *Santos and Castro (2011b; 2012a)* and *Santos and Nieto (2014, 2015)*.

CHAPTER 5

Fundamentals and Basic
Concepts of the proposal
of formulas

CHAPTER 5. Fundamentals and Basic Concepts of the proposal of formulas

5.1. The Universe of the Discourse

This chapter focuses on research into the mapping between the XBRL data model (XBRLDM) and the multidimensional data model (MDM), as well as its automation (Piechocki et al. 2007; Felden 2007).

MDM is a straightforward model that combines objects, dimensions (hierarchies), measures and attributes for representing real work business problems (Kimball, 1996-2004; Inmon, 2005; Jarke et al., 2003). In addition, this model is the heart of *On-Line Analytical Processing* (OLAP), which requires complex queries that can be solved by the MDM in real time.

Conversion between the aforementioned models will be made using the *Model Driven Architecture* (MDA) paradigm, which ensures interoperability and solves the problem of heterogeneity between systems. The MDA has three phases or levels (OMG, 2015): *Computation Independent Model* (the CIM), *Platform Independent Model* (the PIM) and *Platform Specific Model* (the PSM). As is analysed in more detail below, using reverse engineering a conceptual model (the PIM) is obtained from the *Formula Linkbase* of the XBRL specification (the CIM). Following this, the logic model is obtained from the conceptual model in the SQL standard (the PSM). At the end, the code is obtained.

UML/MDA (OMG, 2015) is a powerful tool that has helped in different areas of *Information Technology* (IT) to model structured and robust systems. However, the techniques of verification and validation of the software is not supported in the standard MDA (Cuadra et al., 2011; Aljumaily et al., 2014). For this reason, the main contribution of this thesis related to Information Systems is focused to the automatic transformation of the models, formalising validation rules in each phase, and with the target for validating a high number of rules, which represent hundreds of concepts, dimensions, attributes of dimension, cubes, etc. In addition, this research work shows a new demonstration of software validation that can be applicable to any field in IT, where the semantic interoperability is required.

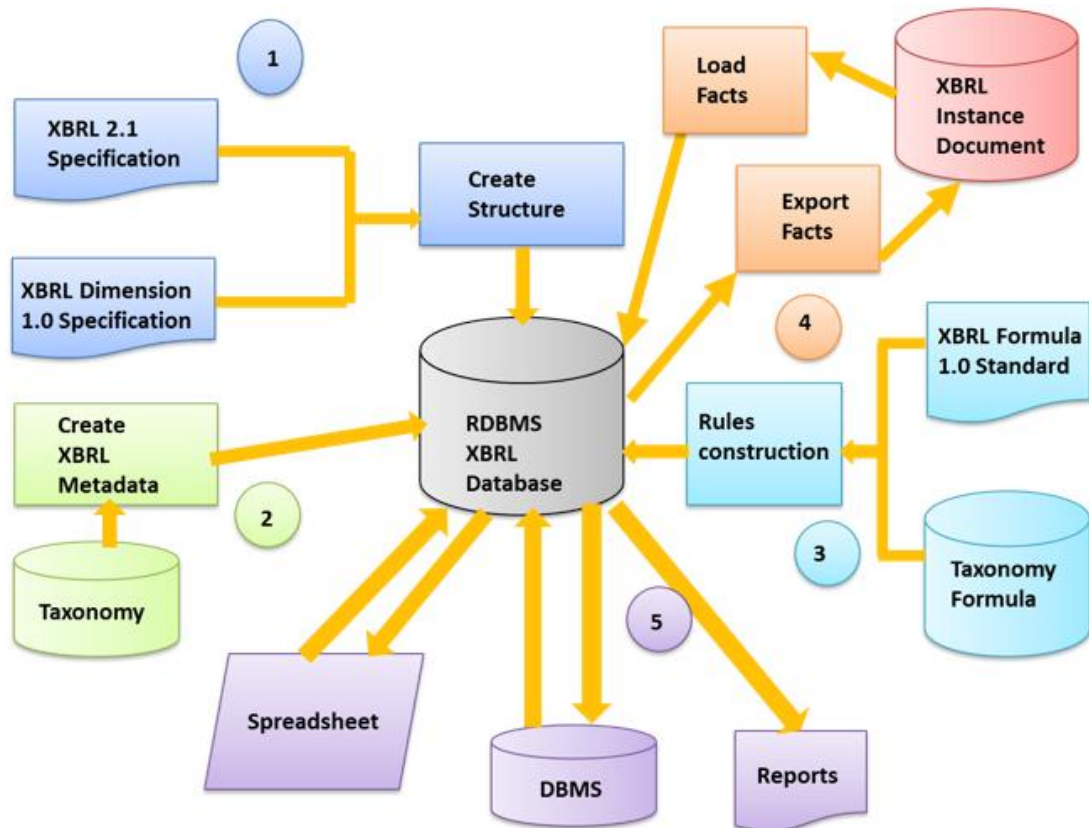


Figure 5.1 General Structure of the transformation XBRL vs. RDBMS

Figure 5.1 presents the general structure of the proposal to be discussed in this chapter, enumerating and color-coding the different steps in the mapping process. In Step 1, the structure is obtained from XBRL 2.1 Specification (*Engel et al. 2008*) and XBRL Dimensions 1.0 (*Hernández-Ros and Wallis, 2006*), and the entities, attributes and relations from the XBRLDM are created (*Santos and Castro, 2010, 2011b, 2011c*). Among these entities are: *Dimensions* (explicit dimensions in the XBRLDM); dimension attributes, *DimAttributes* (members in XBRLDM); the link between dimensions and dimension attributes, *Relation_Dim_AttrDim* (arc in XBRLDM); basic concepts or *PrimaryItems* (primary items in XBRLDM); group of dimensions or *Context* (the same in XBRLDM); the relationship between context, dimensions and dimension attributes, *Context_Dim_AttrDim* (arc in XBRLDM); the unit of measure, *Unit* (the same in XBRLDM); the set of instance document headers (used to map from the RDBMS to an XBRL instance document), *Header*; and, finally, that used for a common special format for the end user, *DataPoint* (really is the fact table).

In Step 2 of the process (*Figure 5.1*), the XBRL metadata is created by loading the database with dimensions, dimension attributes and relationships (*Santos and Castro, 2011b*). For this end, this research work creates an API that generates the SQL sentences from the taxonomy. Step 3 of the process corresponds to the generation of the validation framework which, as it is the focus of the new proposal, is explained in much greater detail in subsequent sections of the paper.

In Step 4, data is loaded from the XBRL instance document into the database. According to this process, the instance document is read and the database is filled with facts and contexts. Similarly, a parallel process is commenced for the creation of an XBRL instance document from the database. Finally, Step 5 obtains the mapping from database facts to other tools (and vice versa) commonly used in IT, as well as by end users and regulators.

The next section will analyse XBRLDM. The introduction of the XBRL specification is studied in section 3. Section 4 proposes the MDA architecture, definitions and mathematical formalisation in the MDM from the XBRLDM. Section 5 analyses some semantic questions of the XBRL formula specification. Finally, section 6 shows the implementation of this research through Proofs of Concept (POCs) and the analysis of the phases of the PIM and the code. The validation of this proposal will be in the following chapter.

5.2. XBRL Data Model and calculations

XBRL semantic information, separated from application software, uses and extends the XML standard. A report or XBRL instance document references a set of XML or XBRL Schemas. This set of schemas in the XBRLDM, called a *Discoverable Taxonomy Set* (DTS), specifies the economic concepts needed. Each *XBRL Schema* has *roles* (*linkbases* in the XBRLDM), in the chapter 3, section 1 were analysed (*linkbases* in the XBRLDM). The role, *Calculation* provides simple calculation relationships between different elements (basic concepts, dimensions and dimension attributes), but does not allow formulas or complex expressions. Valid combinations of dimensions are permitted by the role, *Dimension*; while *Formula*, the mapping of which constitutes the new contribution of this paper, supports complex expressions

based on XPath which can be applied to instance documents for the validation of information.

A XBRL instance document consists of a *header* with the *XML Schemas* referenced, as well as the associated roles and dimension set grouped by *context*, and the *items* (or facts in the MDM). For each *item* (in the XBRLDM) in the instance document, dimensions and *primary item* (basic concept in the MDM) are associated through the context. Each XBRL or XML Schema may have 0, 1 or *n* roles depending on the role type. In this specification, a *primary item* in the XBRLDM defines a dimensionless fact schema in the MDM (Hernández-Ros and Wallis, 2006; Martín Quetglás, 2006; Engel et al., 2008; Schmehl and Ochocki, 2009). A basic concept defines a fact schema, but with the unit of measure, type and, if it is numeric, a precision of a specific number of decimal places. For example, the primary item, *Asset*, could belong to the *equities* or *risk* dimension. The context contains information about a fact or set of facts. Furthermore, it is associated with an entity or financial institution for a specified period and includes a meaning for businesses through a *segment* and a *scenario* (both being dimensions sets in the MDM). The *scenario* shows the specific *dimension-dimension attribute* pairs from business logic.

5.3. Introduction to XBRL formula specification

In this section, the XBRL Formula specification is analyzed from the MDM, and a set of rules and definitions are developed in next sections.

This specification defines a set of validations over certain given facts (XBRL International, 2009). These definitions have syntax and a set of rules. Prior to the release of XBRL 2.1 (Engel et al., 2008), the version currently used, the previous version, XBRL 1.0 (Hamscher and Kannon, 2000), was based on hierarchies with a tree structure, since XML, the language on which it was based, is hierarchical. Thus, XBRL, prior to the 2.1 version operated with a set of limitations. An example of a XBRL instance document generated under these limitations is presented in Figure 5.2 (XBRL International and Novartis International AG, Samples XBRL 1.0, 2002), consisting of a set of basic concepts with a set of *dimension-dimension attribute* pairs (Santos and Castro, 2011a, 2011c). The example is presented graphically in Figure 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
Total long-term assets		32585	25257
Current assets			
-----		----	----
-----		----	----
Total current assets		34200	32939
TOTAL ASSETS		66785	58196

Figure 5.2 Example of hierarchical report

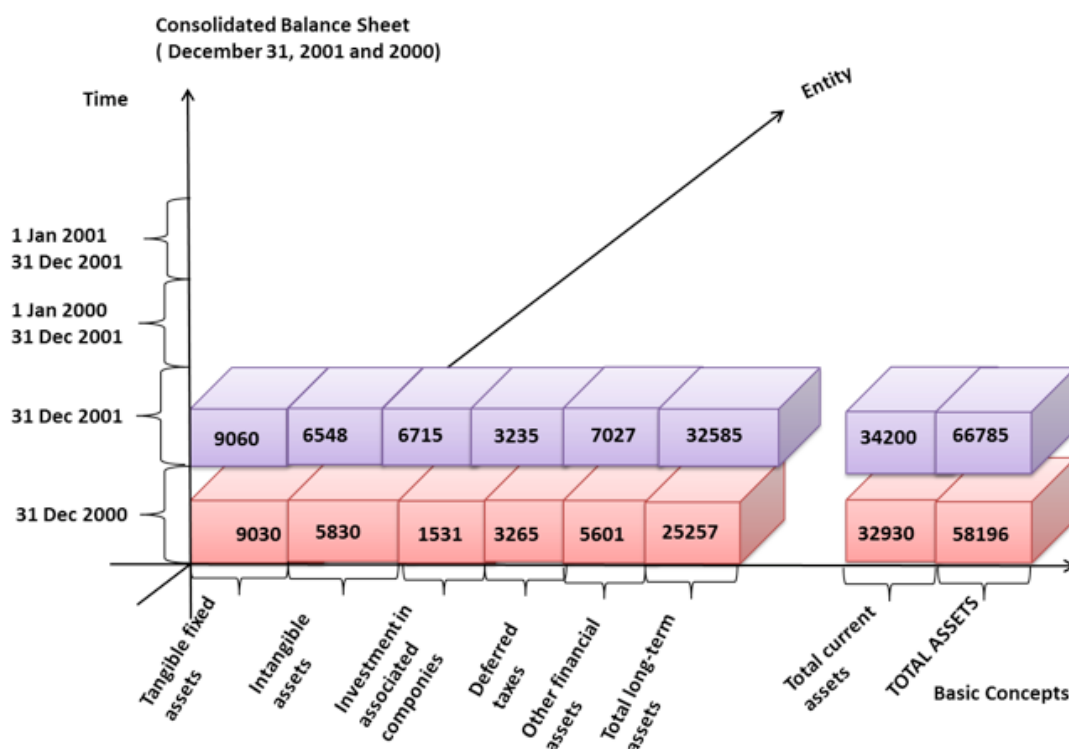


Figure 5.3 Graphic design of the report of the example of Figure 5.2

In Figure 5.4, the same report is displayed with its hierarchical structure. However, with XBRLDM 1.0, dimensions are not used exhaustively and only hierarchies are used. Despite this limited use of dimensions, operating with different dimensions or, in other words, different contexts is not possible. For example, while in Figure 5.3 it is possible to compare two facts on the same date (i.e., same context), it is not possible to compare two facts from two different dates. Nevertheless, there is a dimension in each level of the hierarchy with dimension attributes and basic concepts. In general, each branch of a hierarchy is the sum of its branches. In other words, it is possible to operate with basic concepts that have the same dimension-dimension attribute pair set (i.e., same context) (Figure 5.4). Thus, it is possible to find *total assets* to be equal to the sum of *total long-term assets* and *total current assets*, however, only for December 31, 2001 and not between December 31, 2001 and 2000. For example, it is not possible to validate *total assets* for December 31, 2000 as being the sum of *total long-term assets* for December 31, 2001 and *total current assets* for December 31, 2001, as shown in Figures 5.3 and 5.4.

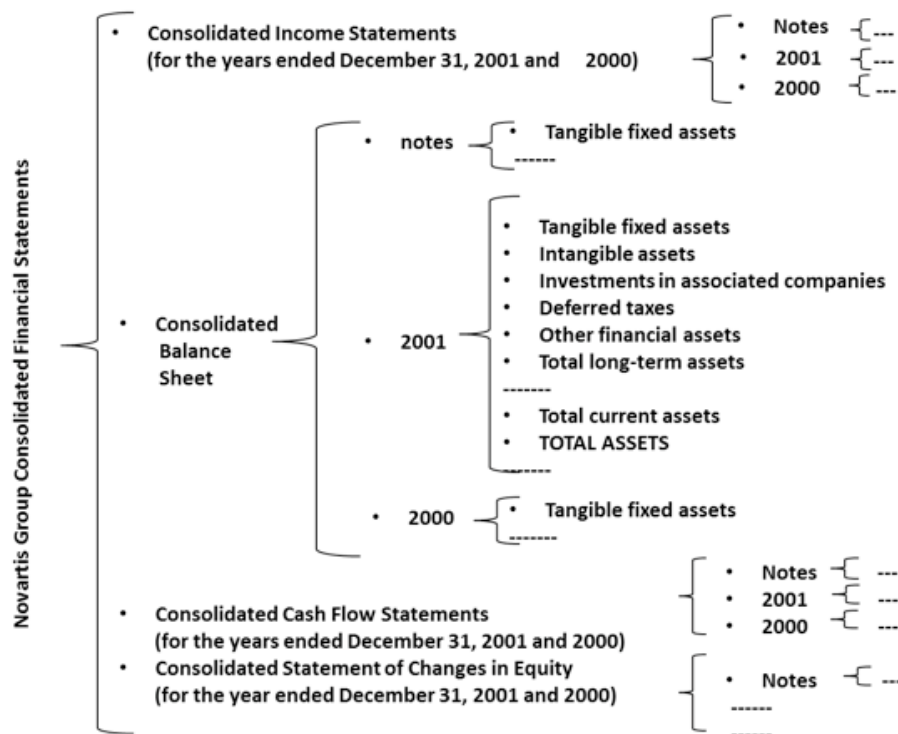


Figure 5.4 Hierarchical structure of the report of Figure 5.2

Whereas *Calculation* works in hierarchies (XBRLDM) or, in other words, with a single context, it does not work, however, with multiple contexts (Santos and Castro, 2010, 2011a, 2011c). In XBRL 1.0, the dimensions are *entity*, *period of time*, *basic concepts* (i.e., set of basic concepts) and a set of dimensions defined by the scenario (0..n dimensions) and the segment (1 dimension). The current version of the standard, XBRL 2.1, was created in direct response to business users requiring the inclusion of more information with more complex structures and validation rules (Hernández-Ros and Wallis, 2006; Engel et al., 2008).

Validation constitutes an important aim in any XBRL instance document, and arithmetic and logical operations, as well as functions based on previous operations, are necessary in any validation. The most important advance in XBRL formula specification, however, is the ability to perform operations with different facts with different dimensions or contexts. If the validation is performed in the destination, any language, including COBOL, Java or .Net, may be used. However, as validation occurs at the document origin, a standard is necessary. These were provided with the publication of XBRL Dimensions 1.0 specification (Hernández-Ros and Wallis, 2006) and XBRL Formula 1.0 Specification (XBRL International, 2009), respectively.

5.4. Definitions and semantics of XBRL formula specification

This thesis proposes an approach of the MDA in the mapping between XBRL formulas and a database (*Figure 5.5*).

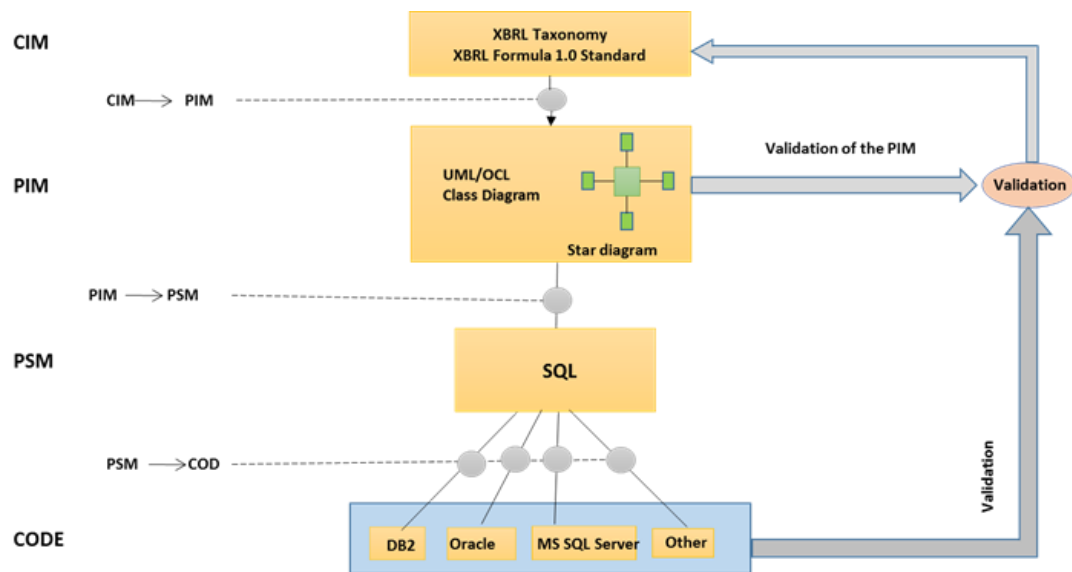


Figure 5.5 Automatic conversion of formulas between XBRL instance documents and Databases using MDA

The CIM model in this research work is the set of XBRL instance documents, the taxonomies and XBRL Formula Specification (sections 5.2 and 5.3). In this section, rules and definitions from the CIM are obtained, through XBRL Formula Specification. The PIM used in this proposal is based on UML, which is a star model, the MDM (*Boixo and Flores, 2005; Piechocki et al., 2007; Felden, 2007; Schmehl and Ochocki, 2009*). The set of constraints, dimensions and dimension attributes are collected in an automatic way from a taxonomy and its algorithm is shown in *Figure 5.12*. The PSM is a set of stored procedures or programmes in Cobol, C++, etc. The algorithm will show the mapping from the PIM to the PSM. In the *Code 5.1* an example of code SQL is depicted.

The process of validation is divided in **two phases** (*Figure 5.5*). The **first** phase is to test the UML star model / MDM (the PIM), from the XBRL taxonomies and the XBRL

reports, and the **second** one is to validate the set of stored procedures using ROLAP technology. The star model is validated with a comparison among the dimensions, dimension attributes, and constraints in the MDM and the XBRLDM (*Chapter 6*).

In this section, a set of eleven definitions and rules used in this specification are proposed, but in the MDM from the XBRLDM. As the proposal is based on the MDM (or star model in UML, the PIM), the author believe that it simplifies the concepts and rules, making them more comprehensible for users. Additionally, this section presents a new dimension to the proposal by offering a novel mathematical formalization (formal language in the MDM). Finally, a more detailed example is given at the end of the section.

The formulas below are located in one or several documents, with each document consisting of assertions. In the *Oxford English Dictionary* (OED, 2010 ed.), *assertion* is a confident and forceful statement of fact or belief. Then, each XBRL instance document can have 0..* XML documents of the role formula, each of the latter consisting of a set of (1..*) assertions. The assertions are used to verify an expression or condition. The following paragraphs present the new definitions used in the XBRL formula specification.

Definition 5.1: *A fact is a value representing a particular measurement provided by the reporting entity.*

Definition 5.2: *An assertion is a general constraint that does refer to a fact or set of facts.*

A formula document consists of a set of groups of assertions. Although each group of assertions can belong logically to a specific XBRL instance document, it is not possible to know for sure if this is the case. Each definition of *assertion* should have one name in an XML document of formula. The assertions are grouped into sets, each of which being logically related to an XBRL instance document and starting with the tag *assertionSet*. Thus, each instance document must be validated against all assertions of all formula documents, whether or not they belong to the group of assertions from the XBRL instance document. For this reason, it is very common to find XBRL instance documents that must be validated with a large set of assertions not used in the instance document.

Definition 5.3: *A filter is set of dimensions whose assertion is confined.*

A filter is a set of *dimension-member* pairs in the XBRLDM or of *dimension-dimension attributes* pairs in the MDM (XBRL International, 2009a, 2009b).

Definition 5.4: *An implicit filter is a constraint inherent in all facts circumscribed by an assertion (i.e., the assertion is constrained to the facts circumscribed by the set of dimension-dimension attribute inherent pairs.).*

The implicit filters as inherent constraints are not written in the assertion. By default, there are always implicit filters in an assertion in the XBRLDM save in the case where, the parameter of the assertion *@implicitFiltering* being false, the default is true. The implicit filter consists of a set of *dimension-dimension attribute* pairs which are common to all facts of the assertion. The implicit filters by default are *entity*, *period of time* and *unit*.

Definition 5.5: *An explicit filter is a semantic constraint to which all facts, groups of facts or assertion facts are circumscribed.*

According to the syntax of the assertion, the constraint can be circumscribed to different environments in the assertion.

In the OED, concept is an idea or mental image which corresponds to some distinct entity or class of entities, which corresponds to the entity or entities' essential features or which determines the application of a term (particularly a predicate) and thus plays a part in the use of reason or language. In the XBRLDM, a concept is that which that defines a dimension, a dimension attribute or a basic concept.

Definition 5.6: *A basic concept defines the name of a concept by type, precision and unit of measure (Santos and Castro, 2011c).*

In the instance document for the example report from *Figures 5.2, 5.3 and 5.4* (XBRL International and Novartis International AG, 2002), *intangible assets* is a basic concept insofar as it defines a financial concept, has a monetary type and a unit of measure which, in this case, is *iso4217:CHF*. In ISO 4217, published by the International Standards Organization, the currency of Swiss francs (used in

Switzerland and Liechtenstein) is defined with the country code *CHF*. Furthermore, the basic concept has a precision indicating the number of decimal places to be used in the calculation, for example, *precision='18'*.

A *basic concept* can be analysed as an attribute of the dimension *primary Items* or *basic concepts*.

Definition 5.7: *A constant is a value or expression yielding one result and whose expression is deterministic.*

A constant can be, for example, *var_BasicThreshold4* where *var_BasicThreshold4* is the name of the constant. Although it may have a value of 3 or $(4 \text{ idiv } 2 + 1) * 1000$, the value will always be unique. The *idiv* operator indicates division without decimals.

An assertion consists of the assertion definition, parameters, constants, preconditions, variables and filters. An assertion has an input and output, with the former consisting of the facts to be validated and the latter being the result of the validation.

Definition 5.8: *A variable is an atomic element of an expression in an assertion which depicts a fact or group of facts.*

A variable can consist of dimensions, dimension attributes, constants, preconditions and other formulas (Morilla, 2008; XBRL International, 2009c; Fischer, 2011). Each variable in each assertion has one name which begins with \$. The dimensions and their attributes constitute the explicit filters. A variable has parameters in the XBRLDM which, in the MDM, are variable attributes since they define variable properties.

From these definitions, the following rules can be developed:

Rule 5.1: *A variable can have explicit dimensions.*

Let v be a variable. Let DE_i be a dimension $\in D$, where D is the set of dimensions and $\forall i = 1, 2, \dots, n$. Let At_j be an attribute of dimension $\in DE_i, \forall j = 1, \dots, n$. Then v is $\langle DE_k At_l \rangle_{k=0}^n, \forall l = 1, \dots, n$.

Rule 5.2: An assertion can have implicit dimensions (i.e., those inherent to the design) which are concealed in the assertion.

Let δ be an assertion. Let DI_i be a dimension $\in D$, where D is the set of dimensions, $\forall i = 1, \dots, n$. Let At_j be an attribute of dimension $\in DI_i, \forall j = 1, \dots, n$.
Then δ has $\langle DI_k, At_l \rangle_{k=0}^n, \forall l = 1, \dots, n$.

Rule 5.3: An assertion can have explicit dimensions which appear in the assertion as an explicit filter in the XBRLDM.

Let δ be an assertion. Let DE_i be a dimension $\in D$, where D is the set of dimensions $\forall i = 1, \dots, n$. Let At_j be an attribute of dimension $\in DE_i, \forall j = 1, \dots, n$.
Then, δ has $\langle DE_k At_l \rangle_{k=0}^n, \forall l = 1, \dots, n$.

Rule 5.4: A variable, with its attributes, explicit filters and general filters (implicit or explicit) of the assertion, determines from 0 to n facts.

Let δ be an assertion. $\langle DI_k, At_l \rangle_{k=0}^n, \forall l = 1, \dots, n$, where DI are the implicit dimensions $\in \delta$. Also, $\langle DE_k At_o \rangle_{k=0}^n, \forall o = 1, \dots, n$, where DE are the explicit dimension $\in \delta$. Let v be a variable, and $\langle DVE_k At_p \rangle_{k=0}^n, \forall p = 1, \dots, n$, where DVE are the explicit dimensions $\in v$. Let φ be a fact $\in v$.
Then, $\langle DI_k At_l \rangle_{k=0}^n$ and $\langle DE_k At_o \rangle_{k=0}^n$ and $\langle DVE_k At_p \rangle_{k=0}^n \rightarrow \rightarrow \varphi$ or is \emptyset .

Rule 5.5: An assertion validates a set of facts through an expression.

Let X be a regular expression. Let σ an expression, where $\sigma \in \text{Boolean data type}$.
The operation of $X, \Theta(X) \rightarrow \sigma$, and σ is a deterministic result. σ is true or false.

Definition 5.9: The parameter *fallbackValue* (in the XBRLDM) or variable attribute (in the MDM) defines a value by default for a fact or set of facts if the latter is null.

Through the implicit and explicit filters, a variable defines a fact or a set of facts. If a fact or a set of facts is null, then it has the default value of the parameter *fallbackValue* for the given variable if *fallbackValue* is not null. An example of this is shown below.

If the fact or set of facts obtained from a variable is null and the attribute *fallbackValue* is also null, then the assertion is not validated.

From these definitions, the following rules can be inferred:

Rule 5.6: *If the fact obtained by the variable is null and the attribute of the fallbackValue variable exists, then fact has the value of fallbackValue.*

Let v be a variable. Let φ_h be a set of facts $\in v, \forall h = 1, \dots, n$, and $fallbackValue \neq null$, $fallbackValue \in v$, and $\varphi_h = null$, then $\varphi_h ::= fallbackValue$.

Rule 5.7: *An assertion validates a set of facts through an expression. An expression consists of a set of variables which perform operations between them. The result of the assertion is Boolean, that is, either true or false.*

Let δ be an assertion and X a regular expression.

Let σ be an expression, where $\sigma \in \text{Boolean data type}$. Let $v_i, \forall i = 1, \dots, n$, a set of variables, where $v_i \in \delta$. where $X = (v_1, \dots, v_n)$. The operation of $X, \Theta(X) \rightarrow \sigma$. σ is true or false.

Rule 5.8: *An assertion validates a set of facts through an expression. An expression consists of a set of variables, filters (i.e., dimension constraints) and constants that perform operations between them and determine a set of facts. If the facts do not exist or a fact in the expression is null, the assertion is not evaluated.*

Let δ be an assertion.

Let X be a regular expression and $X \in \delta$. X is the expression to be validated.

Let v_j be a set of variables, where $v_j \in \delta$ and $\forall j = 1, \dots, m$.

Let φ_h be a set of facts $\in v_h, \forall h = 1, \dots, n$. Then, δ is true if $(\Theta(v_1(\varphi_h), \dots, v_m(\varphi_h)))_{h=1}^n \rightarrow \text{True}$. If not, δ is false where $X = (\Theta(v_1(\varphi_h), \dots, v_m(\varphi_h)))_{h=1}^n$. $\forall \varphi_h$, where $h = 1, \dots, n$. But if $\varphi_h \rightarrow \emptyset$ then $\Theta(X)$ is not evaluated.

Definition 5.10: *The parameter Precondition, or variable attribute or assertion, defines the facts that can be evaluated.*

If the fact or set of facts obtained in a variable or an assertion does not fulfil the precondition, it is not evaluated.

From this set of definitions, the following rules can be inferred:

Rule 5.9: *A variable precondition is an expression that evaluates the facts obtained by the variable. If the fact fulfils the precondition, the fact is evaluated in the assertion. Otherwise it is not evaluated. In the latter case, the assertion is evaluated, but without the facts that do not fulfil the precondition.*

Let δ be an assertion and let v be a variable, where $v \in \delta$.

Let Y be a regular expression. Let φ be a fact, where $\varphi \in v$.

Let Z be a precondition, where $Z \in Y$.

If φ carries out Y , then φ is evaluated. Otherwise, φ is ignored.

This can be explained through propositional logic, namely:

Let preconditionRule(φ, Y, Γ) be a rule, where Γ is the result of checking φ in Y .

And let it be evaluable(Γ) $\begin{cases} 1 & \text{if } \Gamma \text{ is true.} \\ 0 & \text{if } \Gamma \text{ is false.} \end{cases}$ Then, the semantic rule preconditionRule is

defined as $\forall \varphi, Y, \Gamma (\Gamma, \text{preconditionRule}(\varphi, Y, \Gamma) \rightarrow \text{True}(\Gamma))$.

Rule 5.10: *A precondition in an assertion is an expression that evaluates the facts of the assertion. If the fact fulfils the precondition, the fact it is evaluated. Otherwise, it is not. In the latter case, the assertion is evaluated, but without the facts that do not fulfil the precondition.*

Let δ be an assertion. Let Y be a regular expression.

Let φ be a fact $\in \delta$. Let Z be a precondition, where $Z \in Y$, and $Z \in \delta$.

If φ fulfils Y , then φ is evaluated. Otherwise, φ is ignored.

This rule can be explained with propositional logic, namely:

Let $\text{preconditionRule}(\varphi, Y, \Gamma)$ be a rule, where Γ is the result of checking φ in Y .
And let it be $\text{evaluable}(\Gamma) \begin{cases} 1 & \text{if } \Gamma \text{ is true.} \\ 0 & \text{if } \Gamma \text{ is false.} \end{cases}$ Then, the semantic rule preconditionRule is defined as $\forall \varphi, Y, \Gamma (\Gamma, \text{preconditionRule}(\varphi, Y, \Gamma) \rightarrow \text{True}(\Gamma))$.

Definition 5.11: A filter group defines a set of explicit filters to which all assertions are circumscribed.

This set of filters is a set of *dimension-dimension attribute* pairs, where each pair is matched independently with all variables. For example, for an explicit filter group $(D1, A1)$, $(D1, A3)$, where $D1$ is a dimension and $A1, A3$ are dimension attributes, supposing a variable with an explicit filter, in this case a basic concept (*Primary Items*, $C1$), then the facts to be evaluated are:

- $(D1, A1)$ and $C1 \rightarrow \text{Fact1}$.
- $(D1, A3)$ and $C1 \rightarrow \text{Fact2}$.

While $((D1, A1), (D1, A3), C1) \rightarrow \text{Fact3}$ is not evaluated.

From here, it is possible to deduce the following rule:

Rule 5.11: A set of dimension-dimension attribute pairs in an assertion can be defined and is named filter group.

Let δ be an assertion. Let $\langle DE_k At_l \rangle_{k=0}^n$ be a dimensions group, $\forall l = 1, \dots, m$, where DE are the explicit dimensions $\in \delta$. Then, $\langle DE_k At_l \rangle_{k=1}^n$ is a filter group (FG).

Let $\varphi_{q=1}^n$ be facts $\in v_i \in \delta$. Then, $\left((DE_k At_l) \cup (DI_h At_o) \cup (DV_v At_p) \right)_{k=1}^n \rightarrow \varphi_q$,

$\forall k = 0, \dots, n, \forall l = 0, \dots, m, \forall h = 0, \dots, n, \forall o = 0, \dots, n, \forall v = 0, \dots, n, \forall p = 0, \dots, n$.

Table 5.1 shows the mapping from XBRLDM formulas to the MDM, i.e. the mapping from the CIM to the PIM. Where the column 1 shows the number of definition, the column 2 depicted the definition in the XBRLDM and the column 3 the mapping to the MDM or the UML star model.

Table 5.1 From the CIM to the PIM

Def	XBRLDM, the CIM	MDM or star model (the PIM)
1	Fact	Fact
2	Assertion	Constraint
3	Filter	Set of pairs <dimension/attribute of dimension>
4	Implicit filter	Set of pairs <dimension/attribute of dimension>
5	Explicit filter	Set of pairs <dimension/attribute of dimension>
6	Basic concept	Pair <Dimension/attribute of dimension>
7	Constant	Constraint
8	Variable	Set of pairs <dimension/attribute of dimension>
9	FallbackValue	Constraint
10	Precondition	Constraint
11	Filter group	Set of pairs <dimension/attribute of dimension>

The preceding definitions and rules can be explained with the following example using the Bank of Spain's consolidated public sector balance sheet 6610 (2008) (*Bank of Spain, 2011*). The example is oriented to the clarification and simplification of the specification, with a focus on the MDM. The source of the example is XML. Through the Arelle open-source XBRL platform (*Arelle, 2015*), the assertion of the example is obtained in an easier way. As the aim of this thesis is to clarify the function of the XBRL formulas, the following is the analysis of not all, but rather just one assertion from the report complete (*Figure 5.6*). The assertion is displayed graphically in *Figure 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>

Figure 5.6 Example of the assertion *val_6610sh-2*

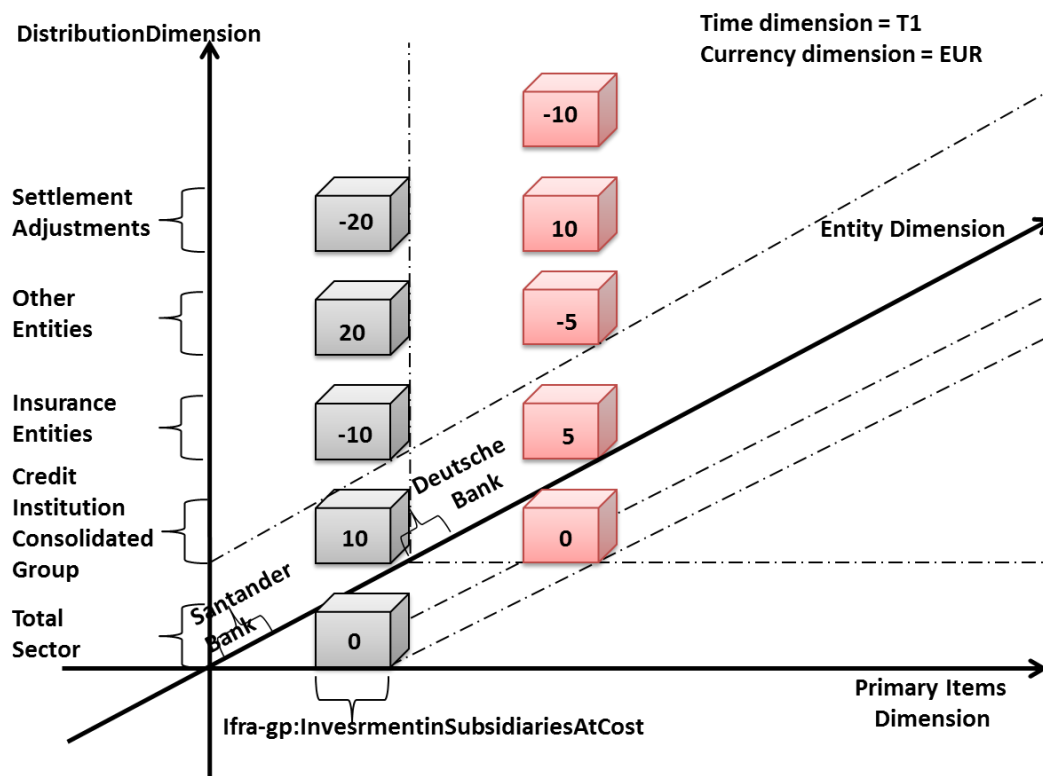


Figure 5.7 Example of the assertion *val_6610sh-2* presented graphically

A set of assertions begins with the tag *assertionSet*. Normally, this group is related to a XBRL instance document, in this case to the *6610 report*, however, the association is only logical, as there are no ways to identify the set with an XBRL instance document.

The assertion starts with the assertion name, here *val_6610-sh-2*, and then the expression to be validated (see Figure 5.6):

$((($creditInstitutions + $InsuranceEntities + $otherEntities + $settlementAdjustments) eq $TotalGroupEntities) and ($TotalGroupEntities eq 0))$

The above expression consists of five variables, namely: *\$creditInstitutions*, *\$InsuranceEntities*, *\$otherEntities*, *\$settlementAdjustments* and *\$TotalGroupEntities*. This expression means that, on the one hand, the sum of *\$creditInstitutions*, *\$InsuranceEntities*, *\$otherEntities* and *\$settlementAdjustments* must be equal to *\$TotalGroupEntities* and that, on the other hand, *\$TotalGroupEntities* must be equal to zero.

Following the variables appears *conceptName ifrs-gp:InvestmentsInSubsidiariesAtCost*, an explicit filter. With the tag *conceptName*, the dimension can be identified as basic concepts, the dimension attribute as *ifrs-gp:InvestmentsInSubsidiariesAtCost* and the *dimension-dimension attribute* pair as (*basic concepts, ifrs-gp:InvestmentsInSubsidiariesAtCost*). In the basic concept, *ifrs-gp* denotes the IFRS General Purpose taxonomy while *InvestmentsInSubsidiariesAtCost* denotes the basic concept.

Variables appear after the definition of the assertion:

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

In the above, *factVariable* is the tag, while the variable, whose name begins with \$, is *\$creditInstitutions*. The attribute *fallbackValue* is also defined and, as *fallbackValue = 0*, all facts determined by the variable as having a null value (i.e., the fact does not exist) are given the *fallbackValue* of zero here. Following this, the explicit filter or *dimension-dimension attribute pair* appears which, in this case, is (*sp-bs-d-FR-dist:DistributionDimension, CreditInstitutionConsolidatedGroup*). In the dimension, *sp* denotes Spain, *bs* the Bank of Spain, *d* dimension, *FR* the *FINREP* general taxonomy and *dist* distribution. Thus, *sp-bs-d-FR-dist* is a logical name of the concept referring to the taxonomy. In this report, there are three implicit filters, namely, *Period of time*, *Entity* and *Currency*. These implicit filters in the formula are not shown here.

5.5. Semantic questions of the XBRL formula specification

One particular problem arises when the instance document uses implicit filters and a fact is null. For example, let *\$A*, *\$B* and *\$C* be variables for a validation rule $A=B+C$ and let each variable have a different explicit filter whose basic concept is *C1*, *C2* and *C3*, respectively. Regarding implicit filters (dimensions) *Entity* and *TimePeriod*, it can be supposed that the former has dimension attributes *E1* and *E2*, while the latter has only one dimension attribute *T1*. Let the example also have the facts *FA1*, *FA2*, *FB1*, *FB2*, *FC1* and *FC2* where:

- 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 $C3 \cup E2 \cup T1 \rightarrow FC2$

Next, it is analysed when, for example, the variable FA does not have the attribute *fallbackValue* and the fact $FA2$ is also null (Figure 5.8).

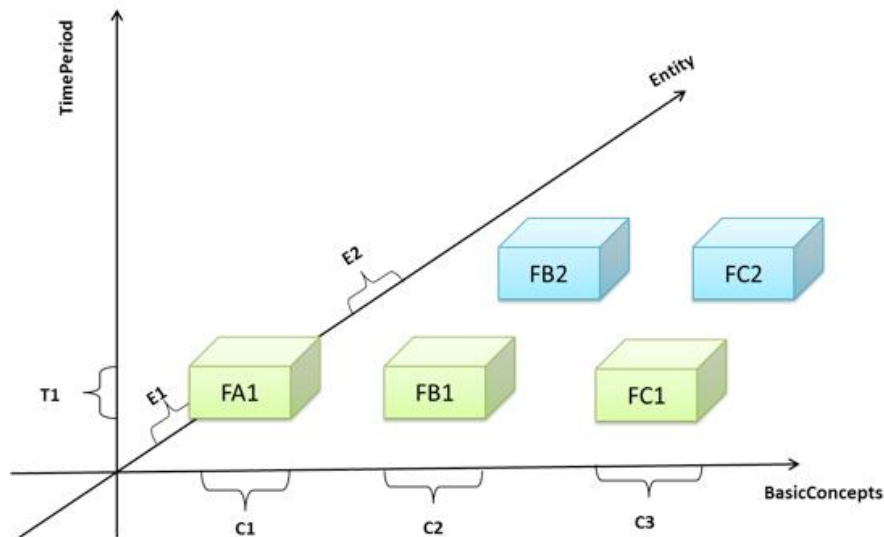


Figure 5.8 Example with null fact

The XBRL processor or DBMS must validate:

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

Since $FA2$ is null according to *Rule 5.8* as shown above, the assertion $FA2=FB2+FC2$ is not validated.

Another potentially problematic case is when the instance document uses implicit filters and a variable has two facts, since it contains an additional dimension (i.e.,

explicit filter) compared to the other variables. For example, let $\$A$, $\$B$ and $\$C$ be variables for a validation rule $\$A=\$B+\$C$ and let each variable have a different explicit filter whose basic concept is $C1$, $C2$ and $C3$, respectively. Regarding the implicit filters (dimensions) *Entity* and *TimePeriod*, it can be supposed that the former possesses the dimension attributes $E1$ and $E2$, while the later possesses only one, $T1$. Let the example be for the facts $FA1$, $FA2$, $FB1$, $FB2$, $FC1$ and $FC2$ where:

- 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 $C3 \cup E2 \cup T1 \rightarrow FC2$

The XBRL processor or DBMS must validate:

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

If another dimension is involved in a fact - for example, the dimension $D1$ with attributes $A1, A2, A3, \dots$ - then, if $C1 \cup E2 \cup T1 \cup (D1, A1) \rightarrow FA2$ and $C1 \cup E2 \cup T1 \cup (D1, A2) \rightarrow FA2'$. However, the new dimension is not defined in the assertion. In such a case, it is possible that $C1 \cup E2 \cup T1 \rightarrow FA2$. Thus, it is possible to generate two contexts with the same basic concept and two different implicit filters for each. An example is shown in *Figure 5.9*.

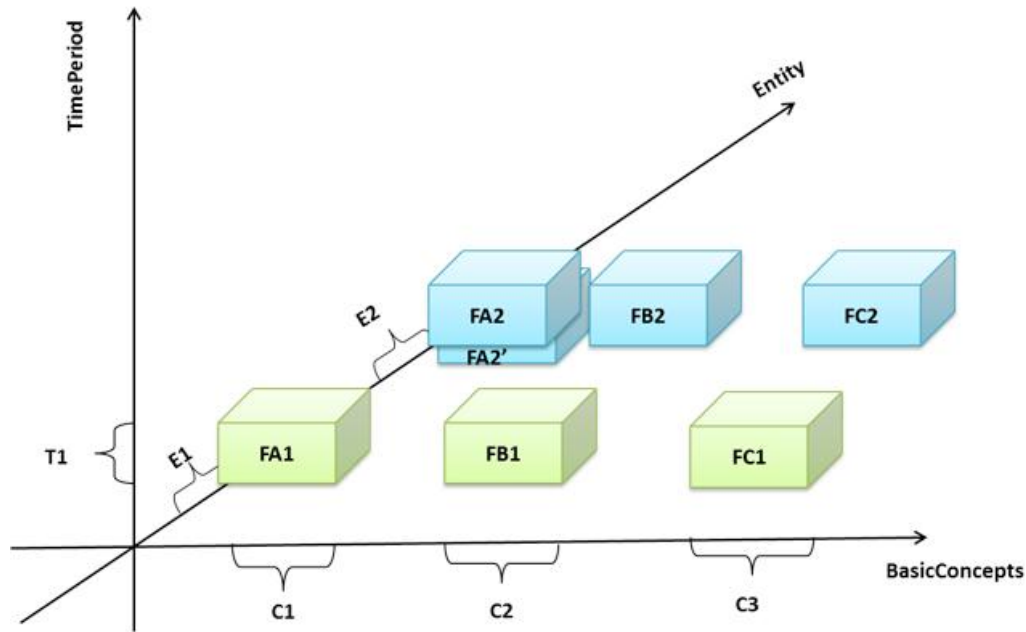


Figure 5.9 Multivalued facts

If $C1 \cup E2 \cup T1$ is $FA2$ and $FA2'$, then it is necessary to decide amongst:

- 1 $FA2 = FB2 + FC2 \vee FA2' = FB2 + FC2$
- 2 $FA2 = FB2 + FC2 \wedge FA2' = FB2 + FC2$
- 3 None

While the standard does not specify anything about the possible choice, some has to choose. The problem is nevertheless rare, since taxonomy, formula or instance designers tend to avoid it, although not in an automatic way through, for example, a CASE (*Computer-Aided Software Engineering*) tool.

5.6. Implementation and proof-of-concept (POC), in the PSM

This section presents the implementation of the proposal of the thesis, in which the mapping, validation and automation of XBRL instance documents through a RDBMS constitute a novel contribution to the field. In order to validate the proposal, a POC is presented in specialised conferences (*Santos and Castro, 2011b, 2012a*).

An earlier presentation by the authors demonstrated the mapping from an XBRLDM to RDBMS and vice versa. This mapping and automation proved useful for two principal reasons; namely, **(1)** XBRL experts were able to validate earlier studies (*Santos and Castro, 2010, 2011a and 2011c*) and **(2)** the mapping and automation could be used as an open-source tool by companies and regulators with more limited resources, a pertinent fact in light of the current financial crisis.

This section presents the architecture for XBRL instance document validation in a DBMS and a process for the automated validation in a database of instance documents in XBRL formula specification. While an earlier presentation by the authors – which has generated some interest among regulators in Spain – offered a POC for the validation of XBRL instance documents in a DBMS, only the code, graphics and demos were shown (*Santos and Castro, 2012a*). The interest of regulators in the POC, even in the more limited version presented, was due in part to the fact that the repository of an XBRL instance document generated is always a database, something they recognized as an important development given the lacklustre performance of many XBRL instance documents with large volumes of data.

The architecture for mapping XBRL formula specification to an RDBMS is divided into two stages. In the first, the code necessary for creating a validation API is generated. This API is executed in the second stage to validate the facts of the report or XBRL instance document in the database.

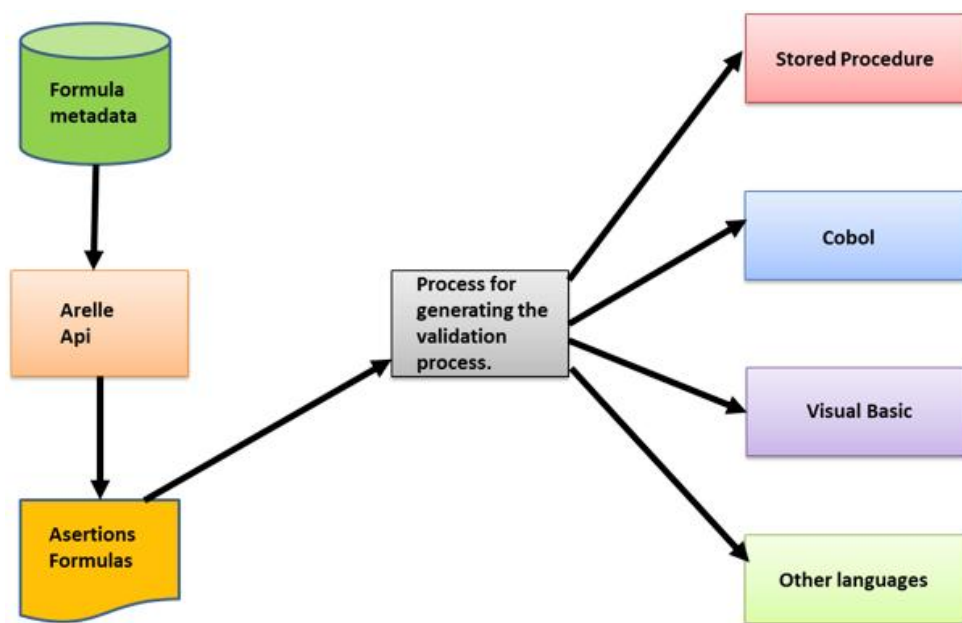


Figure 5.10 Obtaining the validation API

Returning to the earlier representation in *Figure 5.1*, the first stage begins at Step 3 when the metadata structure, such as dimensions and dimension attributes, has been created. Nevertheless, it is important to mention that in the validation API generation process, it is not necessary to have previously loaded the metadata and facts into the database. Rather, the API can be generated at any time prior to validating an instance document in the database and after having created the XBRL metadata structure. *Figure 5.10* displays this first stage. First, the formulas and assertions are obtained through *Arelle* in a simpler format. The *Arelle* API converts the XBRL formulas, originally in XML format, into plain text. Next, a new API proposed by the authors is generates another API to be used to create the validation process. This validation process can be a stored procedure in the database or can be generated in other languages like COBOL, Visual Basic (VB) and C++ that work with the database. This study takes the former approach. Once the validation API has been created and the metadata and facts loaded into the database, the validation API may be executed (see *Figure 5.11*). This validation process can be repeated several times.

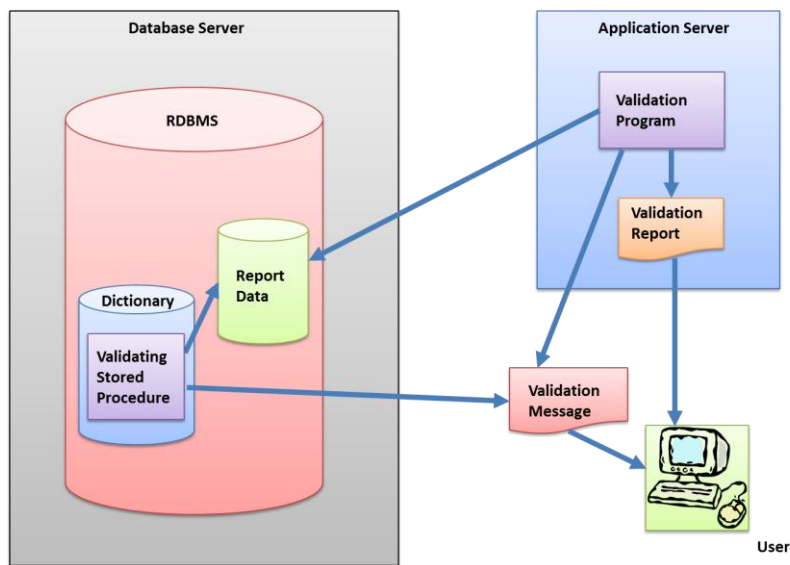


Figure 5.11 Validation process

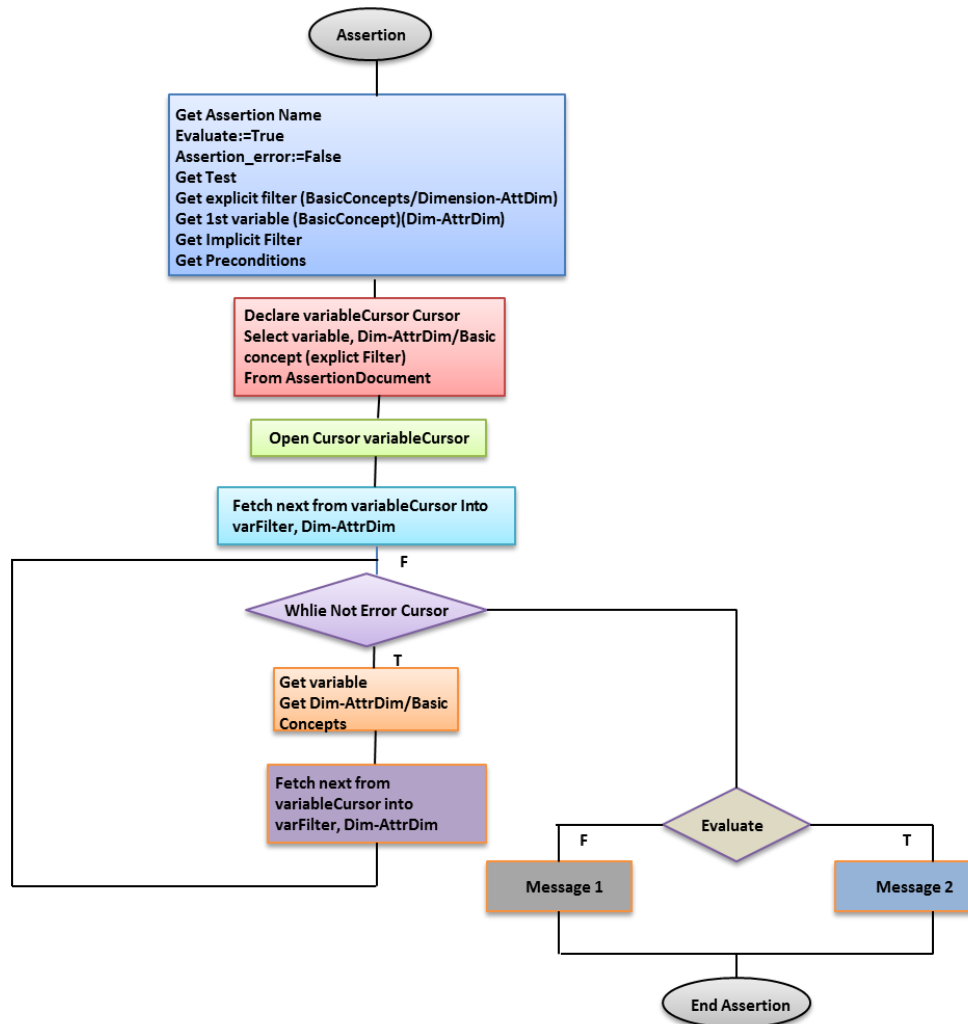


Figure 5.12 Assertion process of validation-process-generating algorithm

In what follows, the validation API generation process of the proposal is briefly explained. Firstly, the XML formulas are read until reaching end of file (EOF). Each assertion is then processed by the assertion algorithm. The algorithm (CIM→PIM) notes the assertion name, the test to be validated and the preconditions if the XBRL has (see *Figure 5.12*). The algorithm then obtains the implicit dimensions which are then processed. The algorithm reads each variable and goes to process these variables. Finally, the algorithm must verify the validation if the assertion can be evaluated or not and a message is displayed (generating several templates with different queries to DBMS). In the *Figure 5.12* the evaluation implies the algorithm of creation of the SQL templates (PIM→PSM), and depending on if there are preconditions, variables, etc., in total there are 7 templates, as the *Algorithm 5.1* shows.

```

Start
  begin
    if there are preconditions
      then if there are variables
        then if explicit filter has basic concepts and not dimensions
          then template SQL V
        else if explicit filter does not have basic concepts and has dimensions
          then template SQL VI
        else template SQL VII
      else template SQL IV
    else if explicit filter has basic concepts and not dimensions
      then template SQL I
    else if explicit filter does not have basic concepts
      then template SQL II
    else template SQL III
  end
End

```

Algorithm 5.1 Creation of SQL templates

These templates (obtained in *algorithm 5.1*) are a set of stored procedures (if the validation is implemented in SQL, if not, they could be any other language such as Cobol, C++, etc.) that depend on whether there are preconditions, variables, etc. In total there are 7 templates. According to the MDA paradigm this is the PIM→PSM

step. The code can be analysed in some POCs of the thesis author (Santos and Castro, 2012a; <http://www.openfiling.info/academy/>).

Next, the code (Code 5.1) of template SQL I is shown, from *Algorithm 5.1* in the MDA paradigm. Template SQL I is used to validate an assertion without preconditions. Code 5.1 is a summary. This code belongs to the example of *Figure 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;
```

Code 5.1 Template I (SQL I), a summary of the code

When validating a fact or set of facts with an assertion, is important to know the number of facts involved in the assertion, because its execution can become unworkable (*Agrawal et al., 1997; Feng et al., 2010*). Only at run-time (during the validation of the report) will the number of facts involved in the validation be known. This number can be very high and will affect system performance. However, the proposal developed is implemented through a management system database, so this process is made affordable in general.

CHAPTER 6

Validation of the proposal

CHAPTER 6. Validation of the proposal

6.1. Introduction to the validation

This chapter shows the validation of this research work. The validation is divided in two parts. **Firstly**, the development of the metadata life cycle of the metadata model of the semantic economic financial accounting reports is validated. **Secondly**, the calculations through formals of these reports are validated. The theoretical development can be found in the chapters 4th and 5th respectively.

The principal objective of this validation is to ensure the ability of the proposed approach to be used and to accomplish the design objectives. The validation of this approach is performed in two phases. The first phase is done to ensure that the transformation of the CIM to the PIM is correctly performed. In this phase the result of this transformation is validated, i.e., the resulting PIM (or UML star model) is correct. The second phase is done to ensure that the transformation of the PIM to the code (or relational model) is correctly performed. Although it might seem redundant to use more than one type of validation, this has been done in order to increase confidence in the validation results. According to *Gogolla et al. (2007)*, the validation of conceptual models at early phases of their development can help correct faults in the design at a point where they may still be corrected with relative ease.

6.2. Validation of the development of the metadata of the semantic reports

This section analyses the validation of the proposal. The process is divided into **two phases**. The **first phase** is to obtain the UML star model, from the set of templates developed for expert users and the **second** is to obtain the relational model using ROLAP technology.

The validation involves testing that the data obtained in the development of this research work match up with expert users' requirements. *FINREP* is a taxonomy sponsored by the EBA and is consolidated and sub-consolidated financial reporting for supervisory purposes based on IAS/IFRS taxonomies (*Eurofiling 2012*). *Solvency*

// codifies and harmonizes the regulation of insurance in the *European Union* (EU) and is sponsored by the EIOPA. It is possible to say that expert users wish to obtain a set of reports of the type of *Figure 4.2*. However, the number of concepts to gather is so large (there were only 4500 in FINREP in 2012, and 45000 in 2015 (*Weller, 2015*)) that it makes it impossible to work around a report as *Figure 4.2*. Therefore expert users build templates in spreadsheets, in which each cell depicts a fact to collect of financial institutions. IT manually analyses each of these cells along with experts users, then domains, concepts, hierarchies between concepts, dimensions, dimension attributes, etc. are obtained. In the initial development (in the CIM) these templates have a large number of unconnected cells and a high level of redundancies. In the first phase, according to *Algorithm 6.1*, the diverse elements of the original templates (*Figure 4.3*) are entered into the relational model of *Figure 4.6*. By applying this algorithm repeatedly the unconnected and redundant cells are analysed. This figure displays the set of artefacts in UML, which will be the initial structure in the automatic validation of the model, in the PIM. *Algorithm 6.1* uses the definitions and rules from *sections 4.3* and *4.4* and verifies the semantic questions of the *section 4.5*. It is this process is really that really makes the structural validation (*Santos and Nieto, 2014, 2015*), verifying if hierarchies of concepts are valid in a domain, with regard to dimensions, dimension attributes and calculated attributes (if dimension attributes belong to one domain rather than two at a time, etc.).

```

start
  read data type, domains, concepts, basic concepts;
  read dimensions, dimension groups;
  verify the hierarchies of the concepts and dimensions;
  obtain dimension attributes, calculated attributes;
  obtain allowed cubes, forbidden cubes;
  obtain UML star model
  create dimension tables from dimensions and dimension attributes in the star model;
  create stored procedure with calculated attributes;
  create base dimension;
  create facts from allowed cubes;
end

```

Algorithm 6.1 Extraction of the metadata model

Next, the UML star model is obtained, as in *Figure 4.4*. To achieve these transformations in the POC, this thesis uses *SQL Server Integration Services* (SSIS)

an ETL (*Extract, Transform and Load data*) product of Microsoft. The second phase obtains the relational model, validating in a semiautomatic way that the implementation is correct, i.e. that the number of dimensions, dimension attributes, calculated attributes, etc. is correct. For this phase, *Algorithm 6.1* uses the transformation rules from section 4.5. Finally, the code is obtained for the ROLAP technology.

After what has been previously mentioned, it is necessary to analyse a little more detail with validation examples, although these are analysed in detail in *Openfiling* (2015), including its implementation. The first phase consists of the verification that the transformation of the set of templates into the star model. For this, firstly, the original templates in an iterative way are validated for eliminate the redundancy. Immediately, these templates are tested, discovering some errors in the initial templates, for example that the connection between concepts and taxonomies were not distinguished, and furthermore, some hierarchies of concepts were not correct (*Santos and Nieto, 2014*). Subsequently, and for a complete set of tests, these templates are updated with insertions or modifications of incorrect elements, as *Table 6.1* shows, after which, it is verified whether the output is as expected. In the same way, the transformation to the relational model is tested. For this reason, the set of data with insertions or modifications of incorrect elements is updated, as *Table 6.2* shows, and afterwards it is verified whether the output is as expected. The process of *Santos and Nieto (2014, 2015)* is based on more than 20 modules, in that each concept is analysed, for example, whether the hierarchy of the concepts in a domain is correct. In this example the elements are loaded in a table with two leaves; upon each insertion a trigger is executed and this program verifies that there are not cycles in the hierarchy. Each of these modules load each corresponding concept in UML structure of *Figure 4.6*, directly verifying the constraints of the model (repeated elements, references to other elements, etc.). These modules are actually *Integration Services* packages, and once the validation is done, they write in a log the validations performed, right and wrong. Finally, each module performs an automatic creation and insertion of various elements in the star structure in the PIM (*Figure 4.4*).

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

Figure 6.1 FINREP 2012, statistical data in the MDM

As indicated in this section, for this set of tests three samples are chosen, *FINREP 2012*, *FINREP 2014* and *Solvency II*. Eurofiling coordinated the development of *FINREP 2012* (Eurofiling, 2012). However, *FINREP 2014* and *Solvency II* were coordinated through the EBA and the EIOPA with Eurofiling. Next the *FINREP 2012* taxonomy is analysed, which is simpler than the other two selected in the test set. This taxonomy is defined in a spreadsheet. This spreadsheet consists of 11 sheets and 13 templates. *Figure 6.1* displays a summary of the concepts published in Eurofiling (2012). In this release there are 187 concepts, 17 domains, 16 basic concepts, 16 groups of dimension (such as *Categories*, CT; *Comprehensive income*, CI; *Amount types*, AT; *Portfolios*, PL; etc.), 157 dimensions (such as *Category of assets*, AS; *Category of assets and liabilities*, AL; etc.), dimension attributes and calculated attributes. The dimension group column EC (*Entity codes*) has no concepts, because it is an implicit dimension, i. e. without dimension attributes in the definition of the taxonomy. In this sample, *FINREP 2012*, the allowed or forbidden hypercubes are not specified. In addition, it is necessary to create 82 rules for the calculated attributes. The next version of *FINREP 2014* is similar to *FINREP 2012* but with the new requirement of the European Parliament (EBA, 2013), more information about the structure can be found in Santos et al. (2013) and the EBA (2014). In relation to the *Solvency II* taxonomy, the first prototype is used in this test case: more information about this structure can be found in Santos et al. (2013) and the current taxonomy in EIOPA (2014). The POCs for the aforementioned proposal and the assumption that expert user can validate the design and its model are shown in Openfiling (2015) and Santos and Castro (2011b, 2012a) and Santos and Nieto (2014, 2015). *Figure 6.2*

shows templates developed for the POC (disclosed in *Eurofiling* forum). In the validation process, the elements of templates are selected and validated in groups, showing a log of errors and validation incidents.

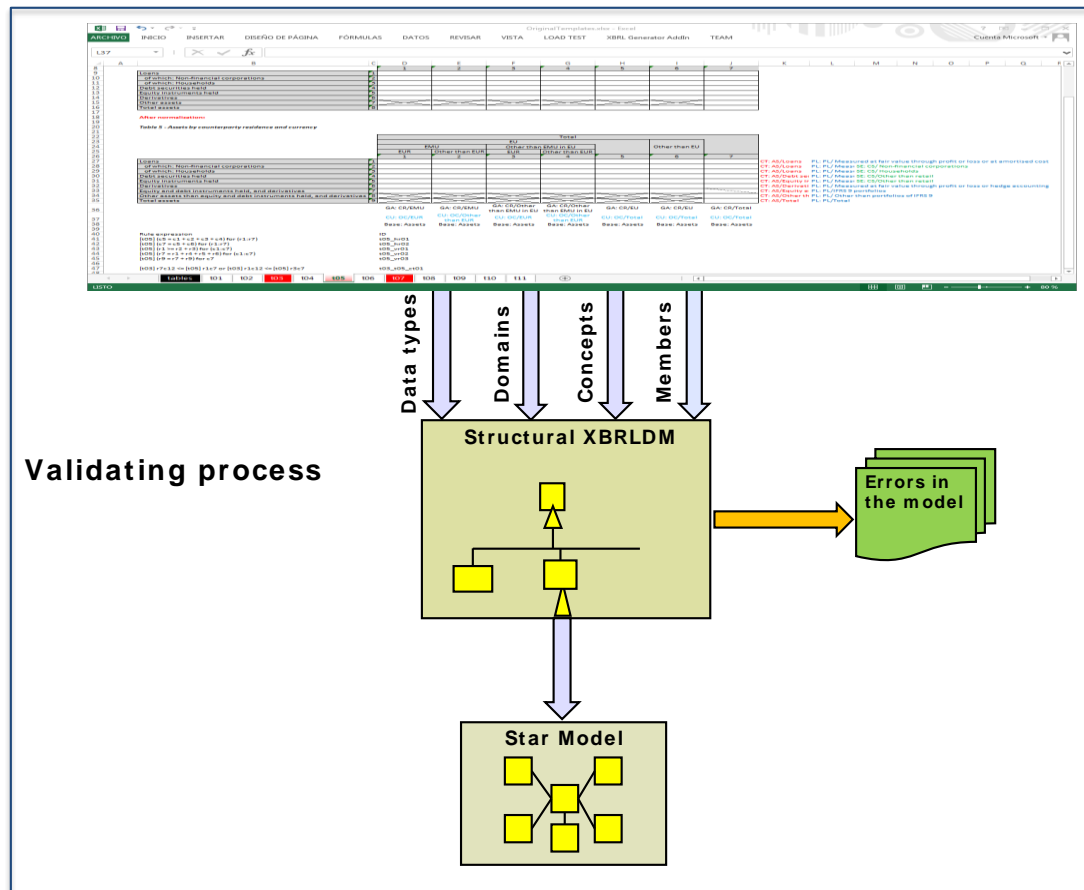


Figure 6.2 Validation of the metadata

In *Figure 6.3* one of the *Integration Services* packages that validates the dimensions is shown. Moreover, this package executes the process of creating the fact table.

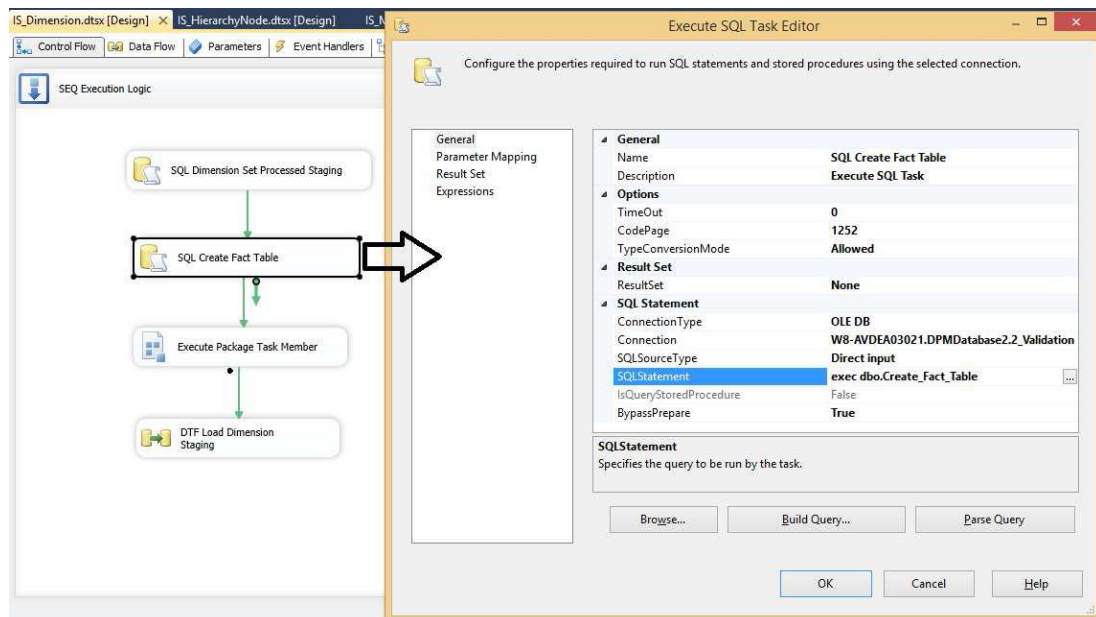


Figure 6.3 Flow of IS_DIMENSION data. Execution of the fact table creation SQL procedure (Nieto, 2015).

The code 6.1 shows part of the creation of the structure of the tables of the star model (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

```

Code 6.1. Creation of the tables of the star model (Nieto, 2015)

According to the way that elements, domains, dimensions and dimension attributes were validated, two triggers were created. A trigger ensures that no sheet is inserted into the lower level which have a different sheet at the upper level (Code 6.2). If this were the case, it would a cycle, and would not be a tree structure.

The second trigger is used to create tables of the dimensions and insert dimension attributes into the dimensions (Code 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

```

Code 6.2. Cycles in the hierarchy of concepts are searched for (Nieto, 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

```

Code 6.3. This trigger creates the tables of dimensions and loads the dimension attributes into the dimensions (Nieto, 2015)

Figure 6.4 shows the MDM diagram of *FINREP 2014* obtained at the end.

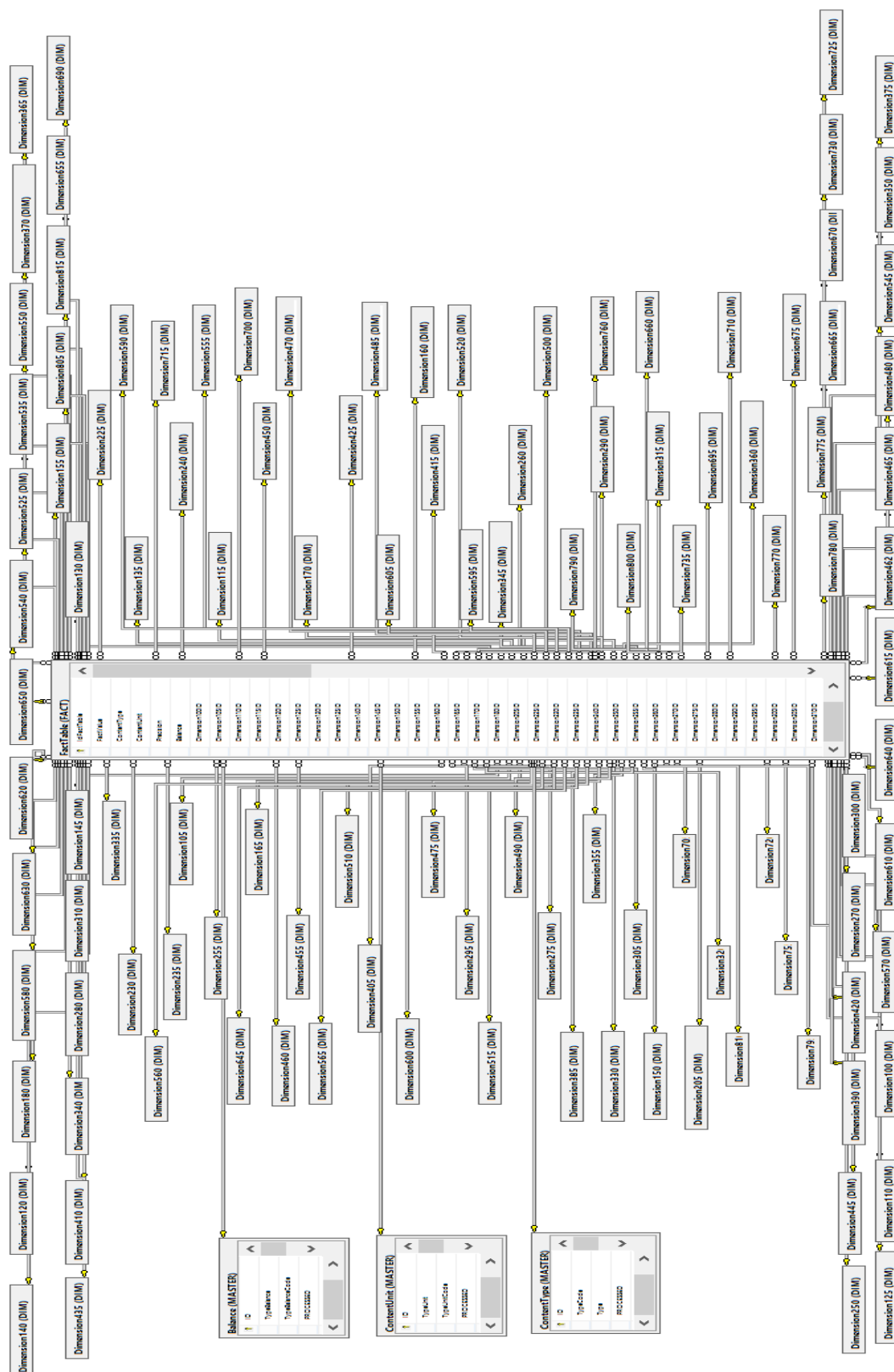


Figure 6.4. Diagram of the database in the MDM of FINREP 2014 (Nieto, 2015)

Tables 6.1 and 6.2 verify a set of validation tests for the proposal. Column 1 shows the test number. Column 2 shows if the validation belongs to the star model in UML or to the relational model (RM). Column 3 shows the test to validate. This column shows the test case, for example, test number 1, *two repeated concepts* means that it is impossible to repeat two concepts. Columns 4, 5 and 6 are inputs to the test. These columns display the set of correct objects and the set of incorrect objects to test. For example, test number 1 shows *187 concepts + 3 repeated concepts*. Finally, the last column gives the test output. In test number 1, in the three samples (*FINREP 2012, 2014* and *Solvency II*) three repeated concepts are inserted, respectively, so the test output is three errors with three repeated concepts, respectively.

Table 6.1 Validation tests belonging to the UML star model

Number of test.	Phase	Test to validate	Input FINREP 2012	Input FINREP 2014	Input Solvency II	Test output
1	UML	Two repeated concepts	187 concepts+ 3 repeated concepts	1632 concepts + 3 repeated concepts	145 concepts + 3 repeated concepts	3 repeated concepts
2	UML	Two repeated domains	18 domains + 2 repeated domains	35 domains + 2 repeated domains	1 domain + 2 repeated domains	2 repeated domains
3	UML	A concept in two domains	187 concepts and 2 concepts in two domains	1632 concepts and 2 concepts in two domains	145 concepts and 2 concepts in two domains	2 concepts in two domains
4	UML	Two repeated dimensions	26 dimensions + 1 repeated dimension	92 dimensions + 1 repeated dimension	2 dimensions + 1 repeated dimension	1 repeated dimension
5	UML	A family group repeated	16 family groups + 1 repeated	35 family groups + 1 repeated	1 family group + 1 repeated	1 dimension group repeated
6	UML	A family group is in two domains	16 family groups and 1 group in two domains	35 family groups and 1 group in two domains	1 family group and 1 group in two domains	1 group in two domains
7	UML	Cycles in the concepts of a dimension	2 dimensions, a dimension with a cycle	92 dimensions, a dimension with a cycle	2 dimensions, a dimension with a cycle	1 dimension with a cycle
8	UML	Creation of dimension attributes	26 dimensions, 157 dimension attributes + 1 incorrect dimension attribute	4 dimensions, 30 dimensions attributes + 1 incorrect dimension attribute	2 dimensions, 2 dimension attributes + 1 incorrect dimension attribute	1 incorrect dimension attribute
9	UML	Creation of calculated attributes	26 dimensions, 82 calculated attributes + 1 incorrect dimension attribute	4 dimensions, 18 calculated attributes + 1 incorrect dimension attribute	2 dimensions, 2 calculated attributes + 1 incorrect dimension attribute	1 incorrect dimension attribute
10	UML	The concepts of a dimension has one only root	26 dimensions, 171 concepts. Two roots in a dimension	92 dimensions, 1632 concepts. Two roots in a dimension	2 dimensions, 4 concepts. Two roots in a dimension	Two roots in a dimension

Number of test.	Phase	Test to validate	Input FINREP 2012	Input FINREP 2014	Input Solvency II	Test output
11	UML	Time period type the same in the dimension and in the basic concept in a hypercube	10 hypercubes and one with a different time period type	(No basic concepts, it is an additional dimension)	2 hypercubes and one with a different time period type	1 different time period type
12	UML	Each implicit dimension is associated with one data type	2 implicit dimensions, one without type	5 implicit dimensions, one without type	1 implicit dimension, one without type	1 without type

Table 6.2 Validation tests belonging to the relational model (RM)

Number of test.	Phase	Test to validate	Input FINREP 2012	Input FINREP 2014	Input Solvency II	Test output
1	RM	Creation of dimension attributes	26 dimensions, 157 dimension attributes + 1 incorrect dimension attribute	4 dimensions, 30 dimension attributes + 1 incorrect dimension attribute	2 dimensions, 2 dimension attributes + 1 incorrect dimension attribute	1 incorrect dimension attribute
2	RM	Creation of calculated attributes	26 dimensions, 82 calculated attributes + 1 incorrect dimension attribute	4 dimensions, 18 calculated attributes + 1 incorrect dimension attribute	2 dimensions, 2 calculated attributes + 1 incorrect dimension attribute	1 incorrect calculated attribute
3	RM	Time period type the same in the dimension and in the basic concept in a hypercube	10 hypercubes and one with a different time period type	(No basic concepts, it is an additional dimension)	2 hypercubes and one with a different time period type	1 different time period type
4	RM	Each implicit dimension is associated with one data type	2 implicit dimensions, one without type	5 implicit dimensions, one without type	1 implicit dimension, one without type	1 without type

The validation in the POCs in the PIM performed on each sample (*FINREP 2012, 2014, Solvency II*) depict all structural validations in a 95%. The relations between concepts and taxonomies are left out due to lack of real data. Moreover, in *FINREP 2014* and *Solvency II* the base dimension is not applicable. Regarding the validation in the PSM, the different POCs show that 75% are validated, because the generation of stored procedures of the calculated dimension attributes in a domain is not tested. This is because the guide to good practice of *Eurofiling* recommended to use the *formula specification* and not the *calculation specification*. On the other hand, the published data for the EBA and the EIOPA in *FINREP 2014* and *Solvency II* also model templates and axes of cubes, however this type of modelling are redundant and is used in the presentation for expert users.

This proposal produces well-built metadata for semantic economic/financial reports because it is a structural validation. However, it is necessary to continue the validation with expert users, in order to validate the semantically-complete design. To achieve this, an accounting/economic/financial study of the concepts domains, hierarchies, etc. is necessary, and that is left for future works.

6.3. Validation of calculations using XBRL formula specification

This section is based on several *Proof Of Concept* (POC). The most important, and with which is the proposal of this thesis, it was shown in *Eurofiling 15th Workshop* (Santos and Castro, 2012a; Santos et al, 2016). The platform of the POC is a database server, the RDBMS chosen is the *Microsoft SQL Server 2012*, and the programming language is *VBScript* and *T-SQL*. As regards the operating system of the database server, *Windows Server 2008 R2* is used. The client is *Windows 7*.

In the POC, the 6610 report was checked, along with the set of tests – consisting of 12 numerically identified assertions –. Firstly, the structure necessary is created for this proof of the concept, i.e., the set of tables. However, in this test instead of using the star model of *Figure 4.4*, it puts a table of contexts between dimensions and the fact table, that is to say, all pairs <dimension - dimension attribute> belonging to a fact (*Figure 6.5*).

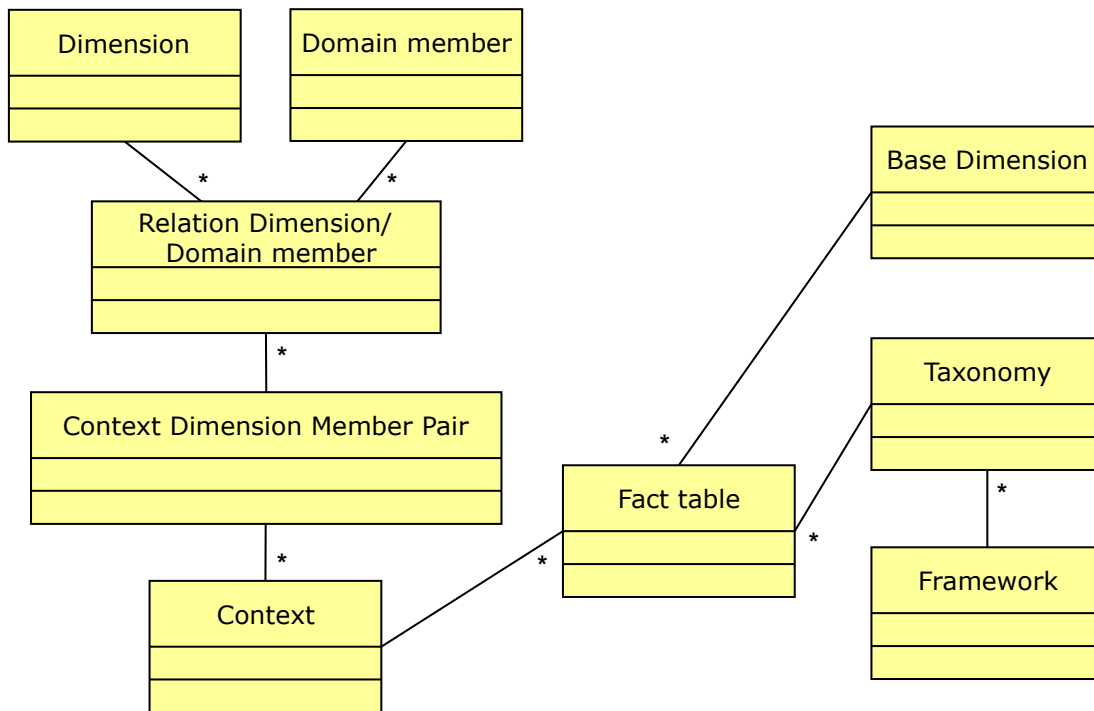


Figure 6.5. Diagram with the context in the star model.

As shown in *section 5.1*, from the *API* of *Arelle* (2015), an open source tool, the dimensions, dimension attributes, basic concepts, etc. are obtained in a flat file, but with XBRLDM format. From this flat file and with all necessary elements, the metadata in the star structure is loaded (*Code 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

```

Code 6.4 SQL script that creates the tables in the star model

Regardless of loading the fact table, i.e., of the reports, in a parallel the stored procedure creation process with the validations of the formulas is executed. The process of validation of the formulas starts with the formulas in the XBRLDM that are loaded in a flat file (Code 6.5) with the API of Arelle (Figure 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_GrupoConsolidableEntidadesCredito390e15,,,fals
e,fallbackValue =0

```



```

,,,explicitDimension,fil_DistribucionDim_GrupoConsolidableEntidadesCreditod390e16,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

```

Code 6.5. Formula val_6610-sh-2 obtained from Arelle (Figure 5.6)

From the algorithm shown in *Figure 5.12*, *Algorithm 5.1*, and *Code 5.1*, the set of stored procedures of validations is obtained, but only for one assertion (*Figure 5.10*). *Code 6.6* shows the stored procedure of validation of the assertion of *Figure 5.6*, obtained in an automatic way.

```

-- 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
else
begin
    set @entidadesSeguros=cast( @entidadesSeguros_A as int);
    --select '@entidadesSeguros....:', @entidadesSeguros
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

    -----

    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
--end

go

```

Code 6.6 Procedure of validation of the assertion *val_6610_sh_2_1*

This is summarized in the table of *Figure 6.6*. The star model is validated with a comparison among the dimensions, dimension attributes, and constraints in the MDM and the XBRLDM (*Table 5.1*). Finally, another validation is done, the data are compared between the XBRL instance documents and the facts in the database. The first column of this table, *Figure 6.6*, shows the assertion number being tested. In the second, the implementation type is identified numerically out of the five possible stored procedures (templates in the algorithm) in the POC. In the following column, the number – there are always two – and types of implicit filters are identified. In the POC, both implicit filters are the basic concepts (BC) *Entity* and *Period of Time*, for this the number is two. The fourth column is for the explicit filters in the whole assertion. As in the previous column, the fourth consists of two sub-columns indicating filter type – here BC or dimension (D) – and the number of dimensions and/or BC. The fifth

column indicates variables and is subdivided into four additional columns corresponding to (1) number of variables, (2) type of variable (here BC or D), (3) number of filters (here BC or D) by number of variables and (4) number of not null *fallbackValues*. The final three columns of the table correspond, respectively, to presenting yes or no responses to constants, the test to evaluate and the precondition.

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

Figure 6.6 Validation test for 6610 report

For example, row one means that it is the first sample of the 6610 selected reports (it is used as an example at the end of section 5.4, *Figures 5.6 and 5.7*). The second cell indicates that the template used is the first one (which is a stored procedure in SQL). The third, fourth, fifth and sixth cells are the filter used (two and one dimensions of basic concepts for each kind of filter, respectively). The seventh, eighth, ninth and tenth cells mean five variables which are dimensions and are not basic concepts, each variable includes a dimension, and all these variables have to be initialized, if the value is null, except the last (*Figure 5.6*). The three final cells indicate that a constant value does not exist, the validation must be made through a test and that no

preconditions are needed, respectively. Thus can be explained the other rows of the table in *Figure 6.6*.

This POC was chosen for containing the largest number of different examples in FINREP 2008. However, not all assertions are validated in an automated way: only a sample of approximately 10% was chosen because the rest of the formulas had the same type. In the POCs of León (2012) and Cid (2013), which were carried out later, the pure star model was used, without the *contexts*, as shown *Figure 4.4*, but with the *FINREP 2014* and *Solvency II* taxonomies. In these tests, 100% of the test kit provided in *Eurofiling*, the EBA and the EIOPA is used. Nevertheless, the author of this work considers this absolutely insufficient, because the set provided by the institutions does not include the full set of formulas defined in *Chapter 5*.

It would be desirable in the near future to have a full set of complete tests for the formulas and/or assertions. In spite of this, it can be concluded that the experiments have demonstrated the validity of the proposed model.

CHAPTER 7

Conclusion and future
work

CHAPTER 7. Conclusion and future work

7.1. Conclusions

This thesis analyses all the development stages of the creation of economic/financial report metadata using the *eXtensible Business Reporting Language* (XBRL) specification, and also analyses the XBRL formula specification itself. This is proposed in this specification due to the importance of the validation of information transmitted using this standard between financial institutions. From this analysis various shortcomings have been identified, such as the impossibility of interoperability in the current situation, a lack of full automatic semantic development and creation semantics for financial reports, and the lack of availability of performance test kits. This paper proposes that these problems be overcome by a general process of development from the real world to its physical implementation in a database. To achieve this, the use of the *Model Driven Architecture* (MDA) paradigm is proposed. MDA is a methodology proposed by the *Object Management Group* (OMG) and widely used in software development projects. This research work presents the design life cycle of software development, from expert users, the *Computation Independent Model* (CIM), to the code. The *Multidimensional Data Model* (MDM) is chosen in the *Platform Independent Model* (PIM), because this model is adjusted to the development of European metadata, and as a relational model (*Relational On-Line Analytical Processing*, ROLAP) in the *Platform Specific Model* (PSM) is chosen, for its implementation in a *Relational Database Management System* (RDBMS). By means of the MDM the definitions and rules are formalised and the semantics of the *XBRL Data Model* (XBRLDM) are audited. The MDM is accompanied by a formalisation and mathematical representation of definitions and rules. Moreover, the automation of this mapping is also proposed.

The aim of this research is to clarify the XBRL and multidimensional data models, as well as the mapping from XBRL to the MDM and vice versa. Furthermore, as an additional objective is the transfer of the conceptual model to a logical and physical model, the mapping between XBRL to the MDM is accompanied by a formalization and mathematical representation of definitions and rules.

The authors hope that the fulfilment of each of these objectives might allow companies, financial institutions and regulatory authorities to better understand XBRL and its capabilities.

This research work is based on the European development of economic/financial concepts. This is because the researchers are in the European framework, and participate in the *Eurofiling* (2015) group and the *Comité Européen de Normalisation Workshop XBRL* (CEN/WS XBRL; CEN, 2013). This work entails a set of developments and *Proofs Of Concept* (POC) that implement this research, which have been presented in specialised forums for their validation (*Santos and Castro, 2010, 2011b, 2012a; Openfiling, 2014; Santos and Nieto, 2014, 2015*).

The implementation of the metadata in a database, instead of creating taxonomies using the XBRL specification, is gaining certain interest among regulators. This may be due in part, to the high cost of the XBRL specification infrastructure, taking into account the economic constraints of most financial institutions, supervisors/regulators and companies in general. In addition, when XBRL reports (XBRL document instances) are very large, they become complex or even impossible to implement by means of *eXtensible Markup Language* (XML) technology (*Santos and Castro, 2011a*). Implementation in a database makes the problem solvable, especially in relation to validation.

In short, it is possible to say that:

- The reports are designed by economists, accountants, or financiers. The main purpose of this thesis is the development of the entire metadata model's life cycle using a robust architecture technology as the MDA, non-existent to date.
- Therefore, it facilitates the extension of new elements and / or modifications that using this methodology (the MDA).
- These reports, which may become large, can be implemented directly in databases. In principle, the implementation of these reports in a database manager can avoid the limitations of processing large reports in XBRL, helping with the possibility of monitoring its performance and subsequent tuning analysis.
- The establishment of a data model design life cycle ensures fewer errors in the design, since it has been proven in concept testing and validation, and

gives the possibility of making a set of test cases for analysing anomalies and other semantic questions.

- This thesis has validated the interoperability of this technology by studying its design. Moreover, it shows how this model can be implemented in different databases of different vendors and even enables mapping to other platforms.
- By using the MDM and OLAP technologies, this research work is helping to simplify complexity, without being less robust, and although the required infrastructure is not any cheaper, it can be shared by other applications.

This research work has been endorsed by the journal *Information Systems* (JCR, Q2), in the paper “XBRL Formula Specification in the Multidimensional Data Model” (Santos, et. al. 2016).

7.2. Future works

This work has to be extended not only to Europe, but to other taxonomies, such as the American/Canadian or Japanese taxonomies. It is hoped with this proposal that XBRL specification will cover other economic fields, e.g. in the exchange of reports with the status of purchase orders, or calculation of a presale, sale and availability of products, in the future.

Another logical model used in Europe (the *European Banking Authority* (EBA), the *European Central Bank* (ECB), the *European Insurance and Occupational Pensions Authority* (EIOPA), etc.) is the *Data Point Model* (DPM), this is very close to end user applications, and is oriented exclusively to development using the XBRL specification (Romanelli, 2007; Ruíz et al., 2012; DPM, 2014). The incorporation of the DPM into this life cycle will be analysed in future studies. Furthermore, if this work analyses the metadata of the reports from the point of view of the scheme, future studies will have to analyse the design life cycle of the reports themselves, that is, from the point of view of the data.

At present, the Information Technology departments of the supervisory/regulatory bodies with very large taxonomies have an important challenge, because taxonomies and their validations are created but without public test cases. This is because

supervisors are continually creating bigger and more complex models. The approach of this paper provides a way forward for the generation of these test sets.

While an automatic validation procedure for the formulas has been implemented here, it has only been conceived of to be stored in a database, inside or outside the database. An important factor to consider is the performance of XBRL formula. While in XML, tuning is more like a black box, in an RDBMS it is well-known and has been extensively analysed. Thus, an area to be explored in future research is the analysis and comparison of XML and database performance.

Chapter 5 presents the basis of the design life cycle of software development, from the expert user, the CIM, to the code. However, at present no automated design from the Universe of the Discourse to the physical model exists. For this reason, Information Systems (IS) has problems in the validation of the implementation for the expert users. While automatic design is not currently possible, the possibility of studying the semi-automation of the process should be considered.

In the validation of the formulas, Chapter 6, section 6.3, when a pure star model is used, without the contexts, as *Figure 4.4* (León, 2012; Cid, 2013) shows, there are four test cases for *FINREP 2014* and only one for *Solvency II*. It was not until earlier this year when both the EBA and EIOPA taxonomies published public test examples. So the possibility of using more sets of tests to validate the proposal is now open. The difference between the coding algorithms designed by the author of this thesis and those generated by the students tutored, is that in the latter case the coding is easier. This thesis has left as future work:

- Comparing the complexity of both types of algorithms, based on a star model with or without context, as well as the performance of both.
- Checking the performance of formula validation through stored procedures and comparing it with the performance using XML validation.

The author of this thesis has a major challenge for future research: if the formulas had already been developed from the real world to the XBRLDM, the next target for research would be based on the development of formulas from expert users and their implementation. This is an important issue, because many organizations cannot present public test cases, and some published cases are invalid, due to the lack of a full development cycle.

Despite the complexity of the domains and financial standards, and perhaps precisely because of this, experts encourage the creation of their own projects and the dissemination of works related to the sector, to clarify and come to the real world this web of definitions, taxonomies and data. The use of natural language processing techniques could help bring this world to end users by semantically describing the content of the economic-financial reports (*Mendez-Nunez and Trivino, 2010*).

Finally, the future work that arises from this thesis can be summarized in six lines:

- Extend the software development life cycle of these reports, not only to the European model, but also to the rest, which are not purely dimensional.
- Expand the use of this specification to other environments that are not Supervision and Regulation.
- Incorporate into this research the *Data Point Model* (DPM), and its implementation in XBRL.
- Through this development life cycle, facilitate the creation of public test games.
- Not only generate templates for SQL stored procedures, but also in other languages such as COBOL, .NET, Java, etc.
- Natural language processing analysis techniques to increase the semantic content of the reports.

Bibliography

Books

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, October 15th, 1999.

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 (Road map of the MDM (implemented in a design ROLAP) of FINREP 2012 and Solvency II (01-07-2012) with validations, Proof of Concept.). Final Project of the Polytechnic School of the Carlos III University of Madrid, Spain. Date: November 14th, 2013. Tutors: Ignacio Santos and Elena Castro. In progress.

Felden C. (2007) Multidimensional XBRL (New Dimensions of Business Reporting and XBRL). DUV Deutscher Universitäts-Verlag, pp 191-209, November 18th, 2007.

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

Jarke M, Lenzerini M, Vassiliou Y, Vassiliadis P (2003) Fundamentals of Data Warehouses, 2nd Edition 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 (Automation and mapping from the data model of the XBRL specification in Database). Final Project of the Polytechnic School of the Carlos III University of Madrid, Spain. Date: October 4th, 2012. Tutors: Ignacio Santos and Elena Castro. URI: <http://hdl.handle.net/10016/16843>.

Martín Quetglás G. (2006). Curso de XBRL. Pearson Education. 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). Final Project of the Polytechnic School of the Carlos III University of Madrid, Spain. Date: October 13th, 2015. Tutors: Ignacio Santos and Elena Castro.

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

Spanish Official Gazette (2007). General Chart of Accounts (“Plan General Contable”). Spanish Official Gazette (“Boletín Oficial del Estado”, BOE). November 20th, 2007, append 278.

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 Larosièrè Group (2009). The High-Level Group on Financial Supervision in the EU. Date: February 25th, 2009. The Larosièrè Group: Jacques de Larosièrè (Chairman), Leszek Balcerowicz, Otmar Issing, Rainer Masera, Callun Mc Carthy, Lars Nyberg, José Pérez, Onno Ruding, Brussels, Belgium.

Valencia, J. (2011), XBRL Meta-metadata Model. Final project of Computer engineering technology management, Carlos III University of Madrid. Date September 27th, 2011. Tutors: Santos I. and Castro E. It can be obtained in: <http://hdl.handle.net/10016/13157>

Journals

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. Volume 26 Issue 2, June 1997, pages 417-427.

Aljumaily H, Cuadra D, Martínez P (2014) Applying black-box testing to UML/OCL database models. *Software Quality Journal*. June 2014, Volume 22, Issue 2, pp 153-184.

Bennett M (2013). The financial industry business ontology: Best practice for big data. Date: July 1st, 2013. *Journal of Banking Regulation*, vol. 14, issue 3-4 (July/November 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. Date: July 18th, 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. Date: September 1st, 2002. *The Review of Business Information Systems*. Quarter 4, vol 6, number 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.

Debreceeny 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*, volume 2, issue 1, January 2001, pp 47-74.

Debreceeny 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. Date: June 1st, 2010. *Journal of Accounting and Public Policy*. Volume 29, edition June 3rd, 2010, pp 296-306.

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, August 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.

Gräning A, Felden C, Piechocki M (2011). Status Quo and potential of XBRL for business and Information Systems Engineering. Date: July 12th, 2011. *Business & Information Systems Engineering*, volume 3, issue 4, pp 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), pages 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, January 2006.

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

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

Loukis E N, Charalabidis (2013). An empirical investigation of information interoperability business in European Firms. Date: May, 2013. *Computers in Industry*, Vol. 64, issue 4, May 2013, pages 412-420.

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

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

Santos I, Castro E, Velasco M (2016). XBRL Formula Specification in the Multidimensional Data Model. *Journal: Information Systems*, volume 57, April 2016, pages 20–37. Doi:10.1016/j.is.2015.11.001

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*, volume 7, number 13, 2007, pp.1-25. ISSN: 1577-8517.

Spies M (2010) An ontology modelling perspective on business reporting. *Information Systems*, volume 35, issue 4, June 2010, pp 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*, volume 2, issue 3, September 2001, pp 169-191.

Conferences

Buder J, Koschtial C, Felden C (2009). Formalization of REA ontology. Date: August 6th, 2009. *Proceeding of the fifteenth Americas conference on information systems*, San Francisco, California, August 6th-9th, 2009.

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, August 4th-6th, 2006.

Declerck T, Krieger H U (2003) Traslating XBRL into description logic. An approach using protégé, sesame and OWL. *9th International Conference on Business Information Systems in cooperation with CAN SIGMIS*, Klagenfurt Austria, May 31st - June 2nd, 2006.

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

Díaz A (2012) Opening & Welcome (XBRL Week in Madrid hosted by Bank of Spain and Business Register, 9th XBRL Europe Day), May 11th, 2012. http://www.eurofiling.info/15th_worksop/presentations/AnselmoDiaz-welcome.pdf.

Golfarelli M, Rizzi, Urdoljak B, (2001). Data Warehouse design from XML Sources. Date: September 3rd, 2001. *Proceedings of 4th ACM International Workshop on Data Warehousing on OLAP*, pages 40 to 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. Date of conference: 18th-23rd July 2010, pages: 1 to 8, ISSN: 1098-7584.

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

Piechocki M, Felden C, Gräning, A (2007). Multidimensional XBRL Reporting. Date: March 25th, 2007. *Proceedings of the Fifteen European Conference on Information Systems*.

Romanelli M. (2007). Matrix Schemas for COREP & FINREP Templates November 6th, 2007. *VIII CEBS-XBRL Workshop*. Amsterdam, 6th-7th November 2007.

Santos I, Castro E. (2010) XBRL, UML and Databases: State of Art. XIII European Banking Supervisors XBRL Workshop. Luxembourg. November 24th-25th, 2010. Organised 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 7th International Conference on Web Information Systems and Technologies*,

WEBIST 2011, pp. 161-164, Noordwijkerhout, The Netherlands, May 6th-9th, 2011.
DOI: 10.5220/0003399001610164

Santos I, Castro E (2011b). Proof of Concept of mapping a XBRL reports versus a RDBMS. September 5th, 2011. *XBRL Openfiling 1st, 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. Proceedings of the IADIS International Conference Internet Technologies and Society Shanghai, China 8-10 December 2011. Edited by Piet Kommers, Ji-ping Zhang, Tomayess Issa and Pedro Isaías. ISBN (Book): 978-972-8939-55-7. *IADIS International Conference on Internet Technologies & Society (ITS 2011)*. Shanghai, China, December 8th-10th, 2011.

Santos I, Castro E (2012a). Validation of a XBRL Document Instance in a RDBMS, Proof of Concept. June 1st, 2012. *15th 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, April 19th, 2013. Hosted by the Central Bank of Ireland. www.openfiling.info/academy.

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

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

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

Snijders P. (2005) XBRL and Complex Data Mapping. Date: February 11th, 2005. Conference for Software Developers, Madrid, Spain,

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”. 15th XBRL EU, XBRL week in Madrid, hosted by Business Register and Bank of Spain, June 2nd, 2015. <http://www.eurofiling.info/201506/index.shtml>.

Zhu H, Fu L (2009). Quality of Data Standards: Empirical finding from XBRL. Date: December 15th, 2009. 30th International Conference on Information System (ICIS’90), December 15th-18th Phoenix, AZ, USA.

References in 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. Date: July, 16th, 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) Description of national adaptation to FINREP taxonomy (“Descripción de la adaptación nacional de la taxonomía FINREP”). November 26th, 2011. Bank of Spain.
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. Date: September 12th, 2010.
<http://www.bis.org/publ/bcbs189.pdf>, www.oecd.org/finance/financiaal-markets/45314422.pdf, 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). Date: January 27th, 2010. www.eurofiling.info/bsi-mirTaxonomies/Taxonomy.Shtml.

CEN (2013). Workshop Agreement CWA. « <http://cen.eurofiling.info/> ». Date: December 11th, 2013.

CEN (2014). Comité Européen de Normalisation/European Committee for Standardization. <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). November 30th, 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, Kannon D V (2006) XBRL Formula 1.0 Specification. 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>. EBA (2011). EBA BS 180 (CPT50 ITS on reporting-Annex III to V FINREP templates and instructions). Date: November 30th, 2011.

Eurofiling (2011) Proof-of-concept simple, Eurofiling data modelling and taxonomy architecture, June 2nd, 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.

EXGEN (2012). "Data Point Modeler, Excel XBRL Generator (Excel add in), February, 15th, 2012, www.openfiling.info/exgen/.

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

Hamscher W, Kannon D V (2000). *Extensible Business Reporting Language (XBRL) Specification*. Date: July 31st, 2000 por *XBRL International*. www.xbrl.org.

Hernández-Ros I, 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. Public Working Draft, February 4th, 2009. XBRL International Inc.

IFRS (2015) International Financial Reporting. www.ifrs.org/home.htm.

Inteco (2009), User guide API XBRL LENLOC PENLOC. Date: November 1st, 2009. Author: National Institute of Communications Technology of Spain (Instituto Nacional de Tecnología de la Comunicación de España, Inteco). 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.

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, V, Sever, W van, Boixo, I (2012). Data Point Modeller: EXGEN. Excel XBRL Generator (Excel add in). February 15th, 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.

SDMX (2015) Statistical Data and Metadata Exchange. <http://sdmx.org/>.

SEC (2015) U. S. Securities and Exchange Commission. 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. URLs: <http://www.ecb.europa.eu/paym/t2s/html/index.html>,
<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”, 15th XBRL EU, XBRL week in Madrid, <http://www.eurofiling.info/201506/index.html>.

XBRL España (2006). White Paper, Working Group of 2006 Technology (Libro Blanco, Grupo de Trabajo de Tecnología 2006). www.xbrl.es/formacion/formacion.html”.

XBRL Formula Overview 1.0.Public Working Draft, December 21st, 2011. XBRL International Public Working Draft.

XBRL International (2009) Formula 1.0 Recommendation 22 June 2009. <http://www.xbrl.org/formula-1.0-recomendation-22-june-2009>.

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

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

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

XBRL International and Novartis International AG (2002) Samples XBRL 1.0. November 15th, 2002. <http://www.xbrl.org/taxonomy/int/fr/ias/ci/pfs/2002-11-15/Samples.htm>.

XML.com (2015). O'Reilly "XMLO.com", using W3C XML Schema. O'Reilly Media, Inc. www.xml.com.

Acronyms

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

NAME	DESCRIPTION	REFERENCE
CWA	CEN Workshop Agreement	www.eurofiling.info/cen/cwa/ .
DBMS	Database Management System	
DM	Data Mart	
DPM	Data Point Model	
DTD	Document Type Definition	
DTS	Discoverable Taxonomy Set	Set of taxonomies (XML files) that are necessities for processing a XBRL document.
DW	Data Warehouse	
EBA	European Banking Authority	www.eba.europa.eu .
ECB	European Central Bank	
EDGAR	Electronic Data Gathering, Analysis, and Retrieval system.	www.sec.gov/edgar.shtml .
EDITRAN / EDI	Electronic data interchange Transactions / Electronic Data Interchange	www.editran.info .
EDOC	Enterprise Distributed Object Computing	http://ieeexplore.ieee.org/xpl/conhome.jsp?punumber=1000274 .
Eurofiling	Eurofiling community is an open joint initiative of the European Banking Authority (EBA) and the European Insurance and Occupational Pensions Authority (EIOPA) in collaboration with XBRL Europe, as well as stakeholders as banks, solutions providers, academy and individuals.	" http://eurofiling.info/portal/ .
EIOPA	European Insurance and Occupational Pensions Authority	www.eiopa.europa.eu .

NAME	DESCRIPTION	REFERENCE
EMU	Economic and Monetary Union	www.ecb.int/ecb/history/emu/html/index.en.html .
ESCB	European System of Central Banks	http://www.ecb.europa.eu/ecb/orga/escb/html/index.en.html .
E/RM	Entity/Relationship Model	
EU	European Union	
EUROSYSTEM	Euro System	It is the monetary authority of the Eurozone.
EUROZONE	Euro Area	States of the European Union with the euro (€) as currency.
EXGEN	Data Point Modeler: EXGEN. Excel XBRL Generator (excel add in)	http://www.openfiling.info .
FED	Board of Governors of the Federal Reserve System	http://www.federalreserve.gov/
FIN	Financial message	Service of SWIFT
FINREP	Financial Report (taxonomy)	http://www.eurofiling.info/finrepTaxonomy/taxonomy.shtml .
FIX	Financial Information eXchange.	
GML	Generalized Markup Language	
HOLAP	Hybrid OLAP	
HTML	HyperText Markup Language	
HTTP	HyperText Transfer Protocol	
HTTP-S/HTTPS	Hypertext Transfer Protocol Secure	
IAS/IFRS	International Accounting Standards/International Financial Reporting Standard	www.ifrs.org/home.htm .
IBM	International Business Machine	www.ibm.com .

NAME	DESCRIPTION	REFERENCE
IFRS	International Financial Reporting	www.ifrs.org/home.htm .
INTERACT	Interactive transactions	A protocol to exchange automated and interactive messages, synchronous or asynchronous used for SWIFT.
IPSEC	Internet Protocol security	It is used for SWIFT.
IS	Information Systems	
ISDA	International Swaps and Derivatives Association	http://www2.isda.org/ .
ISO	International Organization for Standardization/ Organización Internacional para la Estandarización	www.iso.org .
IT	Information Technology	
ITS	Internet Technologies & Society	http://its-conf.org/
ITU	International Telecommunication Union/ Unión Internacional de Telecomunicaciones.	www.itu.int .
MDA	Model Driven Architecture	www.omg.org/mda .
MDM	Multidimensional Data Model	
MIR	Monetary Financial Institutions Interest Rates (taxonomy).	
MOF	Meta-Object Facility	http://www.omg.org/mof/
MOLAP	Multidimensional OLAP	
MS	Matrix Schema	
NASDAQ	National Association of Securities Dealers Automated Quotations	http://www.nasdaq.com/ .
NCBs	National Central Banks	http://www.ecb.europa.eu

NAME	DESCRIPTION	REFERENCE
NSIs	National Statistical Institutes	http://epp.eurostat.ec.europa.eu
OLAP	On-Line Analytical Processing	
OMG	Object Management Group	www.omg.org
OWL	Web Ontology Language	www.w3.org/TR/owl-ref
PIM	Platform Independent Model	MDA
POC	Proof of Concept	
PSM	Platform Specific Model	MDA
RDBMS	Relational DBMS	
RM	Relational Model	
ROLAP	Relational On-Line Analytical Processing	
SAX	Simple API for XML	Parser XML
SDMX	Statistical Data and Metadata eXchange	www.sdmx.org
SEC	U.S. Securities and Exchange Commission	www.sec.gov
SEPA	Single Euro Payments Area	http://www.ecb.europa.eu/paym/sepa/html/index.en.html
SGML	Standard Generalized Markup (ISO 8879-1986)	www.w3.org/Markkup/SGML
SPEM	Process Engineering Metamodel	http://www.omg.org/spec/SPEM/2.0/
SQL	Structured Query Language	Special propose programming language for relational database (RDBMS).
SSE	Shanghai Stock Exchange	http://english.sse.com.cn/
SZSE	ShenZhen Stock Exchange	http://www.szse.cn/main/en/

NAME	DESCRIPTION	REFERENCE
SWIFT	Society for Worldwide Interbank Financial Telecommunication/Sociedad para las Comunicaciones Financieras Interbancarias Internacionales	www.swift.com
T2	TARGET2	http://www.ecb.europa.eu/paym/t2/html/index.en.html
T2S	TARGET2 Securities	http://www.ecb.europa.eu/paym/t2s/html/index.en.html
UD	Universe of the Discourse	
UML	Unified Modelling Language	
URI	Universal Resource Identifier	
URL	Universal Resource Locator	
US-GAAP	United States-Generally Accepted Accounting Principles	
WAN	Wide Area Network/ Red de Área Extensa	
XBRL	eXtensible Business Reporting Language	www.xbrl.org
XBRLDM	XBRL Data Model	
XDT	XBRL Dimensional Taxonomies	XBRL Dimensional 1.0
XLink	XML Linking Language	www.w3.org/TR/xlink/
XMI	XML Metadata Interchange	Standard for exchanging metadata information through XML. http://www.omg.org/spec/XML/
XML	eXtensible Markup Language	
XPath	XML Path Language	http://www.w3.org/TR/xpath20/
Xpointer	XML Pointer Language	www.w3.org/TR/xptr/

NAME	DESCRIPTION	REFERENCE
Xquery	XML Query Language	www.w3.org/TR/xquery/
W3C	World Wide Web Consortium	www.w3.org

Glossary

A

Allowed fact

User constraint.

Assertion

It is a general constraint that does refer to a fact or set of facts.

Attribute of dimension

Set of concepts of a dimension.

Attribute by default

Each domain has a concept by default.

B

Basic Concept

It is a special concept that has associated a data type, a time period type, and a balance type. A basic concept defines the name of a concept by type, precision and unit of measure.

Base dimension

It is a domain with only basic concepts.

C

Calculated attribute

An aggregate of dimension attributes of a dimension, and/or calculated attributes.

Capital Requirements Directives (CRD)

Framework of the financial supervision in the European Union, based on Basel II and Basel III rules on capital measurement and capital standards.

Conceptual Data Model

This model shows the semantics of real world and depicts a set of assertions, trough of entities and relationships.

Constant

It is a value or expression yielding one result and whose expression is deterministic.

Computation Independent Model (CIM)

The business or domain model. In this level of the MDA, the real world is analysed, including concepts, data and rules.

Concept

It is the definition of a business concept or item. Each concept is associated to a time period type attribute (Instant, Period, and Forever).

COREP

It is a standard issued by the European Banking Authority (EBA), for Capital Requirements Directive reporting, it covers credit risk, market risk, operational risk, own funds and capital adequacy ratios.

D

Dimension

It is a set of concepts of a domain. These concepts have a tree-like structure.

Dimension group

Group of dimensions of a domain.

Discoverable Taxonomy Set (DTS)

Set of taxonomies (XML files) that are necessities for processing a XBRL document.

Document Object Model (DOM)

It is a platform that will permit to programs and scripts to access and update the content, structure and style of documents.

Domain

It is a group of concepts belonging to a field or scope of knowledge or activity. In this model a domain can contain basic concepts or non-basic concepts but not both.

E

Explicit dimension

This is explicit if the dimension attributes are defined.

Explicit filter

it is a semantic constraint to which all facts, groups of facts or assertion facts are circumscribed.

F

Fact

A fact is a value representing a particular measurement provided by the reporting entity.

FallbackValue

It defines a value by default for a fact or set of facts if the latter is null.

Filter

It is set of dimensions whose assertion is confined.

Filter group

It defines a set of explicit filters to which all assertions are circumscribed.

FINREP

Covers consolidated and sub-consolidated financial reporting for supervisory purposes based on International Accounting Standards (IAS) and the International Financial Reporting Standard (IFRS) and is issued by the EBA.

Forbidden fact

User constraint.

H

Hierarchical Constraint

Concepts in a dimension have a tree-like structure. Validation between a leaf and its leaves below, that is to say, it used for the calculated attributes.

I

Implicit dimension

It is implicit if they are not defined.

Implicit filter

it is a constraint inherent in all facts circumscribed by an assertion (i.e., the assertion is constrained to the facts circumscribed by the set of dimension-dimension attribute inherent pairs.).

M

Model Driven Architecture (MDA)

In software engineering provides a good framework for the automatic generation of code for application development. It focuses on using models as approaches to cover the life cycle of software development. The heterogeneity and interoperability problems between systems with different implementation platforms are resolved by using this approach. The MDA stratifies the design in three phases or levels to allow for easier development.

Multidimensional Data Model (MDM)

It is a straightforward model that combines objects, dimensions (hierarchies), measures and attributes for representing real work business problems. In addition, this model is the heart of On-Line Analytical Processing (OLAP), which requires complex queries that can be solved by the MDM in real time.

P

Platform Independent Model (PIM)

This focuses on high-level business logic in the MDA, without considering the features of the implementation technology of the system.

Platform Specific Model (PSM)

This represents the detail of using a specific platform for a system in the MDA.

Precondition, or variable attribute or assertion

Defines the facts that can be evaluated.

R

References

References to directives or laws of the concepts.

S

SAX (Simple API for XML)

It is an event-driven online algorithm for parsing XML documents, with an API interface developed by the XML-DEV mailing list.

Solvency II

It is a directive of the European Union (EU) which codifies and harmonizes EU insurance regulations and is issued by the European Insurance and Occupational Pensions Authority (EIOPA).

V

Variable

It is an atomic element of an expression in an assertion, which depicts a fact or group of facts.