American Journal of Embedded Systems and Applications
2019; 7(1): 1-8
http://www.sciencepublishinggroup.com/j/ajesa
doi: 10.11648/j.ajesa.20190701.11
ISSN: 2376-6069 (Print); ISSN: 2376-6085 (Online)

A Meta-model for Diverse Data Sources in Business Intelligence

Fatima Kalna, Abdessamad Belangour

Faculty of Science Ben M'Sik, University Hassan II, Casablanca, Morocco

Email address:
Fz.kalna@gmail.com (F. Kalna), belangour@gmail.com (A. Belangour)

To cite this article:
Fatima Kalna, Abdessamad Belangour. A Meta-model for Diverse Data Sources in Business Intelligence. *American Journal of Embedded Systems and Applications*. Vol. 7, No. 1, 2019, pp. 1-8. doi: 10.11648/j.ajesa.20190701.11

Received: January 18, 2019; **Accepted**: February 28, 2019; **Published**: March 20, 2019

Abstract: With the manifestation and evolution of Internet, several new types of data have emerged (videos, images, audio files, documents…). These new types of data classed as unstructured are more and more used and exchanged between IT systems, therefore their exploitation in Business Intelligence (BI) systems will absolutely provide a gold mine of information that guarantee a better and rich decision-making. Unfortunately BI systems don't consider this sort of data and they are still limited to classical data sources: structured as Relational data source and semi-structured as XML files. Many research works separate the treatment and the design of a data warehouse that involves heterogeneous sources in order to avoid any problems of data integration and storage. However, the need for an approach that gathers diverse data sources still present. In this paper we appeal Model Driven Engineering (MDE) to propose a meta-model that assemble and describe all sort of structured, semi-structured and unstructured data sources such as relational, multidimensional, XML and NoSQL databases. Models conforming this meta-model will serve as an input for our BI process and for designing and modeling a data warehouse.

Keywords: Business Intelligence, Data Source, Meta-Model, Relational, Multidimensional, NoSQL

1. Introduction

The BI process is a sequence of steps that leads to the preparation and the storage of data intended to decision-making. This process consists of three parts [1]: The System Of Records (SOR), the System Of Integration (SOI) and the third one is the System Of Analysis (SOA). The system of records (SOR) contains the data sources that have to be included in the design of the data warehouse. Usually, the data derived from these data sources are structured or semi-structured. But unfortunately, the larger part of data scattered throughout the Web, which will certainly have a huge impact on decision-making, are not considered in the classical decisional paradigm [2] especially traditional databases that become inadequate to support applications with unstructured data [18]. The system of integration (SOI) ensures the integration of data from its different sources. Its first goal is to establish the connection with the system of records (SOR), the second is to extract the data from their sources, and the third one is to carry out the sequence operations of transforming, formatting and cleaning data to avoid any kind of ambiguity and inconsistency that can come from the diversity and the heterogeneity of data types [3]. Finally, the main purpose is to load the aforementioned data into a data warehouse or data marts according to the architecture chosen beforehand [4-5]. The system of analysis (SOA) provides the components to be used and queried by applications and decision support tools. In this paper, we are interested in the system of records (SOR) and the structure of its data sources, which can be structured, semi-structured or unstructured. Thus, our goal is to take advantage of this diversity provided by Web applications and the traditional data sources to design an unified, integrated and valid data warehouse [5] that enables decision makers and managers to decide gradually with ease based on multiple and varied sources. For this purpose, we describe in this paper, using a meta-model based on Model Driven Engineering (MDE) [21], the structure of unstructured data that are managed by NoSQL databases (key-value, column-oriented [19], document-oriented and graph-oriented [20]), as well as the semi-structured data structure that are presented by the XML files, and the structure of the structured data stored and manipulated by the relational and multidimensional databases. Our work is organized as follows: In the first section, we present a meta-model de-

scribing the four types of NoSQL database (column-oriented, key-value, graph-oriented, document-oriented). The second is devoted to the meta-model that exposes the components of multidimensional databases. The third is dedicated to the meta-model of relational data source. In the fourth section, we introduce the XML meta-model, and we present in the fifth section a generic meta-model that unites the previous meta-models. Finally, we conclude our paper and we present our prospects for our future work.

2. NOSQL Databases

NoSQL databases have emerged to deal with the problems of storing and managing large quantities of data produced by objects connected to Internet and which become unmanageable by the traditional relational databases management systems (As proclaimed by the big players of Web: Yahoo, Amazon, Google, Facebook and Twitter) [6]. To solve these problems, the NoSQL databases propose four figures to implement its solutions:

(1) Column oriented
(2) Key-Value
(3) Document oriented
(4) Graph oriented

2.1. Column Oriented Databases

A column-oriented database is a collection of data structured by rows, where the number of columns differs from one row to another instead of relational database where the number of columns is fixed. A column-oriented database consists of column families that resemble to the concept of a table in a relational database [7]. In fact, each column family include a set of rows each with a unique identifier, and each row receive a variable number of columns structured as "key/value".

To describe the structure of columns-oriented database, we propose the meta-model below:

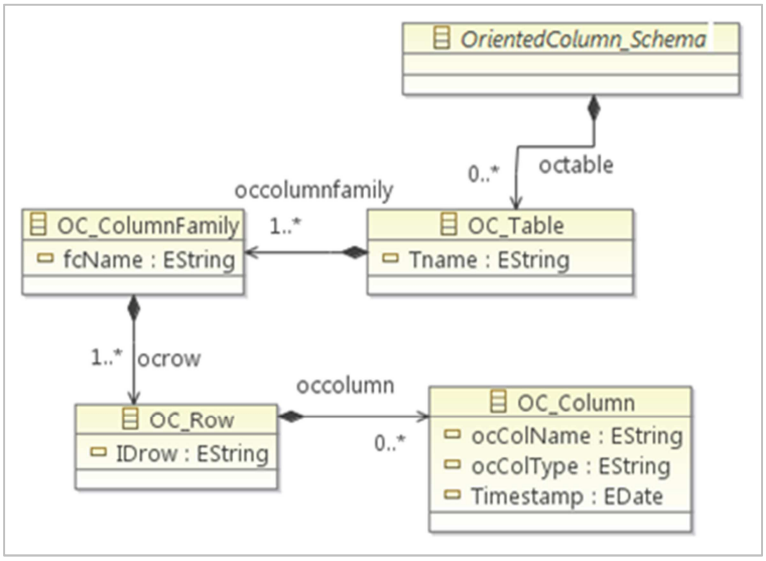

Figure 1. *Column Oriented database meta-model.*

The table below describes the main components of column-oriented database and their type:

Table 1. *Structure of Column Oriented databases.*

Entity	Type	Description
OC_Table	Meta-Class	-Equivalent of a package -Includes related objects
OC_Column Family	Meta-Class	-Equivalent of a table in RDB -Linked to OC_Table by a composition association «occolumnfamily » -An OC_Table can contain one to several OC_ColumnFamily -Having a unique identifier « IDrow »
OC_Row	Meta-Class	-Linked to OC_ColumnFamily composition association « ocrow » -An OC_ColumnFamily can contain one to several OC_Row -Structured on doublet (key ; value)
OC_Column	Meta-Class	-Linked to OC_Row by a composition association « ocrow » - An OC_Row can contain one to several OC_Column

2.2. Key-Value Databases

Data within a key-value database is stored and managed by the doublet key-value, where each key has its own value [7]. To describe the structure of a key/value database, we propose the meta-model below:

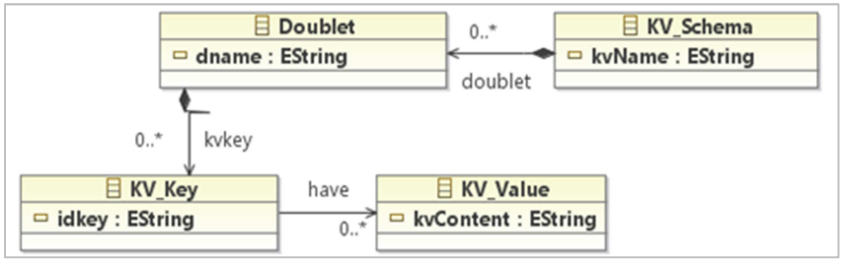

Figure 2. *Key-Value database meta-model.*

In the table below we describes the components of key-value database:

Table 2. *Structure of Key-Value databases.*

Entity	Type	Description
KV_Schema	Meta-Class	-Equivalent of a package
Doublet	Meta-Class	-Consists of the double Key/Value linked to KV_Schema by a composition association
KV_Key	Meta-Class	-Unique identifiant
KV_Value	Meta-Class	-Each key is linked to a value (association) -Each value is composed of a Key / Value doublet

2.3. Document Oriented Databases

A document-oriented database is a collection of documents organized by hierarchy where each document is a doublet of key/value. In a document, the value can be a set of documents. This type of database is more adapted to the Web [6]. The following meta-model presents the structure of a document-oriented database:

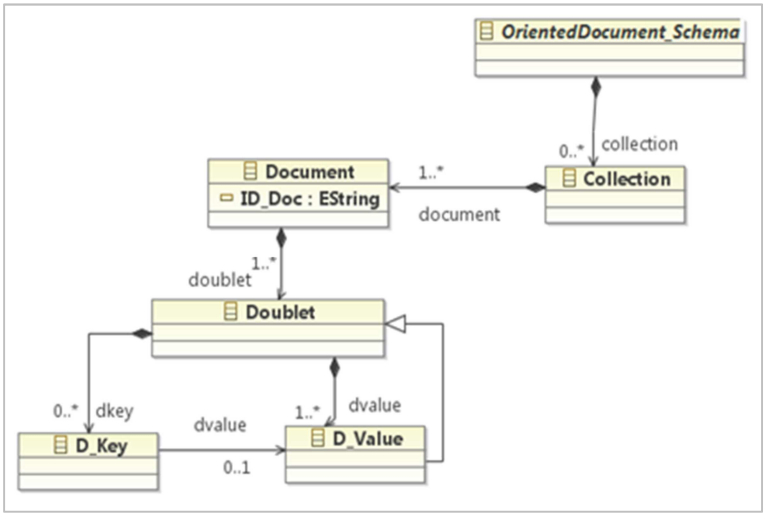

Figure 3. *Document Oriented database meta-model.*

An overview of the structure of document-oriented databases is also presented in the table below:

Table 3. *Structure of Document Oriented databases.*

Entity	Type	Description
Collection	Meta-Class	- Consists of a set of documents - A collection has a name
Document	Meta-Class	- Basic unit of a document-oriented structure - A document has a name - A document has a hierarchical structure - A document is composed of the doublet Key/Value - A document has an identifier
KV_Key	Meta-Class	- Label of a field - A key has a name
KV_Value	Meta-Class	- Each key has a value - A value can be nested to one or more documents

2.4. Graph Oriented Databases

Graph-oriented databases are the most widely used for network design (e.g. social networks) [8]. They consist of nodes and relationships. Each node has a property described by the doublet key/value. A node source is linked to a node destination by an information-bearing connection in the form of the doublet key/value as well. To describe the structure of a graph-oriented database, we propose the meta-model below:

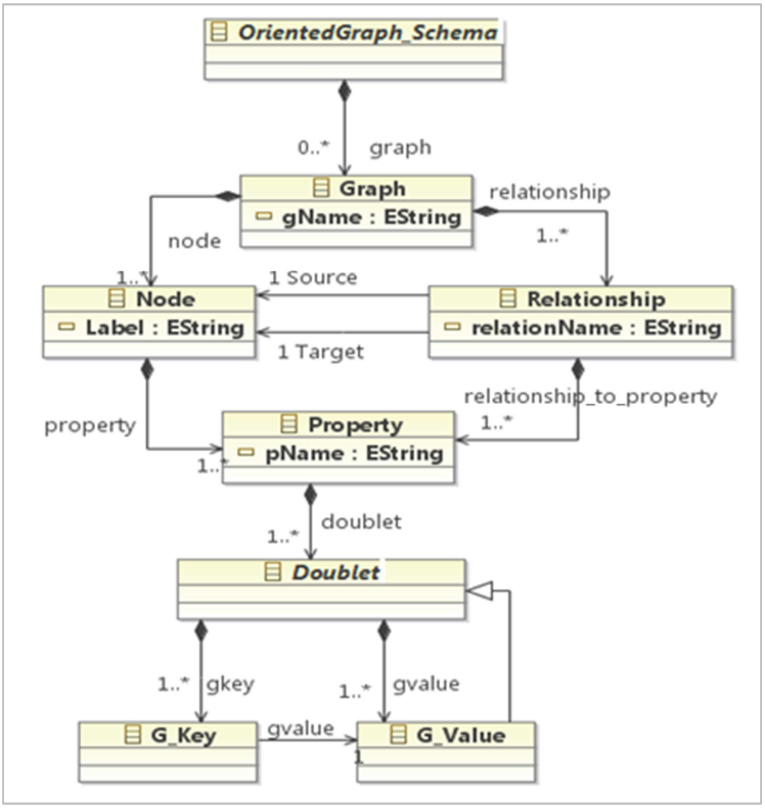

Figure 4. *Graph Oriented database meta-model.*

The structure of graph-oriented database is detailed in the table below:

Table 4. *Structure of Graph Oriented databases.*

Entity	Type	Description
Graph	Meta-Class	-Equivalent of a package -A graph has a name -A graph consists of two to several nodes -A graph consists of one to several relations
Node	Meta-Class	- A node has a name - A node can be a source or a destination or both at the same time -A source node is linked to a destination node by a relation - A source node can be linked to multiple destination nodes -A destination node has one and only one source node - A node has a property
Relationship	Meta-Class	- A relationship has a name - A relationship has a property
Property	Meta-Class	A property consists of the doublet key / value
G-Key	Meta-Class	- Label of filed - A key has a name
G-Value	Meta-Class	- Each key has a value - A value can be nested to one or more doublet (Key/Value)

3. Multidimensional Databases

The purpose of multidimensional databases is to support data analysis (OLAP) addressed to decision support, unlike relational databases that are dedicated to daily transactions (OLTP) of an entity [16]. BI systems aim to extract and transform data from

relational and other databases into data that can be presented and analyzed by decision-makers [14]. BI systems rely on data warehouses and data marts to store data under a dimensional schema (star, flake or constellation) that consists of a fact (or set of facts) which is the subject to be analyzed, this fact is associated with shared or specific dimensions that represent the axis of analysis [4]. We follow the meta-model proposed by Faten Atigui [9] in the following figure:

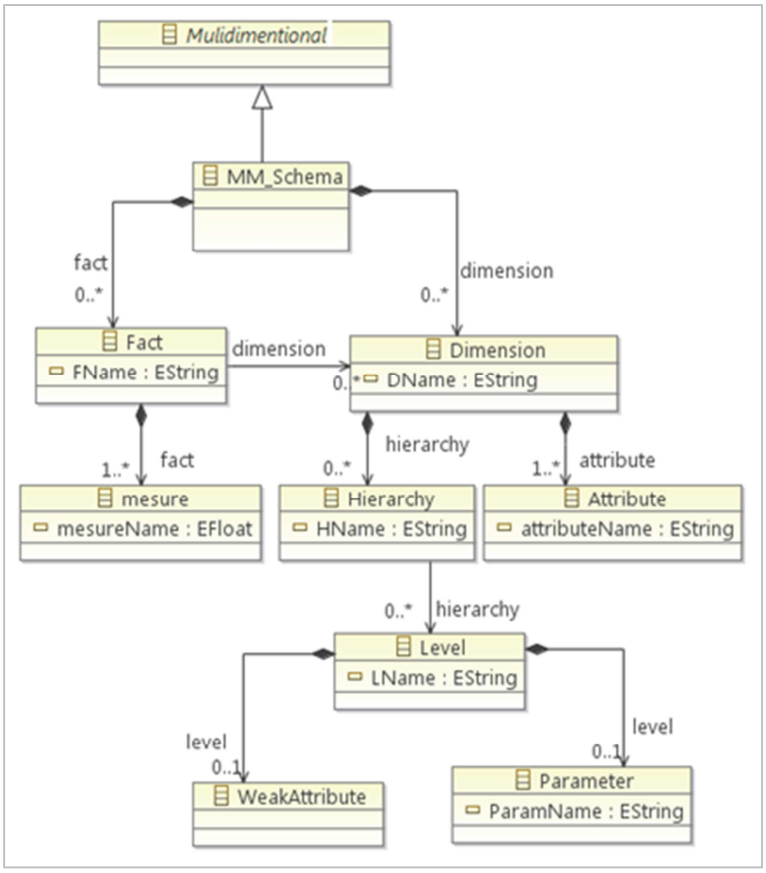

Figure 5. *Multidimensional database meta-model.*

For more details, we can follow the table below about the components of multidimensional database:

Table 5. *Structure of multidimensional databases.*

Entity	Type	Description
Multidimensional_ Schema	Meta-Class	-Non-empty set of facts -Non-empty set of dimensions -Associates each fact with a set of dimensions [17]
Fact	Meta-Class	-Represents the subject to be analyzed -Star and flake schema contains a single fact -Schema in constellation contains several -A fact has a name -A fact contains measures
Mesure	Meta-Class	-A measure has a name -A measure is a numeric -A measure is submitted to aggregation functions
Dimension	Meta-Class	-A dimension represents an axis of analysis -A dimension has a name -A dimension contains attributes -A dimension contains a set of hierarchies
Hierarchy	Meta-Class	-A hierarchy has a name -A hierarchy consists of levels (Parameter)
Parameter	Meta-Class	- A level has a name - A parameter represents the level of the hierarchy
WeakAttribute	Meta-Class	-A weakAttribute has a name -A weak attribute completes the semantics of the parameters -Eventually an empty set

4. Relational Databases

Relational databases are designed to respond to the transactional uses of an entity. In a relational schema, the data is represented by values in columns and rows of tables (Rule1, Codd) [10], these tables are linked by relations and managed by foreign keys.

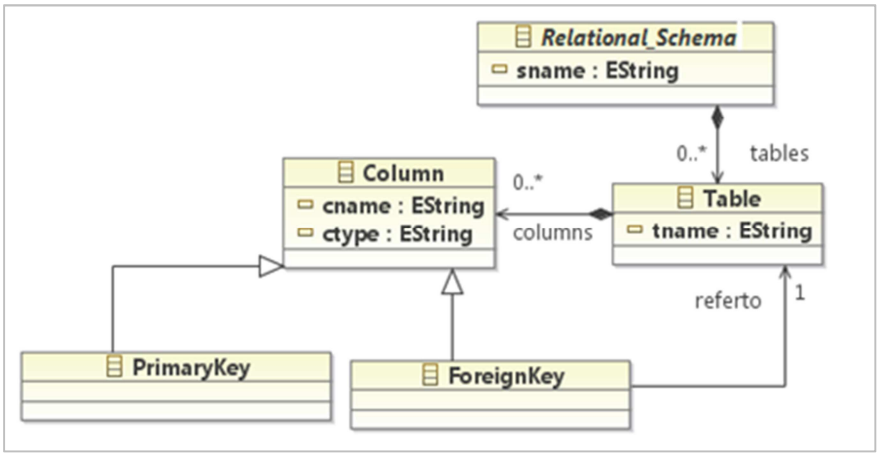

Figure 6. *Relational database meta-model.*

Each row is identified by a primary key and represents an object that includes all information about columns, these values are from various types. Figure 6 above shows a meta-model of a relational schema and its components.

5. XML Files

XML is one of the most formats used and exchanged on Internet, its markup design allows structuring and representing information in the form of a tree. On the other hand, XML is the standard format for the exchange and the description of data (metadata for example, XMI), Therefore, XML data documents can appear everywhere [11]. An XML schema consists of the document's root element, which contains a set of structured elements as attributes/contents. The meta-model below proposed by the OMG [11], describes an XML document:

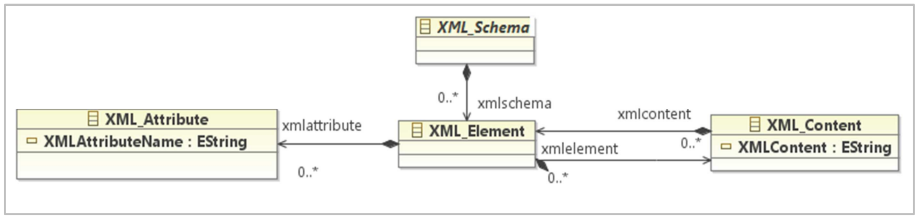

Figure 7. *Meta-model of XML Schema.*

6. Generic Meta-model

The data sources are diverse and multiple, the description of their schemas will absolutely facilitate the steps that follow for designing a data warehouse with varied databases. To describe these data sources, we propose in Figure 8 a meta-model implanted and validated using the Eclipse Modeling Framework (EMF) platform [15]. This meta-model is generic because it presents and encompasses five meta-models of different data sources: The NoSQL database meta-model presented in Section 1, the multidimensional database meta-model described in Section 2, the relational database meta-model cited in Section 3, and the XML meta-model shown in Section 4. In fact, this generic meta-model provides an overview of the structure of the system of record (SOR) that appeal different data sources with different data schemas.

In MDA (Model Driven Architecture) approach, the proposed generic meta-model presents the PIM Conceptual (Platform Independent Model) which describes the structures of our data sources independently from any specific platform [12].

Our future data warehouse is going to take this schema as input and following a number of transformation based on ATL language [13] the result will be a single schema describing our data warehouse. We will choose as final schema of data warehouse one of the three models of Nosql databases, this choice gives us more advantages to our source ready to feed data marts. By respecting this process we will benefit from the variety and the volume of data stored and from the velocity of treatment of these data, thus will provide a rich, refined and rewarding analysis results that will help in decision-making.

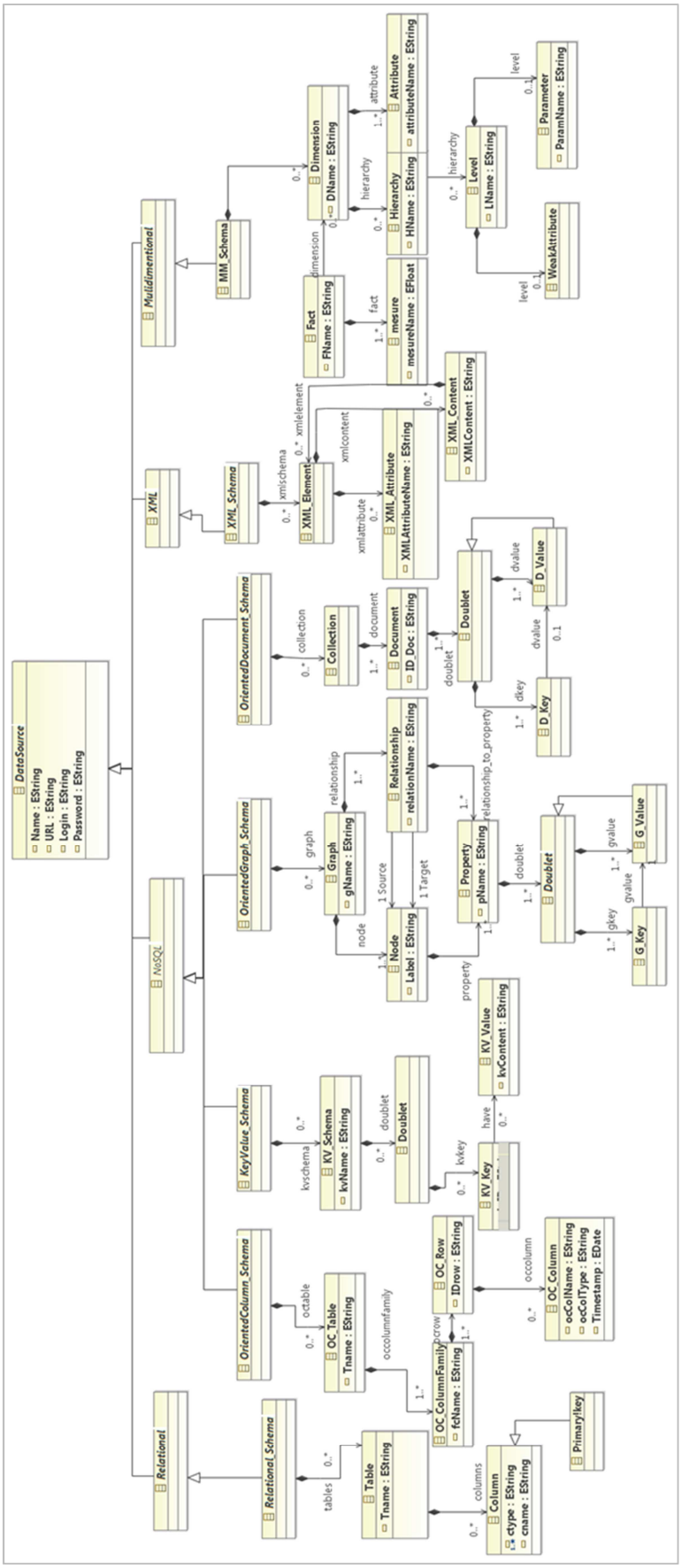

Figure 8. *Generic Meta-model of data sources.*

7. Conclusion

Our work is the first step to design a data warehouse based on a variety of data type. Our goal is to describe all data sources that can be taken to feed a data warehouse. Thanks to the generic meta-model proposed in this paper we have been able to describe all available databases schemas that can represent the system of records (SOR) of a decision-making system. In our next work, this meta-model is going to be submitted to a sequence of transformation using Model Driven Engineering (MDE) approach and the ATL language to solve the problems related to the data quality (formatting, redundancy, Nullability ...) and the heterogeneity of the data sources. This is going to come in handy in the development of several meta-models describing different levels of abstraction.

References

[1] R. Sherman, Business Intelligence Guidebook From Data Integration to Analytics, Elsevier, 2014.

[2] M. Chevalier, M. El Malki, O. Kopliku, R. Teste, Tournier, Multidimensional Data Warehouses NoSQL, EDA, 2015, pp.161-176.

[3] T. Etienne, Establishment of a data warehouse for the strategic decision, Academia, 2008, pp 2.

[4] R. Kimball, The Data Warehouse Toolkit: Practical Techniques for Building Dimensional Data Warehouse, John Wiley, 1996.

[5] B. Inmon, Building the Data Warehouse, John Wiley and Sons, New York, 1996.

[6] W. Brand, NoSQL For Dummies, John Wiley & Sons, 2015.

[7] F. Abdelhadi, A. Ait Brahim, F. Atigui, G. Zurfluh, Logical Unified Modeling For NOSQL Databases, ICEIS, 2017.

[8] J. Han, E. Haihong, G. Le, J. Du, Survey on NoSQL Database, Pervasive Computing and Applications, ICPCA 6th International Conference, 2011.

[9] F. Atigui, Model-Driven Approach for Implementing and Reducing Data, 2013, pp 6-133.

[10] R. Bruchez, Les bases de données NoSQL et le Big Data, Eyrolles 2nd edition 2015.

[11] Object Management Group OMG, Common Warehouse Meta-model (CWM) Specification, Version1.1, March 2003.

[12] X. Blanc, MDA in action: software engineering guided by models, Eyrolles, 2005.

[13] F. Allilaire, T. Idrissi, Eclipse development tools for atl, http://www.sciences.univnantes.fr/lina/atl/Members/allilaire/Paper/ADT% AllilaireIdriss, 2004.

[14] P. Vassiliadis, A Survey of Extract-Transform-Load Technology, International Journal of Data Warehousing and Mining IJDWM, 2009.

[15] Object Management Group OMG, Meta Object Facility (MOF) Core Specification, Version 2.4.1, August 2011.

[16] C. Favre, F. Bentayeb, O. Boussaid, J. Darmont, G. Gavin, N. Harbi, N. Kabachi, S. Loudcher, Data warehouses for dummies. . . or not !, 2nd Workshop helps the Decision at all Floors (EGC / AIDE) , January 2013.

[17] F. Ravat, O. Teste, R. Tournier, G. Zuruh, Algebraic and graphic languages for olap manipulations, International Journal of Data Warehousing and Mining IJDWM, 2008, pp.17-46.

[18] A. Abello, Big data design, DOLAP, 2015.

[19] K. Dehbouh, F. Bentayed, O. Boussaid, N. Kabachi, Using the column oriented model for implemeting big data warehouses, PDPTA, 2015.

[20] X3 G. Daniel, G. Sunyé, J. Cabot, UMLtoGraphDB: Mapping conceptuel schemas to graph databases, ER 2016 - 35th International Conference on Conceptual Modeling, Gifu, Japan, November 2016.

[21] F. Abdelhedi, A. Ait Brahim, F. Atigui, G. Zurfluh, MDA-based approach for NoSQL Databases modelling, International Conference on Big Data Analytics and Knowledge Discovery (DaWaK 2017), France, 2017.