Volume 3, Issue 11, November 2013 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Transformation Rules According to MDA Approach for a Dependable Data Warehouse System

Imane Hilal
RITM Lab., ESTC,
CED ENSEM Hassan II University,
Casablanca, Morocco

Nadia Afifi ,
RITM Lab., ESTC
Hassan II University
Casablanca, Morocco

Mohammed Ouzzif
RITM Lab., ESTC
Hassan II University
Casablanca, Morocco

Reda F.Hilali , Hicham Belhadaoui
RITM Lab., ESTC
Hassan II University
Casablanca, Morocco

Abstract— Data Warehouse systems can extract relevant information from a large volume of data. Since the purpose of this information is to facilitate the decision making, it's important to ensure Dependability of the system which delivers it. Thereby the dependability's requirements must be taken into account in the early stages of the development process of the Data Warehouse system (DWS). In previous work, we proposed to adapt the MDA (Model Driven Architecture) approach to model dependable DWS, by defining CIM (Computation Independent Model) and PIM (Platform Independent Model) of dependability's aspects. And we also demonstrated a relationship between the two models. In this article, we focus more on transformations from the CIM to PIM to ensure automatic management of these models on the one hand, and to keep traceability's link of dependability's requirements through the different stages of MDA on the other hand.

Keywords— Data Warehouse Systems, MDA, Model Transformation, Dependability, Atlas Transformation Language.

I. Introduction

The role of Decision Support Systems and more specifically the Data Warehouse System (DWS), is to support analytical queries that may overcharge the transactional system [1]. To do this, it extracts the raw data from the latter, cleans and consolidates them, in order to store them according to a philosophy which is oriented analysis. It aims to provide synthetic information at the appropriate time in order to facilitate and improve the process of decision making [2]. This is why, the DWS must justify some level of confidence in order to be able to trust the information they deliver. Hence it's important to take into account the dependability's requirements of the system since the early stages of its development process. In a previous work [3], we proposed to adapt the MDA approach to model dependability of DWS. Thus, we defined the CIM and PIM models regarding dependability's aspects which are: availability, reliability, maintainability and security. And we modeled their integration to better take into account their interaction. However, we have not developed the transformation between the two models; we have only demonstrated its existence. In this work, which represents a logical extension of our previous contribution, we explain with details the transformations between CIM and PIM models. Our goal is to provide an automated way to manage models on the one hand; and on the other hand to keep traceability of dependability's requirements through the different abstraction levels of MDA.

This article is organized as follows: In Section 2, we present a related work of models transformation in the context of the DWS. In Section 3 we present the key concepts that we have used to realize our work: MDA approach, model transformation, and Atlas Transformation Language (ATL). Section 4 is dedicated to the transformation rules that we have developed. And we finish with a conclusion and perspectives.

II. Related Work

In the context of DWS, the MDA approach has been implemented by Mazon et al [4] who have aligned all of the DW development process to MDA, using models for each stage of development and specifying transformations between them through the QVT (Query / View / Transformation) standard. On the other hand, the same authors proposed a set of QVT transformation rules to ensure that the DW developed from the user requirements is valid with respect to the data sources which will be supplied [5]. However, these works have particularly focused on the functional part and have not address the particular case of non-functional requirements.

Soler et al. [6] proposed a security requirement model for the DW based on MDA approach. This approach has been extended by Carlos Blanco [7] who has developed a secure DW, where the security requirements have been involved at each stage of development process. This is the only approach that considers security as non-functional aspect from the early stages of DW development. The authors have also defined the rules of transformations between PIM and PSM (Platform Specific Model). However security is not the only aspect to ensure a dependable DWS, there are other aspects involved such as availability, reliability and maintainability. These were treated separately or in combination, but to our knowledge never aligned with the MDA. Finally, the transformation of models representing the dependability of DWS has not yet been addressed in its entirety, only security has been developed in this context given the confidential and critical nature of DW.

III. Model Driven Architecture (MDA) and models Transformation

A. MDA approach

MDA is an initiative of the OMG [8], which is defined as architecture and a development approach. Its goal is the separation between system functional specifications and details of its implementation on a platform. In this sense, MDA proposes a structured specification architecture in several types of models: (i) Computation Independent Model (CIM) that allows to model the system requirements and make easier the understanding of the problem, (ii) Platform Independent Model (PIM) that describes the business information separated from platform details, (iii) Platform Specific Model (PSM) which allows to reproduce the PIM taking into account the details of the platform based on the PDM (Platform Description model).This latter provides platform necessary information [9].

B. Models Transformation

The OMG defines QVT standard (Query / Views / Transformation) which represents the meta model for the elaboration of models of transformation.

Three approaches of models transformation are recognized [10]:

- Programming Approach: based on programming languages, generally object-oriented, to describe the transformations.
- Template Approach: based on templates for the target models. The transformation consists on replacing the target canvas parameters by the source model values.
- Modelling Approach: it consists on modelling the transformation for making them more productive. MOF 2.0 QVT standard (Meta Object Facility) has been proposed by the OMG [11] to define the meta-model transformations. In this approach the transformations are based on rules that describe the mapping between the source and destination entities. For example this kind of transformation is supported by the ATL transformation language. In this paper, we use the ATL to describe our transformation.

C. Atlas Transformation Language (ATL)

ATL is designed to perform transformations in context of MDA approach. ATL is a hybrid language because it allows both declarative and imperative transformations. Its abstract syntax is defined by MOF [12] as shown in Fig. 1. It also has a textual concrete syntax.

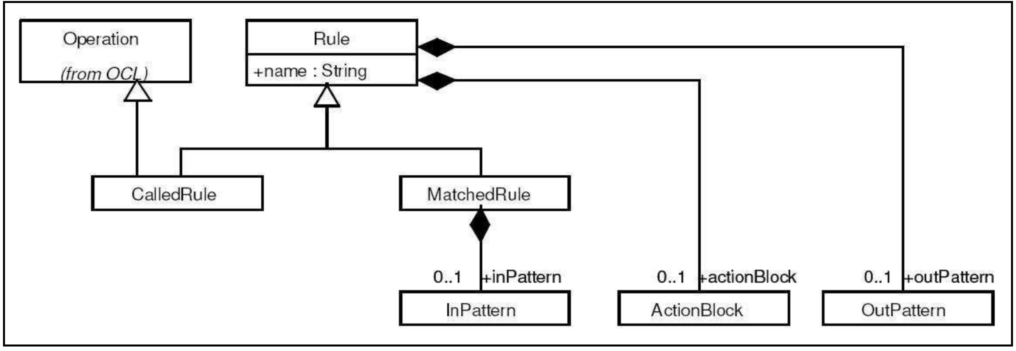

Fig. 1 Abstract syntax of transformation rule extracted from the meta model ATL[13]

A transformation model is used to transform source models to target models based on their metamodels which are conform to MOF. It is described by a set of transformation rules. ATL can navigate in the models using OCL (Object Constraint Language) queries. Before the execution of a transformation model, it must be consistent with: (i) ATL meta-model; (ii) source meta-model; (iii) destination meta-model; as explained in Fig. 2.The transformation rules are applied by searching the source model elements that correspond to their pattern source to transform them into elements of the target model.

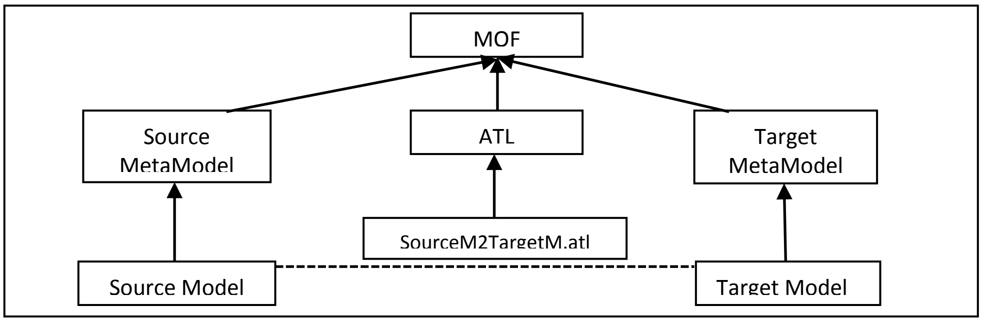

Fig. 2 Mechanism of ATL transformations

In the next section we will present our approach and the ATL transformation rules to ensure the passage of the CIM to PIM.

IV. Transformation Rules

A. Overview of our approach

Our proposal [3] aims to provide a process design for dependable DWS, taking into account the constraints of dependability since the early stages of modelling. As these vary according to the intended use of the DWS, we propose generic approach aligned to the standard MDA. Our approach allows their integration in a refined way, taking into account their interaction. Fig. 3 represents our approach explicitly.

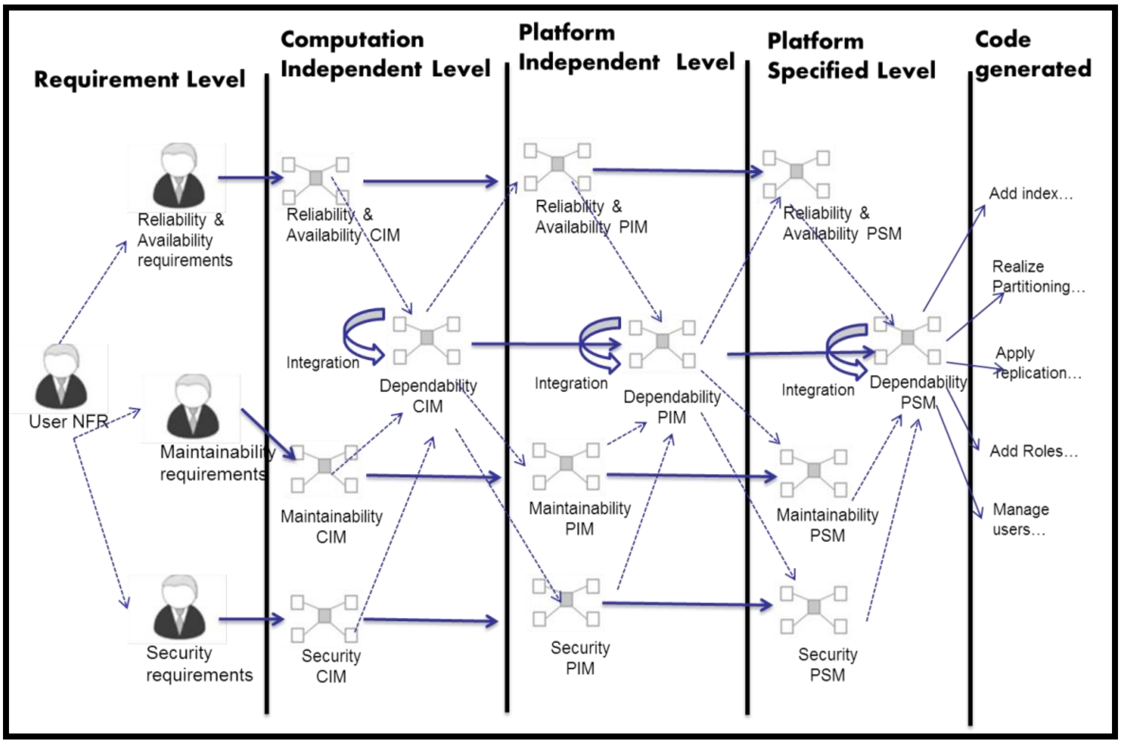

Fig. 3 Proposed approach to design a dependable DWS [3]

In a first stage we have proposed CIM and PIM models, for each aspect of dependability. CIM models were developed using NFR (Non Functional Requirement) Framework, while PIM were developed using QoS (Quality of Service) UML profile. Thus the two models represent UML profiles which are by definition consistent with MOF. Fig 4 and Fig 5 show an example of CIM and PIM of DWS's maintainability [3].

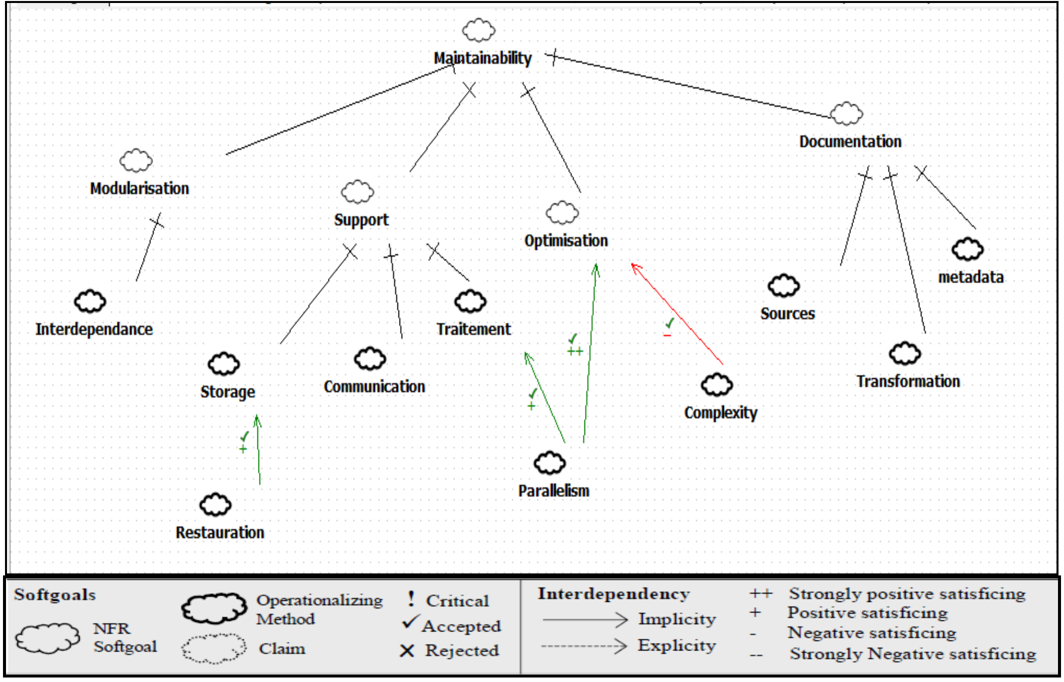

Fig. 4 CIM Maintainability [3]

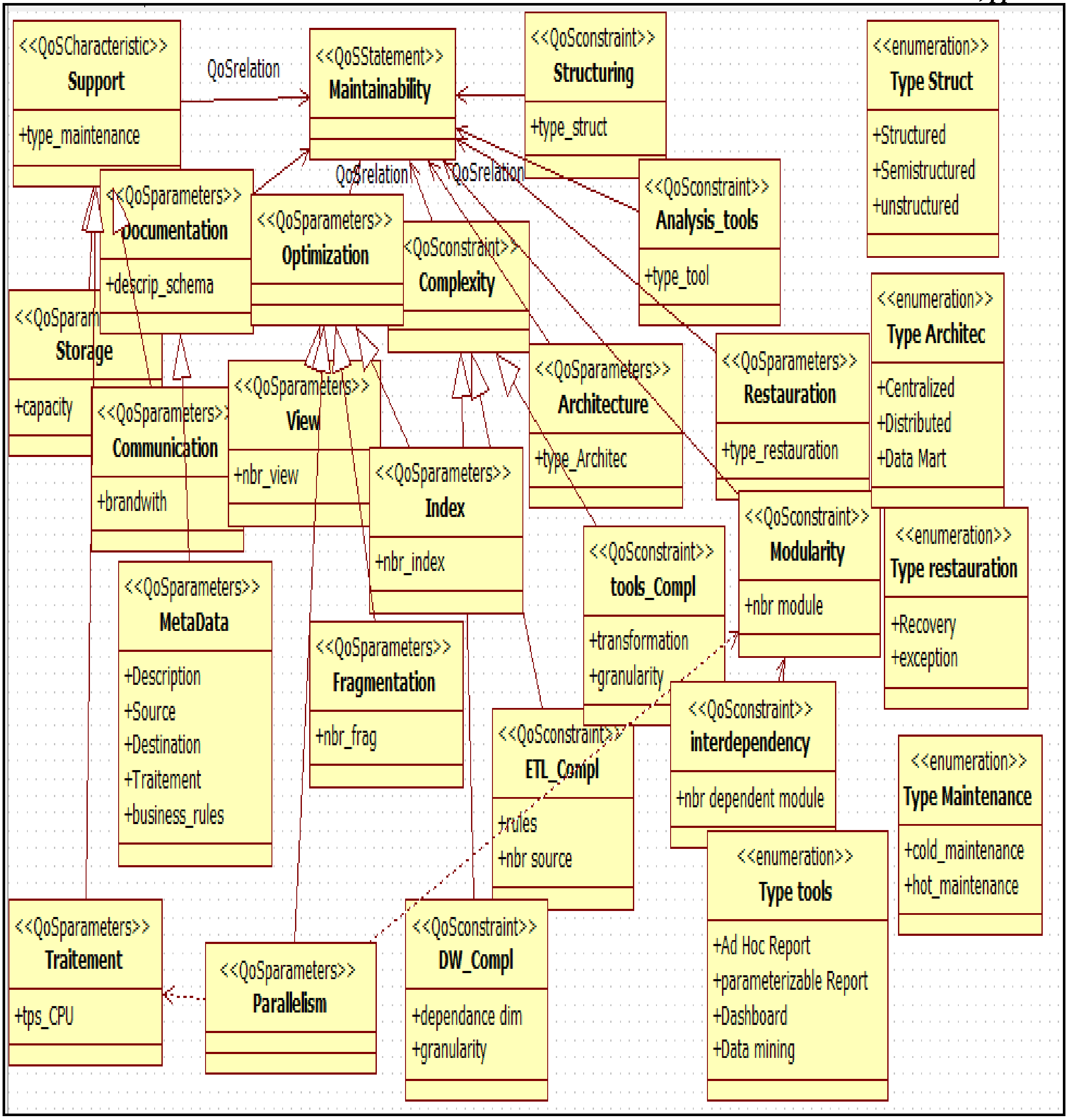

Fig. 5 PIM maintainability [3]

In the next sections, we will define the meta model source and destination then we will present the transformation rules as is it shown in Fig. 6.

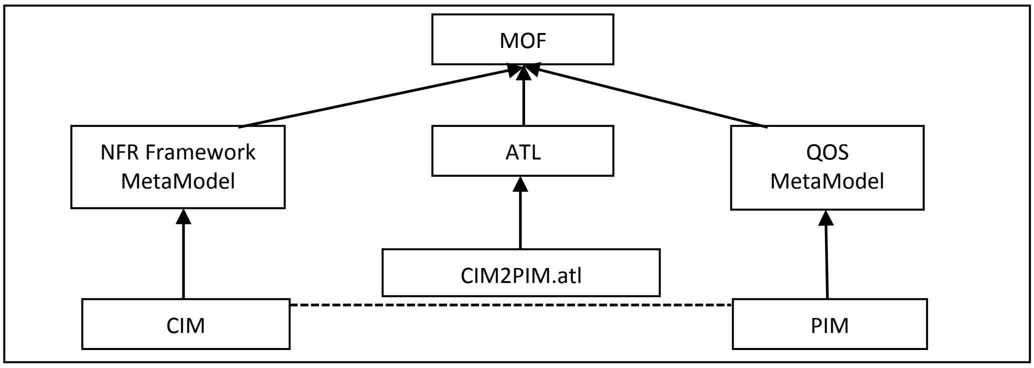

Fig. 6 Transformation mechanism of CIM to PIM

B. *Definition of meta models*

1) *Meta Model Source:* According to [14], the meta model NFR Framework is defined in Fig. 7:

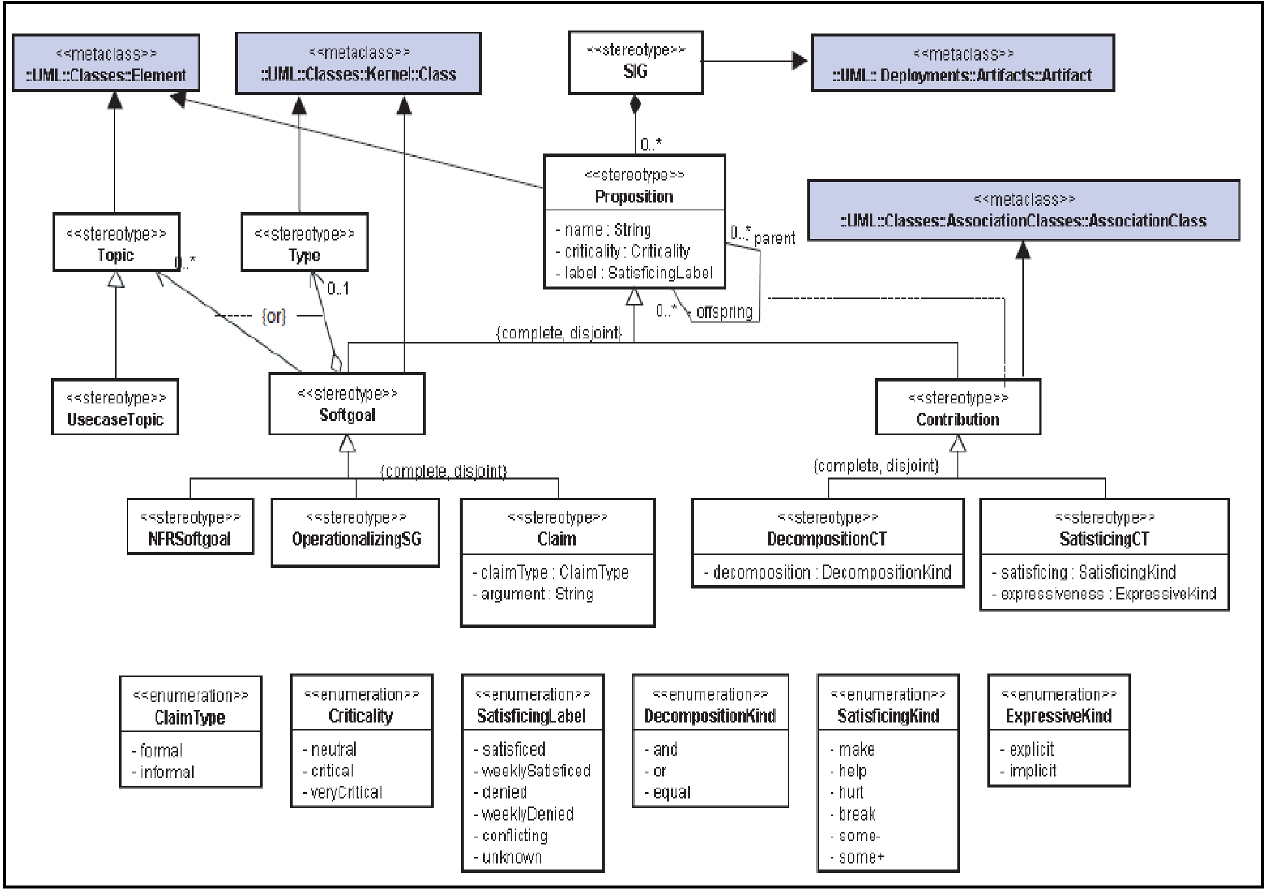

Fig. 7 Meta Model of the NFR Framework [14]

NFRSoftgoal Stereotypes and OperationalizingSG can represent in our context, an aspect of dependability, a parameter or a constraint of it. Therefore, to better represent this information in accordance with the NFR Framework and meet our needs, we propose to refine it.

This refinement of the meta-model NFR Framework is achieved by the addition of three stereotypes respectively Aspect, parameter and constraint. As it is showed in Fig. 8.

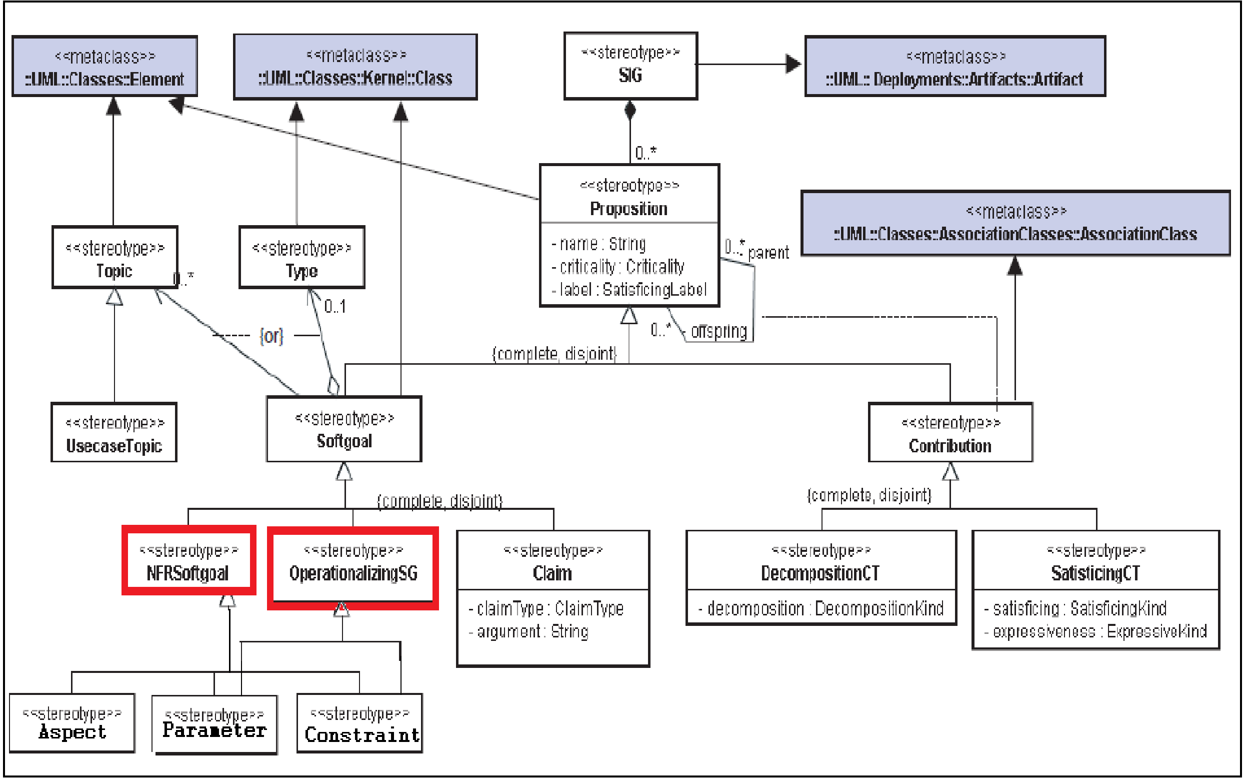

Fig. 8 Refinement of the NFR Framework

Figure 9 represents an extract from CIM of maintainability with the new proposed stereotypes.

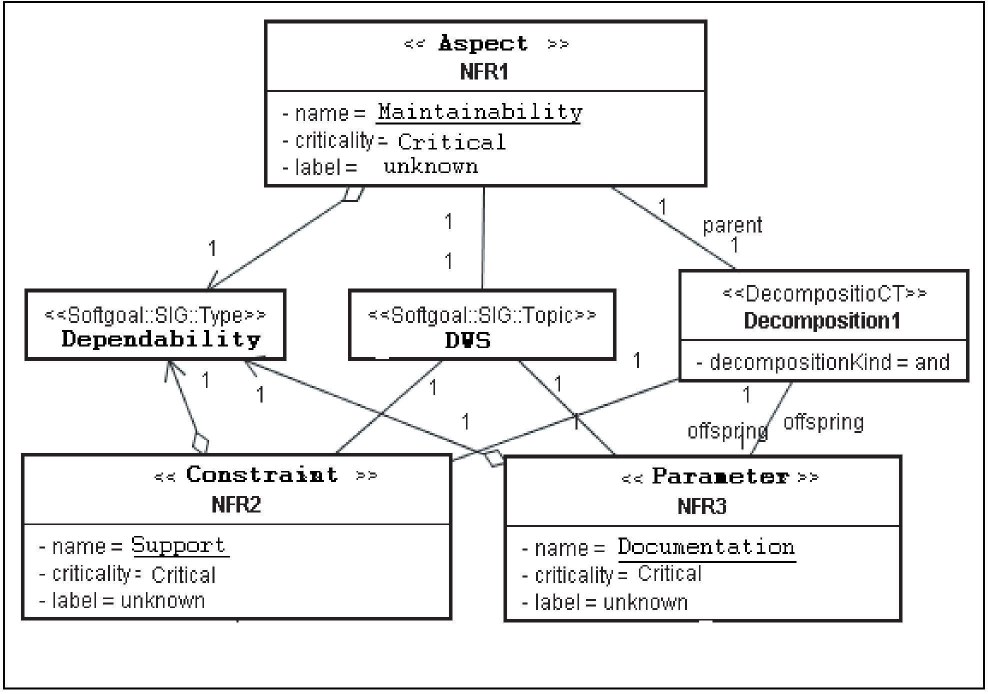

Fig. 9 Extract from the maintainability refined CIM

2) *Meta Model Destination:* The QoS profile and Fault-tolerance (FT) is an UML extension proposed by the OMG in order to correct problems related to the UML representation of QoS and FT properties; by integrating them as extra-functional aspects of UML models. The Fig.10 is an extract of the meta model of QoS profile defined by the OMG [14].

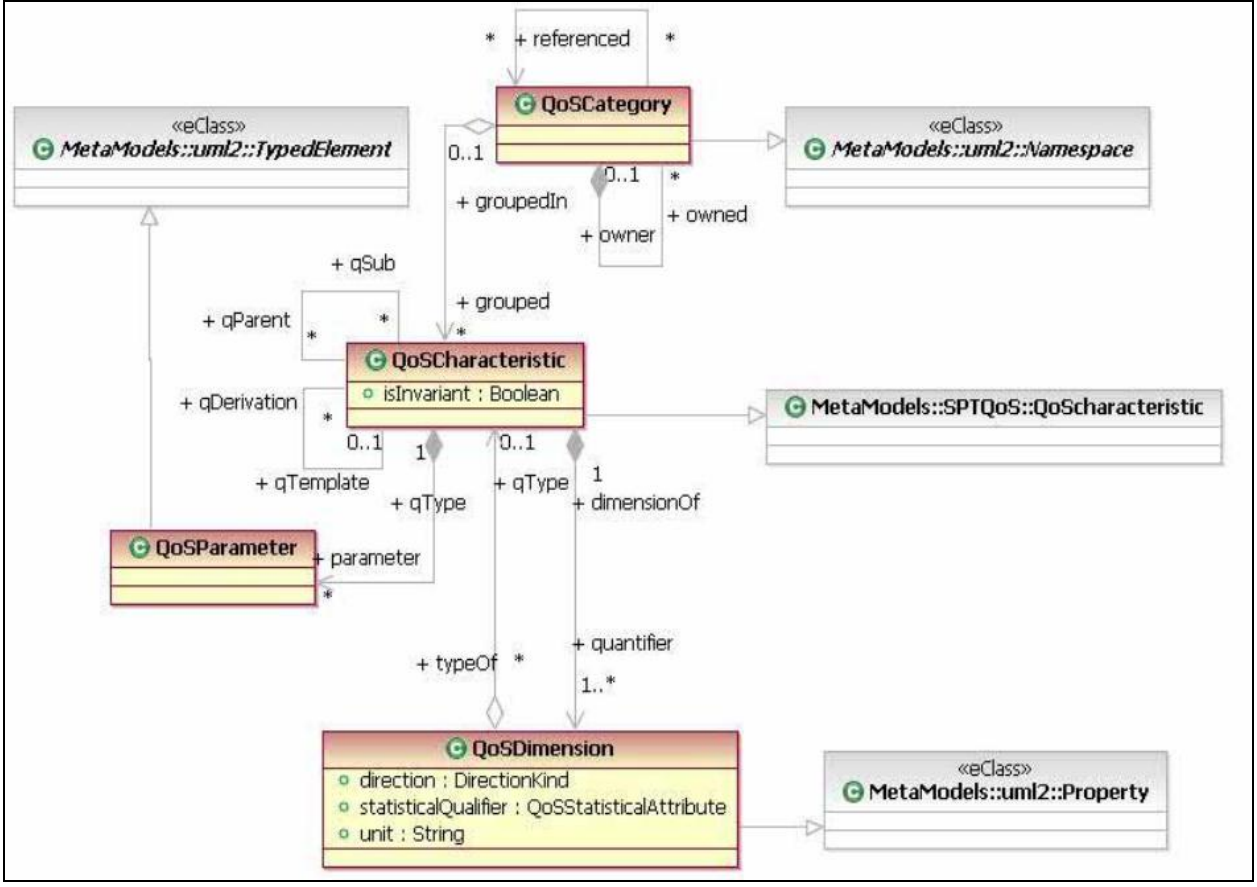

Fig. 10 Extract from the Meta model of QoS profile [13]

C. Transformations Definition

We will describe in this paragraph the ATL transformations that will allow to pass from a meta model NFR to a QoS meta-model. We used the syntax of the OCL according to ATL.

```
Module NFR2QoS
          Create out : QoS  from  in NFR

Rule SIGSoftgoal2class
          {
          from   S : NFR!SIGSoftgoal
            To       T : QoS ! Class
          If ( S.stereotype ←—— includes(« Aspect »)     )

          Then          T.stereotype ←— « QoSCharacteristic »
          Else
                    If ( S.stereotype ←— includes(« Constraint »)     )
                    Then T.stereotype ←—« QoSConstraint»
                    Else  T.stereotype ←——   « QoSParameter»
          }
Rule operationalizingSG2QoS
          {
          from   S : NFR! operationalizingSG2QoS
            To       T : QoS ! Class
                    If ( S.stereotype ←———   includes(« Constraint »)     )
                    Then  T.stereotype ←————— « QoSConstraint»
                    If ( S.stereotype ←———  includes(« Parameter »)     )
                    Then    T.stereotype ←————— « QoSParameter»
          }
}
```

V. Conclusions and Perspectives

Our work was to define transformation rules for passing from CIM to PIM through the ATL transformation language. Our goal is to align the MDA approach to develop a dependable Data Warehouse System. This is why, we have defined the two meta model of CIM and PIM then proposed transformation rules that ensure automatic switching between them.

As prospects, we intend to apply the same approach to defining the transformation rules of Platform Independent Model to Platform Specific Model (PSM) by refining the PIM. The objective is to facilitate the integration of aspects of dependability in Data Warehouse Systems at the beginning of their life cycle.

References

[1] W. Inmon, Building the Data Warehouse, 3rd edition, Wiley, 2002.J. Breckling, Ed., The Analysis of Directional Time Series: Applications to Wind Speed and Direction, ser. Lecture Notes in Statistics. Berlin, Germany: Springer, 1989, vol. 61.

[2] R.Kimball, L.Reeves, M.Ross, and W.Thornthwaite, “The Data Warehouse Lifecycle Toolkit”, Wiley, 2004

[3] I.Hilal, N.Afifi, H. R.Filali, M.Ouzzif, "Toward A New Approach For Modeling Dependability Of Data Warehouse System" International Journal of Computer Science and Information Security (IJCSIS) Published JUNE 2013 Volume 11 No. 6 (pp. 47-54)

[4] J. N. Mazon, J.Trujillo, M.Serrano, M.Piattini, (2005, November). Applying MDA to the development of data warehouses. In Proc / * eedings of the 8th ACM international workshop on Data warehousing and OLAP (pp. 57-66). ACM.

[5] J. N. Mazon, J.Trujillo, J. Lechtenbörger, (2006). A set of QVT Relations to ensure the correctness of data warehouses by using multidimensional normal forms. In Conceptual Modeling-ER 2006 (pp. 385-398). Springer Berlin Heidelberg.N. Subramanian

[6] C. Blanco, I. Garcia-Rodriguez de Guzman, E. Fernandez-Medina, J. & Trujillo (2010). Defining and transformingsecurityrules in an MDA approach for DWs. International Journal of Business Intelligence and Data Mining, 5 (2), 116-133

[7] E. Soler, V. Stefanov, J. N. Mazon, J. Trujillo, E. Fernández-Medina, M. Piattini, (2008, March). Towards comprehensive requirement analysis for data warehouses: Considering security requirements. In: The Third International Conference on Availability, Reliability and Security ARES 2008, Barcelona, Spain, p. 104-111. IEEE Computer Society, Los Alamitos (2008)

[8] The OMG. "MDA Guide Version 1.0.1" Available At http://www.enterprisearchitecture.info/Images/MDA/MDA% 20Guide% 20v1-0-1.pdf, 2003.

[9] D. S. Frankel (2003). Model Driven Architecture Applying Mda. John Wiley & Sons.

[10] Diaw, S. Lbath, R., & Coulette, B. (2010). State of the art of software development based on model transformations. Technology and Computer Science, 29 (4-5), 505-536.

[11] L. Chung (2001, September). Software architecture adaptability: an NFR approach. In Proceedings of the 4th International Workshop on Principles of Software Evolution (pp. 52-61). ACM.

[12] L. Chung, J. C. S. do Prado Leite (2009). On non-functional requirements in software engineering. In Conceptual modeling: Foundations and applications (pp. 363-379). Springer Berlin Heidelberg.

[13] J. Ø. Aagedal, & E. F. Ecklund Jr. (2002). ModellingQoS: towardsa UML profile. In "UML" 2002-The UnifiedModelingLanguage (pp. 275-289). Springer Berlin Heidelberg.

[14] S. Supakkul, L. Chung (2005, August). A UML profile for goal-oriented and use case-drivenrepresentation of NFRS and FRs. In Software Engineering Research, Management and Applications, 2005. Third ACIS International Conference on (pp. 112-119). IEEE.

[15] E. Soler, R. Villarroel, J. Trujillo, E. Fernandez-Medina, & M. Piattini (2006, April). Representing security and audit rules for data warehouses at the logical level by using the common warehouse metamodel. In Availability, Reliability and Security, 2006. ARES 2006. The First International Conference on (pp. 8-pp). IEEE.