Proceedings of the Federated Conference on
Computer Science and Information Systems pp. 1391–1395
ISBN 978-83-60810-51-4

An IFRS Compliant Balance Sheet Metamodel

Nenad Krdzavac, Rafiqul Haque, and Tom Butler
Department of Accounting and Information Systems College of Business and Law,
University College Cork, Cork, Ireland,
Email: { N.Krdzavac, ARHaque, Tom.Butler}@ucc.ie

***Abstract*—This paper proposes a balance sheet metamodel using Model Driven Architecture (MDA) methodology. In order to do that, we use the international financial reporting standard (IFRS) as a good starting point owing to its wider adoption across countries in preparing balance sheet. The balance sheet is summary of financial position of a financial entity such as credit institutions. The reason for applying MDA in developing balance sheet metamodel is twofold: to automate transfer and sharing knowledge of regulations of capital adequacy of credit institutions and to make a clear difference between conceptual and concrete modeling of regulations of capital adequacy of credit institutions.**

I. Introduction

TO CREATE interoperable balance sheet in the scope of capital adequacy system for credit institutions that can be easy to maintain and reuse, the Model Driven Architecture (MDA) standard [13] can be used. The core concept in this paper is to change the system's specification rather than implementation [14] using the Unified Modeling Language (UML) [20].

A balance sheet is the summary of the financial position of a business partnership, sole proprietorship, and corporation [9]. It is also described in [21] as a *snapshot of the financial condition*. The primary components of a balance sheet entail *asset*, *equity*, and *liability* [9]. International Financial Reporting Standard (IFRS) [9] has recently been adopted as a single standard by many countries for presenting balance sheet. It is worth noting that balance sheet is one of the four types of IFRS financial statement.

The ultimate goal of this research is automatization of preparing the balance sheet of a financial institute consistently and accurately. We have designed a research *roadmap* consisting of multiple *success points*. This paper is the first milestone of our research. In this paper, we propose a meta-model for IFRS compliant balance sheet. Additionally, we briefly explain how to perform reasoning with balance sheet models. We choose IFRS standard due to its wider adoption across the continents.

II. IFRS Standard - A Short Overview

The IFRS standard was adopted by International Accounting Standard Board(IASB) in 2001 [19] with an aim to provide universal structure and rule to assembling the financial information for a given period. IFRS guides the financial professionals across the countries to understand and prepare financial statements. IFRS has been adopted by well beyond expected number of countries. IFRS supports preparing four different types of financial statement embodying *statement of financial position (better known as Balance Sheet)*, *statement of comprehensive expense*, *statement of change in equity*, and *statement of cash flow* [8]. Each statement contains different components and has different purposes. Note that, only balance sheet is included within the scope of our proposed meta-model that is presented in this paper.

Digitalization of paper-based balance sheet is an ongoing initiative that has been started lately. The goal of this initiative is to represent IFRS using eXtensible Business Reporting Language (XBRL). XBRL is a member of eXtensible Markup Language (XML) family. It provides a common and electronic format for reporting business and finance of an organization [9]. Current version of XBRL lacks accuracy—is argued in [3].

III. Related Work

There is an effort to develop ontology-based expert systems for rating financial position of a financial entity [18]. They employed specific methodology in order to improve reusability and interoperability of the financial expert system. The model and implementation proposed in [18] cannot provide full interoperability. The implementation framework [18] uses Java classes as well as ontologies and rule-based reasoning. Such framework cannot be used to create heterogeneous financial system that interoperates at the level of standard component interfaces [14]. The Jess expert system shell [10] is used as inference engine. However, there are reasoning tasks that are difficult for Jess to solve particularly, it can not store reasoning process in machine readable format, such as XML, that is using for reasoning process debugging and it can not explain inconsistency more precisely. To overcome such shortcomings, we propose to use model transformations from balance sheet model to the tableau model [12].

IV. Problem Statement

To develop adaptable capital adequacy system to be able to presenting and analyzing the balance sheet of credit institutions, it is important to define the system requirements more specifically. These requirements are: the system has to be designed based on the MDA standard; the system has to deliver a machine readable format for storing balance sheets, in a way independent of the tools that can use them; it has to support the deployment of financial components including assets, liabilities and equites across a wide range of capital adequacy systems; it has to provide other systems with the ability to report balance sheet results in an intelligent manner by using for instance, description logics (DLs) reasoning techniques based on model transformations [2]. By creating a system to achieve the above requirements (taken from [14]), one should obtain: a flexible and quickly developed capital adequacy environment to be able to using balance sheet; high-level of interoperability between components within a financial system; easy testing of balance sheet consistency, done in an intelligent way; an easy-to-extend capital adequacy system that can be improved by including new subsystems.

V. A Balance Sheet Metamodel

To guarantee interoperability between homogeneous and heterogeneous capital adequacy systems it is important to use MDA standards in capital adequacy system development, as follows:

- the interoperability in heterogeneous environments could be achieved via shared metadata of subsystems that create the capital adequacy system.
- According to [16], strategy of sharing metadata consists of development, publishing and interpreted models.

The expressive power of a model depends on defined metamodel [4] because, to define model we use concepts defined in a metamodel. The proposed balance sheet metamodel captures main concepts suggested by IFRS Standard [9] related to capital adequacy of credit institutions. As noted in section II, balance sheet specifies the summary of financial position of a financial entity e.g. credit institutions [9].

The following subsections provide balance sheet metamodel. We use Eclipse Modeling Framework [5] as a tool for development our metamodel.

A. Balance Sheet Metapackages

According to IRFS standard [9], we have defined the following metapackages (see Fig. 1):

- Assets metapackage;
- Liabilities matapacke;
- Equities metapacke;

The $Equities$ metapackage depends on $Assets$ and $Liabilities$ metapackages (see Figure 1). In financial practice, an equity uses asset and liability in final presentation of balance sheet of a credit institution [9].

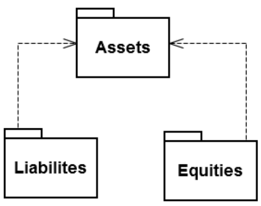

Fig. 1. Balance sheet metapackages

B. Assets Metapackage

Core metaclass in $Assets$ metapackage is $Asset$. The UML-based metamodel of assets metapacke is shown on Figure 2. The metaclass $Asset$ represents a balance sheet item representing what a firm owns. Metaclasses $CurrentAsset$ and $FixedAsset$ inherit the metaclass $Asset$ and at the same time there is composition relation between the $Asset$ metaclass and these two metaclasses. Fixed asset is also known as non-current asset [9].

An asset must have at least one fixed asset and current asset as its parts (part-of relation). It is represented by cardinality at the end of composition relation between the metaclasses. Metaclass $CurrentBiologicalAsset$ depends on $CurrentAsset$ metaclass. Also, metaclass $FixedAsset$ is linked to $TangibleAsset$, $IntangibleAsset$ as well as $FixedBilogicalAsset$ via composition relation. The balance sheet total is composed of fixed assets and current assets. An entity can have very complex transaction [9] which are described by $CurrentBiologicalAsset$ and $FixedBiologicalAsset$ metalcasses.

The metaclass $CurrentAsset$ is linked to the $Cash$ metaclass which is very important in case of presenting *"cash on hand, demand deposits"* and *"other short-term highly liquid investments"* [9]. The metaclasses that are in charge of payment and investments are $PrePayment$, $ShortTermInvestment$ and $DerivativeFinancialInsturment$ (see Figure 2). The $PrePayment$ metaclass represents costs and other activities related to future transactions presented as pre-payment activity [9]. $ShortTermInvestment$ includes stocks and bonds or any other business which can be liquidated quickly [9]. The metaclass $DerivativeFinancialInsturment$ represents derivative instruments (or simply derivatives) as financial instruments whose value is derived from the values of some other financial instruments or variables. Metaclass $Inverntory$ is linked to $CurrentAsset$ metaclass. $Trade$ metaclass is linked to the $CurrentAsset$, includes *"progress billings not yet paid by customers and retention"* [9].

$TangibleAsset$ metaclass represents assets that have a physical form [9], while $IntangibleAsset$ metaclass represents non-physical form of asset [9]. Examples of non-physical forms of asset embody patents, copyrights, brand names like the Google (www.google.com) etc. Both, tangible and intangible assets are part of fixed asset [9]. The $TangibleAsset$ metaclass is linked to the following metaclasses:

- *Property* represents houses, cars, and other physical substances;
- *Plant* includes for example vineyards, followers, etc;
- *Equipment* represents machineries [9]. For example computers in offices, manufacturing machines.

$IntagibleAsset$ metaclass contains the following attributes:

- *goodwill* denotes the reputation or brand value of an organization. Goodwill in real-world is a non-physical object that has a value.

$IntagibleAsset$ contains $IntellectualPropertyRight$ which

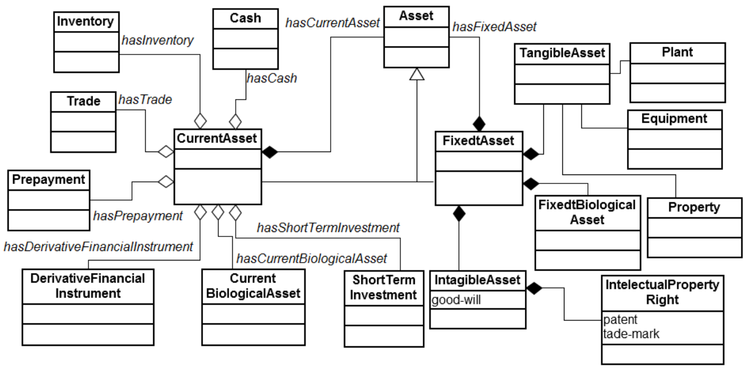

Fig. 2. Assets metapackage

is a meta-class shortly called IPR. IPR incorporates the following attributes:

- *patent* represents an exclusive right for an innovation, given to an organization by the government.
- *trade-mark* indicates the logo, theme phrase of an organization. A trade-mark is approved by the government and it always has value.

C. Liabilities Metapackage

Central metaclass in this package is $Liability$ which is linked, via composition relation, to $CurrentLiability$ and $longTermLiability$ metaclasses (see Figure 3). The $Liability$ metaclass represents current obligations of a financial entity or corporation [9]. The metaclasses contains the $borrowing$ and $provision$ attributes. The $borrowing$ attribute represents the amount of money that a financial entity has borrowed [9]. As oppose to burrowing attribute, $provision$ attribute represents a liability that does not have any given period [9]. The metaclasses $CurrentLibility$, and $LongTermLibility$ inherit $Liability$ metaclass (see Figure 3). These two metaclasses inherit also borrowing and provision attributes. When borrowing is an attribute of $LongTermLibility$ metaclass then it denotes the amount that can be repaid after a year from the date it was borrowed. Nonetheless, when borrowing is an attribute of $CurrentLiability$ metaclass then the amount has to be repaid within a year.

The $CurrentLiability$ metaclass is linked to the following metaclasses defined [9]:

- *ShortTermLoan* represents a loan which is repayable within short period of time (less than a year);
- *OverDraft* represents the amount that has been withdrawn by an individual;
- *OtherPayable* represents the financial obligation (usually money) that has to be paid off within short-period of time to the creditor. OtherPayable and *Account Payable* can be used interchangeably in a balance sheet because semantically they are same;
- *CurrentTaxLiability* represents the amount that of financial entity is obliged to pay for the current calendar year to the government. In a balance sheet for credit institutions, current tax liability is part of current liability. This is defined in the metamodel using composition association between metaclasses $CurrentLiability$ and $CurrentTaxLiability$.

$LongTermLiability$ metaclass is associated to the following metaclasses:

- *DeferredTaxLiability* denotes amount payable at some point of time in future. This amount although company owes but does not pay within current calender year, for which it is called deferred tax liability;
- *LongTermBond* is a contract that can be liquidate after some or many years that is called *maturity date*. The contract issued by a financial institute to the bond holder and remains valid until the maturity period is over. During maturity period, the bond issuer pays interest to the bond holder; This implies the long-term bond holder receives the principal amount after some or many years;
- *LongTermLoan* is a loan that a burrower has to repay after one or more years.

$LongTermLiability$ metaclass contains *debenture* attribute which refers to evidence (certificate) of loan that a debenture seller owes to debenture holder. This is an attribute of long-term liability because the debenture issuer repays to debenture holder after a year or more [9].

D. Equities Metapackage

Equity in a balance sheet represents residue after subtracting liability from asset [9]. The $Equity$ metaclass contains $option$ attribute that denotes a contract between an issuer who sells the option and a holder who buys it for future transaction at a reference price (see Figure 3). $OwnershipEquity$ inherits $Equity$ metaclass. Note that, when ownership is distributed among the shareholders, then it is called $shareholderequity$. For example, if a corporation is owned by sole proprietor then the equity of balance sheet is single owner's equity, whereas, shareholder equity denotes a corporation is owned by many owners through owning common or preferred stock.

The $Stock$ metaclass refers to the ownership of a company and gives the right to claim assets and earnings. $Stock$

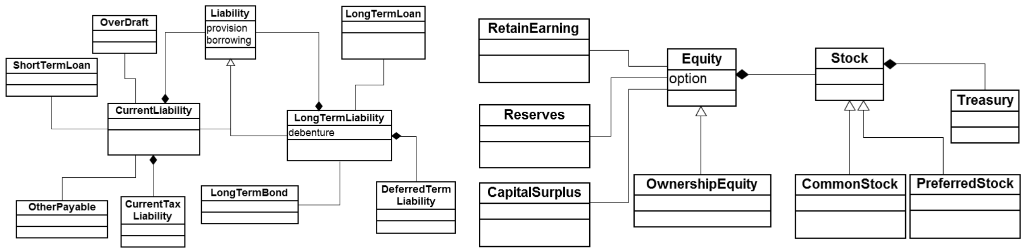

Fig. 3. Liabilities and equity metapackages

metaclass depends on $Equity$ metaclass has the following sub-metaclasses:

- *PreferredStock* represents the ownership of a company without any voting right;
- *CommonStock* represents the ownership of a company and gives the stockholders voting right.

It is worth mentioning that every stock can be liquidated. This implies that preferred and common stocks have values which represents money. $Treasury$ metaclass depends on $Stock$ metaclass and includes the amounts of shares that a company stored in its treasury. In a balance sheet model treasury is a part of stock [9].

$Equity$ metaclass is associated to the following metaclasses:

- *Reserve* denotes the amount of funds that credit institution must hold opposite to customers' reserves to meet any potential unexpected financial crisis.
- *RetainEarning* represents a percentage of net income that is kept by a company and reinvested in its core business or paid debt [9].
- *CapitalSurplus* denotes the amount raised by an organization. The amount is raised in excess of the nominal value of the company shares. It is also called *share premium*.

VI. Ecore Representation of Balance Sheet Metamodel

Eclipse Modeling Framework EMF (Core) [5] is project focused on development of model based technologies within the Eclipse. It includes technologies for model transformations, querying models, code generation, validation models [5]. In this section we shortly explain how to transform UML balance sheet metamodel into Ecore-based metamodel using Eclipse Modeling Framework tools [5]. The transformation is important because of creating, exchanging, and searching balance sheet models in MDA technological space [7]. First step is to transform UML based metamodel into genmodel, then to transform the genmodel into Ecore-based metamodel and corresponding Java interfaces.

From the balance sheet genmodel we generate the set of Java packages related to $modelcode$, $editcode$, $editorcode$ and $testcode$ [5]. For example, figure 4 shows Java packages generated for $Assets$ metapackage.

Each Java package contains interfaces and classes with methods for instantiation of metaclasses. The interfaces extend $EFactory$, $EObject$, $EObjectImpl$, $EFactoryImpl$ interfaces and classes. For example, if we consider $Asset$ metaclass, the following public interfaces and classes will be generated from balance sheet genmodel [5], [6]:

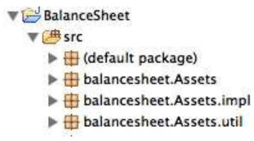

Fig. 4. Java packages generated from the Assets metapackage

- $Asset$ interface that extends $EObject$ interface;
- $AssetImpl$ class that extends $EObjectImpl$ and implements $Asset$ interface;

For each metapackage, corresponding interfaces will be generated. In case of $Assets$ metapackage, the following Java packages will be generated:

- $Assets$ package that contains interfaces for each metaclass i.e. representation of the model object for each metaclass, as well as,
  - $AssetsFactory$ class that provides create method for each non-abstract class of the model;
  - $AssetPackage$ interface that contains accessors for meta objects to represents each class, each feature of a class and each data type
- $Assets.impl$ package that contains an implementation of the model object i.e. the Factory [5].
- $Assets.util$ package that provides
  - an adapter ($AssetsAdapterFactory$ class) to create method for each class of the balance sheet model
  - the "Switch" [5], [6] for the model's inheritance hierarchy

EMF [5] contains the graphical editor which allows users to check validation of a metamodels.

A. A Balance Sheet Model

To create a balance sheet model which conforms to balance sheet mdetamodel, we use a part of consolidated balance sheet example taken from [9]. First step is to create an object *model* of $AssetsFactoryImpl$ class (line 13 on Figure 5). Then, we create asset and related fixed and current assets (see 15, 16 and 19th line shown on Figure 5). The same object "*model*" is using to create other parts of fixed and current assets.

```
package bsmodel;

import balancesheet.Assets.impl.AssetsFactoryImpl;

public class AssetModel {

    /**
     * @param args
     */
    public static void main(String[] args) {
        // TODO Auto-generated method stub

        AssetsFactoryImpl model = new AssetsFactoryImpl ();

        model.createAsset();
        model.createFixedAsset();
        model.createIntangibleAsset();
        model.createFixedBiologicalAsset();
        model.createCurrentAsset();
        model.createInventory();
        model.createCash();

    }

}
```

Fig. 5. Creating an asset using Java

B. Reasoning With a Balance Sheet Model

Reasoning with a balance sheet model is based on model transformation from a balance sheet model to a tableau model [12] (see Figure 6). DLs reasoning architecture is adapted from [15] to support reasoning service with balance sheet models. The first step is to transform a balance sheet model into OWL model (conforms to OWL metamodel) [17] which is an important part of Ontology Definition Metamodel (ODM) [17]. The second step is to transform corresponding owl model to a tableau model. To automate this process, we propose to use Atlas Transformation Language (ATL) [1].

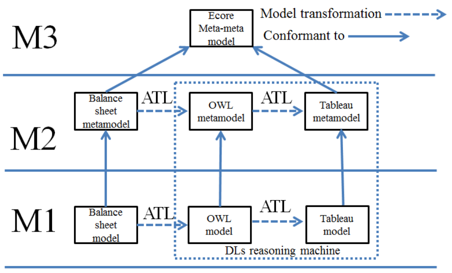

Fig. 6. Reasoning with a balance sheet model(updated from [14])

VII. Conclusion

The balance sheet metamodel proposed in this paper is an important research step with a plenty of practical applications. The metamodel can help software industry to apply MDA standard in developing flexible and interoperable capital adequacy systems which must use balance sheet motamodel. In order to use proposed balance sheet metamodel, they need to create model transformations using, for example ATL. Since the metamodel makes clear differences between conceptual and concrete modeling, developers will be able to create mechanisms to share and transfer knowledge automatically. When new capital adequacy regulations appear, it is easier manage the changes in balance sheet metamodel then the implementation. We have proposed tableau deduction system to check consistency of balance sheet model.

Acknowledgment

This research reported in this paper is sponsored by Enterprise Ireland (EI) as part of the project Governance Risks and Compliance Technical Competency (GRCTC) center to University College Cork, Ireland. The authors are grateful to Patrick O'Sullivan, Peter O'Sullivan, John Lombard, and Leona O'Brien for their support. The authors are also grateful to the anonymous reviewers for their comments and suggestions.

References

[1] Atlas Transformation Language - ATL Starter's Guide, ver. 0.1, ATLAS group, Nant, France. 2005. [Online]. Available: http://www.eclipse.org/m2m/atl/doc/ATL_Starter_Guide.pdf, Last visited: May, 2012.

[2] F. Baader, U. Sattler, An Overview of Tableau Algorithms for Description Logics, *Studia Logica*, vol. 69, no. 1, pp.5–40, 2001.

[3] Bartley, J. W., Chen, Y. A., Taylor, E. Z., A comparison of XBRL filings to corporate 10-ks—evidence from the voluntary filing program, SSRN, 2010.

[4] J. Bézivin, S. Gérard, P. A. Muller, and L. Rioux, MDA Components: Challenges and Opportunities, *In Metamodelling for MDA*, First International Workshop, York, UK, November 2003.

[5] Eclipse Modeling Framework. [Online]. Available: http://www.eclipse.org/modeling/emf/, Last visited: April, 2012.

[6] Eclipse Documentation. [Online]. Available: http://www.eclipse.org/documentation/, Last visited: May, 2012.

[7] D. Djurić, D. Gašević, V. Devedžić, The Tao of Modeling Spaces, *Journal of Object Technology*, vol. 5. no. 8, pp. 125-147,2006.

[8] IFRS for SMEs - Illustative consolidated financila statements, *Price Water House Coopers*, Dec. 2010. Online Available: http://www.pwc.com/gx/en/ifrs-reporting/illustrative-financial-statements/ifrs-for-smes.jhtml, Last visited: May 2012.

[9] IFRSs and XBRL. [Online].Available: http://www.ifrs.org/XBRL/XBRL.htm, Last visited: May, 2012.

[10] Jess, the rule engine for Java platform. [Online]. Available: http://herzberg.ca.sandia.gov/, Last visited: May, 2012.

[11] N. Krdžavac, D. Djurić, V. Devedžić, *MDA-Based Architecture of a Description Logics Reasoner*, In Prof. of Artificial Intelligence Applications and Innovations (AIAI2006), Springer Boston, pp.98-105, 2006.

[12] N. Krdžavac, V. Devedžić, *A Tableau Metamodel for Description Logics*, In Proceedings of The Thirteenth Workshop on Automated Reasoning, Bristol, UK, pp. 17–18, 2006.

[13] MDA Guide Version 1.0.1. [Online].Available: www.omg.org/docs/omg/030601.pdf, Last visited May, 2012.

[14] S. Radenković, N. Krdčavac, V. Devedžić, *A QTI Metamodel*, In Proceedings of the International Multiconference on Computer Science and Information Technology, pp. 1123-1132, 2007.

[15] S. Radenković, N. Krdžavac, and V. Devedžić, *MDA and Semantic Web Technologies for Assessment Systems*, In Proceedings of the SWEL'08 Workshop on Ontologies and The Semantic Web for Intelligent Educational Systems, held in conjunction with The 9th International Conference on Intelligent Tutoring Systems, ITS 2008, 06/2008, Montreal, Canada, pp. 65–69, 2008.

[16] (2002) OMG XMI Specification, ver. 1.2, OMG Document Formal/02-01-01. [Online].Available:http://www.omg.org/cgibin/doc?formal/20020101.pdf

[17] Ontology Definition Metamodel, ver. 1.0. [Online].Available: http://www.omg.org/spec/ODM/1.0/, Last visited: May, 2012.

[18] Li-Y. Shue, Ching-W. Chen, W. Shiue, Expert Systems with Applications,*An International Journal Archive*, vol. 36 no. 2, March 2009.

[19] The Move IFRS and XBRL, May, 2008. [Online]. Available: http://blog.technologyevaluation.com/blog/2008/12/05/the-move-to-ifrs-and-xbrl/

[20] UML 2.0 Infrastructure Overview. [Online]. Available: http://www.omg.org/issues/UMLSpec.pdf, Last visited: May, 2012.

[21] Williams, Jan R., Susan F. Haka, Mark S. Bettner, Joseph V. Carcello: Financial and Managerial Accounting. McGraw-Hill Irwin, 2008, pp. 40. ISBN 978-0-07-299650-0.