International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-1, October, 2019

MDA Transformation Process of A PIM Logical Decision-Making from NoSQL Database to Big Data NoSQL PSM

Fatima Kalna, Abdessamad Belangour, Mouad Banane, Allae Erraissi

Abstract*: Business Intelligence or *Decision Support System (DSS) is IT for decision-makers and business leaders. It describes the means, the tools and the methods that make it possible to collect, consolidate, model and restore the data, material or immaterial, of a company to offer a decision aid and to allow a decision-maker to have an overview of the activity being treated. Given the large volume, variety, and data velocity we entered the era of Big Data. And since most of today's BI tools are designed to handle structured data. In our research project, we aim to consolidate a BI system for Big Data. In continuous efforts, this paper is a progress report of our first contribution that aims to apply the techniques of model engineering to propose a universal approach to deal with Big Data is to help decision-makers make decisions.

Keywords: Model Driven Engineering; meta-model; business Intelligence; Big Data.

I. INTRODUCTION

In recent years there has been an explosion of data generated and accumulated by increasingly numerous and diversified computing devices. Constituted databases are referred to as "Big Data" and are characterized by the so-called "3V" rule. This is due to the volume of data that can exceed several terabytes and the variety of these data that are described as complex. Besides, these data are often entered at very high frequency and must, therefore, be filtered and aggregated in real-time to avoid unnecessary saturation of the storage space [1]. However, because of the complexity of the database schemas to implement and the specificity of the current physical models associated with NoSQL DBMS, Big Data users are faced with the challenge of implementing a NoSQL database [2]. The term NoSQL refers to a type of database management system that goes beyond the relational systems associated with the SQL language by accepting more complex data structures. According to their physical models, the databases managed by these systems fall into four categories: columns, documents, graphs and key-value [3]. Each of them offering specific features. For example, in a document-oriented database like MongoDB [4], data is stored in tables whose lines can be nested. This organization of the data is coupled to operators that allow access to the nested data. The work we are presenting aims to assist a user in the implementation of a massive database on a NoSQL database management system. To do this, we propose a transformation process that ensures the transition from the NoSQL logic model to a NoSQL physical model ie. Existing NoSQL solutions such as HBase [5], Cassandra [6], MongoDB [4], and Neo4J [7]. We have at the entrance of a logical NoSQL model which describes structures and data independent of the systems. From this model, in our process, we apply a chain of transformations that translate it into a succession of models to arrive at a target NoSQL model that complies with existing solutions. The following section 2 describes the context of our study as well as our research problem aimed at transforming logical NoSQL data PIMs into specific NoSQL platforms. Section 3 presents our contribution to formalize with MDA the process of transformation of NoSQL logical schemas to specific NoSQL platforms (PSM). Section 5 describes the experimentation of the proposed process. Finally, we discuss the results obtained in section 6.

II. PROBLEM

From a set of data sources representing structured, semi-structured and unstructured data we need to present a set of transformations needed to prepare the data that will feed into our data warehouse which will be based under a NoSQL database. In our study, we use the ETL approach where we will perform extractions from relational databases, multidimensional, XML and NoSQL [8], each of these data sources has its data model, its own attached entities and their management rules, attributes, and name, mnemonic code, type, size, and format. As a result, the problem is to define the mechanisms to manage this diversity at the model and data level to obtain a unified model with its standardized data for all data sources and the design of the schema. The data warehouse and its power supply are trouble-free.

Revised Manuscript Received on October 30, 2019.
* Correspondence Author
Fatima Kalna*, Laboratory of Information Technology and Modeling, Hassan II University, Faculty of sciences Ben M'Sik. Casablanca, Morocco. Email: fz.kalna@gmail.com
Abdessamad Belangour, LTIM, Hassan II University. FSBM. Faculty of sciences Ben M'Sik. Casablanca, Morocco.
Mouad Banane, Laboratory of Information Technology and Modeling, Hassan II University, Faculty of sciences Ben M'Sik. Casablanca, Morocco. Email: mouadbanane@gmail.com
Allae Erraissi*, Laboratory of Information Technology and Modeling, Hassan II University, Faculty of sciences Ben M'Sik. Casablanca, Morocco. Email: Erraissi.allae@gmail.com

III. RELATED WORK

A Big Data database contains a variety of data, that is, data of non-standard types usually referred to as complex objects: text, graphics, documents, video sequences, and so on. To implement such databases, studies have focused on the conceptual modeling of complex objects and have shown that these models can be applied to Big Data. Other works have proposed processes for transforming a database schema into a NoSQL schema. Complex Data Modeling, Darmont et al. [9], has been the subject of much research; we will focus on three of them: Pedersen and Jensen [10] Tanasescu et al. [11], Midouni et al. [12] that we considered the most important work in this context. Tanasescu et al. [11] are to design a UML diagram to conceptually identify and represent complex data in preparation for the multi-dimensional modeling process. In the context of data warehouses, the work of Chevalier et al. [13] defined rules for translating a multidimensional star model into two NoSQL physical models, a column-oriented model, and a document-oriented model. The links between facts and dimensions have been translated as nesting. In Li [14], the mechanisms for implementing a relational database in the HBase system were studied. The proposed method is based on rules allowing the mapping of a relational schema into an HBase schema; the relationships between the tables (foreign keys) are translated by adding the families of columns containing references. Other works, Yan et al. [15], studied the transformation of a DCL into an HBase data schema with the MDA approach. The basic idea is to build meta-models corresponding to the UML class diagram and the HBase column-oriented data model and to propose transformation rules between the elements of the two built meta-models. These rules make it possible to transform a DCL directly into an implementation scheme specific to the HBase system. This state of the art shows that few previous studies have studied the correspondence of a conceptual model of complex data with a Big Data model. In the work closest to our problematic, Yan et al. [15], the proposed schema transformation rules are not compatible with other column-oriented NoSQL systems, such as Cassandra and BigTable, the proposed transformation model does not consider a logical level independent of any technical platform. The article by Chevalier et al. [13] is part of the context of data warehousing as it studies the rules for moving from a multidimensional schema to NoSQL physical schemas; two platforms were chosen: the HBase column-oriented system and the MongoDB document-oriented system. Although the starting point of the process (a multidimensional scheme) is at the conceptual level, this scheme does not have the same characteristics as a UML DCL; in particular, it includes only Fact and Dimensions classes and a unique type of link between these two classes. Li's article [14] deals with the transformation of a relational schema into an HBase-oriented schema. This work responds well to the concrete expectations of companies who, in the face of recent developments in computing, want to store their current databases in NoSQL systems. But the source of the transformation process, here a relational schema, does not present the semantic richness that can be expressed in a DCL (notable thanks to the different types of links between classes: aggregation, composition, inheritance,etc.). The works presented in Yan et al. [15] are intended to specify an MDA transformation process from a conceptual schema (DCL) to an HBase physical schema. This process does not propose an intermediate level (the logical level) that would make the outcome of the process independent of a particular system platform. On the other hand, the transformation of DCL links does not take into account data organization constraints that have been dictated by the requirements of our application context. The use of NoSQL is considered today as the most efficient solution for managing large volumes of data, either structured or semi-structured like the works [16,17] that propose the use of NoSQL systems for data management. Semantic Web [18,19]. We also based our work on the work done by Erraissi et al [20,21] who have already defined meta-models for NoSQL databases by applying techniques related to model engineering [23, 24].

IV. PROPOSED APPROACH

In previous work we have already defined our approach which aims to transform at first the data source schemas: relational, multidimensional, NoSQL and XML in one of the NoSQL model (value key, column-oriented, document-oriented, graph-oriented) to have a single logical schema of data, and in a second time to transform the data in order to make them conform to a common schema [25].

The following figure shows the levels of modeling as well as the transformations proposed in our approach:

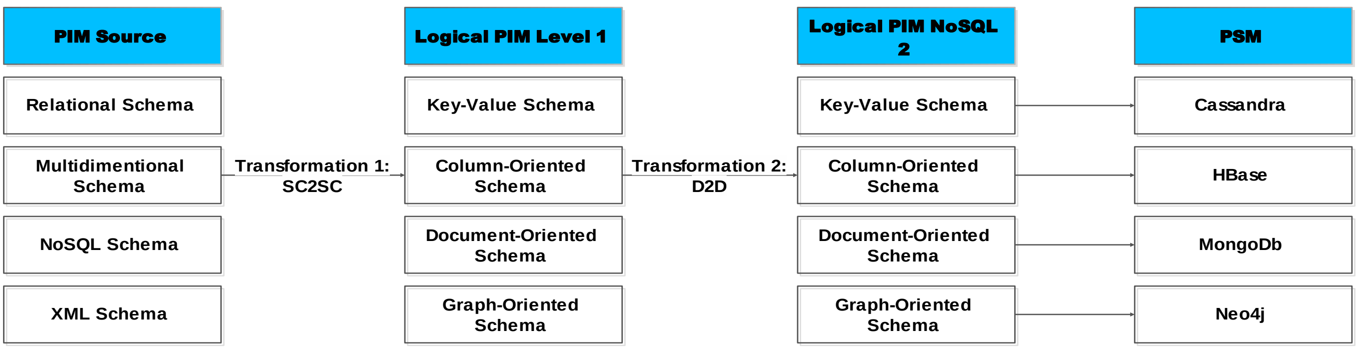

Fig. 1.modeling levels.

In our work, we have used the Model-Driven Approach (MDA), which offers meta-models and transformations that allow the transition from one level of abstraction to another [26]. First, a Source PIM in the proposed approach represents schemas of Big Data or other data sources. We have already applied an SC2SC Transformation allowing the transition from a suite of data source schema types to a NoSQL schema. So, we went to the second level that the Logic PIM 1, after that we applied another D2D Transformation to unify and standardize the Level 1 Logic PIM data and make them conform to a common schema. The work [8,25] presents the work done in this context. In this paper, we continue our efforts to transform meta-models of NoSQL databases to PSMs in the MDA. The following figure presents the meta-model of the Logical PIM layer NoSQL 2 taken from the work of Erraissi et al [20]:

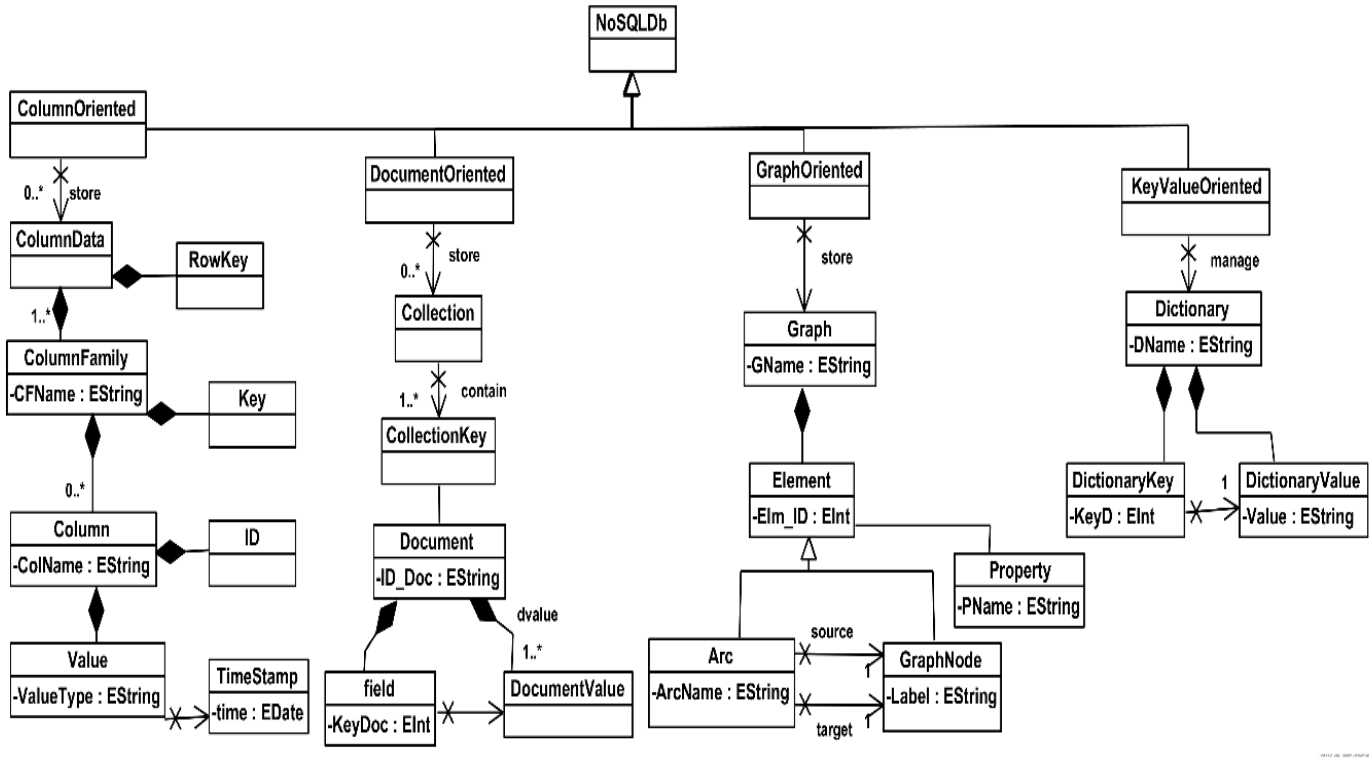

Fig. 2.Generic meta-model for Logical PIM NoSQL 2 [20].

The study of these correspondence principles is the basis of the work we are doing on ETL mechanisms dedicated to supplying a warehouse from Big Data sources. These mechanisms, taking into account the semantics of the data, require knowledge of the conceptual descriptions of the sources. The correspondence rules that we have just defined make it possible to obtain a logic diagram independent of any implementation platform. This principle ensures the independence of the logical level given the technical evolutions of the underlying NoSQL systems. We present briefly four implementation platforms: Cassandra, HBase, Neo4j, and MongoDB.

A. HBase

HBase is a column-oriented NoSQL system that was developed on top of the Hadoop Distributed File System (Hadoop) file system on the Hadoop platform, Vora [27]. By default, an HBase database consists of a single table that is denoted HTable (the administrator can modify this parameter to create multiple tables). When creating an HTable, we can associate a fixed number of families of columns; only the name of the family is specified without mentioning the column names. A family is a logical grouping of columns that will be added when the data is inserted. Each line (or record) within an HTable is identified by a key marked RowKey and chosen by the user. The triplet (RowKey, family of columns, column) corresponds to a single cell that will contain a value. The following figure shows the proposed meta-model for HBase:

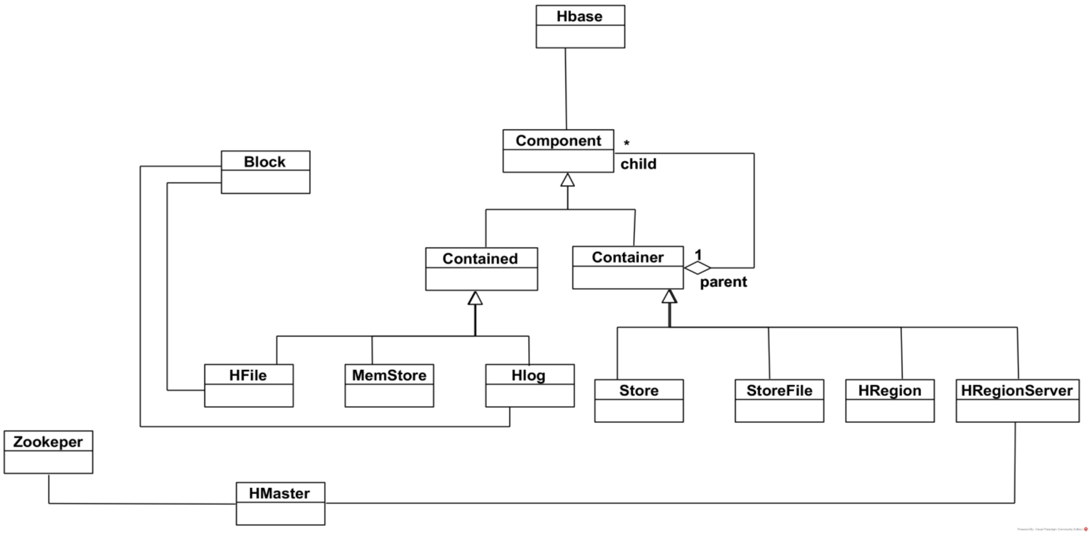

Fig. 3.Meta-model of HBase.

B. Cassandra

Cassandra is a column-oriented NoSQL DBMS, initially based on Google's BigTable model, but which also borrows features from Amazon's Dynamo 5 system. A Cassandra database is by default composed of a single data container noted KeySpace. The latter is associated with one or more families of columns, each of which is a logical grouping of lines. A row is composed of a set of columns and is identified by a key labeled PrimaryKey. Each column is represented by a triplet corresponding to a name, a type, and a timestamp. Note that the "Table" and "key-line" concepts will be replaced respectively by the HTable and RowKey concepts under HBase and by KeySpace and PrimaryKey under Cassandra.

The following figure shows the meta-model that we proposed for the Cassandra NoSQL database, we just focused on its part that manages key-value records.

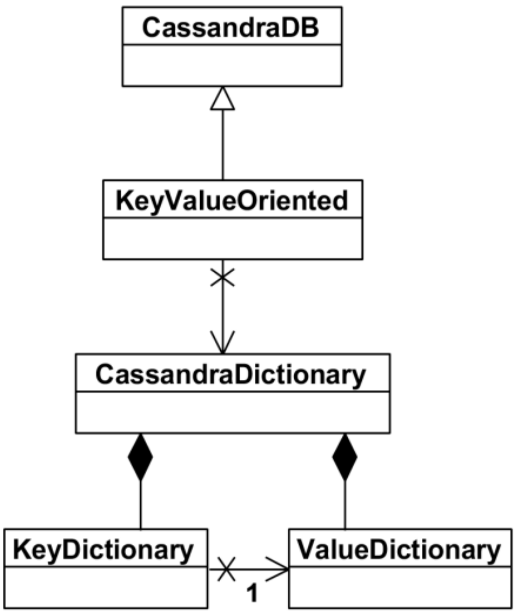

Fig. 4.Meta-model of Cassandra database.

C. MongoDB

MongoDB is the most well-known NoSQL database. It is an open-source document-oriented database. MongoDB is a scalable and accessible database. He is in C ++. MongoDB can also be used as a file system. In MongoDB, JavaScript can be used as a query language. Using the horizontal scales of MongoDB from sharding. It is very useful in popular JavaScript frames. People really like sharding, advanced text search, gridFS. Amazing performance and new features propelled this NoSQL database to the top of our list.

The following figure shows the meta-model we proposed for MongoDB:

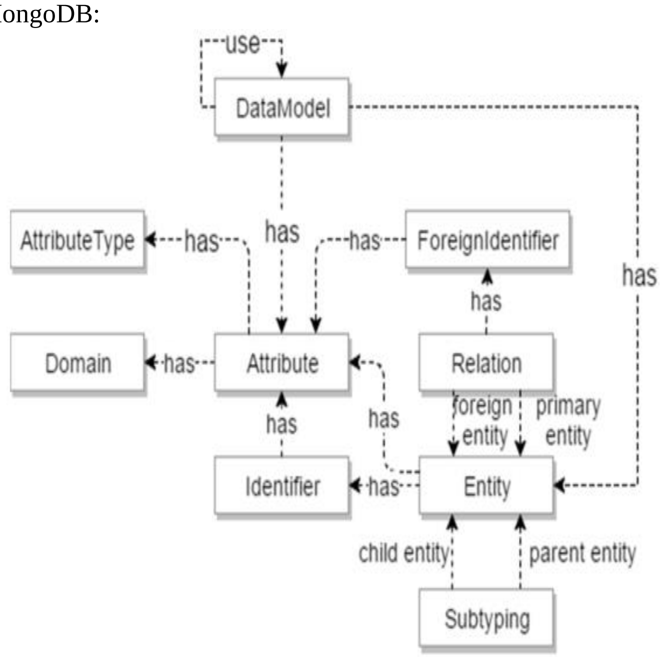

Fig. 5.Meta-model for MongoDB.

Document Models Meta-model layer allows us to define multiple models for semantic document data. Each of these models describes a specific domain and metadata for the data layer.

The first and most important domain for our work is the document management domain itself. Indeed, MongoDb implicitly covers all document management system domain. But that knowledge is embedded in the application and relational data, and we need to extract it to define our document management model.
After analyzing, we have extracted the domain-specific parts of the MongoDb relation model and define as an instance of document meta-model. This model contains all the essential concepts of the document management domain. For simplicity, only a small part of this model is illustrated in Fig. 5. The document management model is a must for our annotation system but not enough, because its coverage is very limited. The contents of documents saved by users not only contain data about the document management domain but also about other domains. The following figure shows an instance of document meta-model:

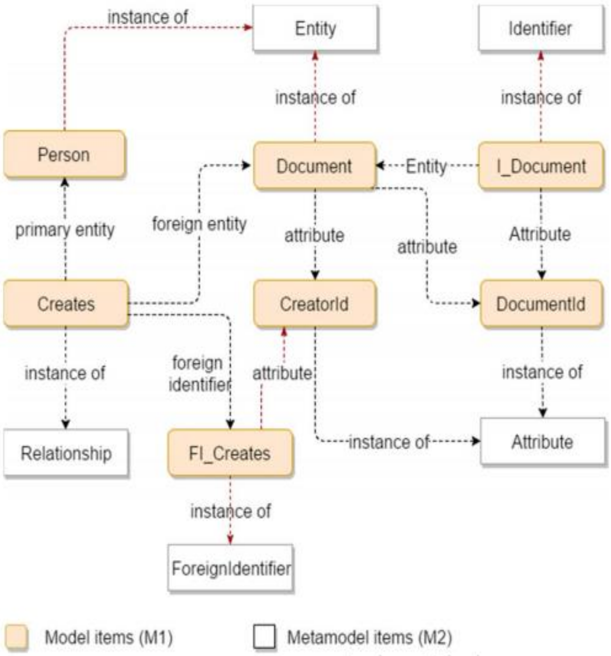

Fig. 6.An instance of a document meta-model.

D. Neo4j

Neo4j is considered a native graph database because it effectively implements the property graph model down to the storage level. This means that the data is stored exactly as you save it on a whiteboard and that the database uses pointers to navigate and navigate the chart. Neo4j has both a Community Edition and an Enterprise Edition of the database. Enterprise Edition includes everything Community Edition has to offer, plus additional corporate requirements such as backups, clustering, and failover capabilities.

The following figure shows the meta-model that we proposed to manage the graph-oriented databases:

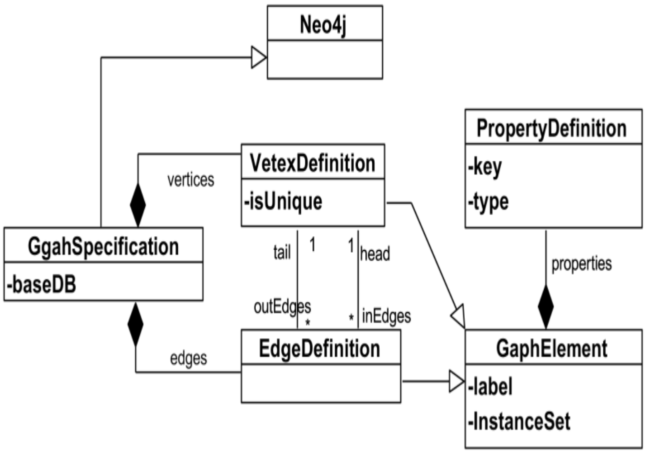

Fig. 7. Meta-model for Neo4j.

After defining the meta-models of the four types of NoSQL databases. We now move to define the transformation rules that will allow us to move from our Logical PIM NoSQL to the PSM of the four tools used in our research project.

V. TRANSFORMATIONS

In the logical level of the description of a database, the implementation choices are not completely specified. The principles of organization of the data are specified but it is abstracted from the DBMS used to implement the database (this choice is made at the physical level); only the type of DBMS is taken into account. We selected a NoSQL system of type oriented columns. This choice was dictated by the needs of our applications based on multicriteria queries involving several attributes simultaneously. But column-oriented systems offer storage techniques that serialize the values of columns and thus accelerate access to data. The problem, therefore, consists of moving from a conceptual database schema (DCL) to a physical NoSQL schema that will be implemented. But several column-oriented NoSQL systems coexist; the best known are BigTable, Chang et al. [28], HBase, Carstoiu et al. [29], Cassandra and Accumulo. They present specific technical specificities that are mainly related to implantation techniques. To disregard these specificities, we will integrate the logical level in the matching process between schemas. According to the column-oriented model, a database (BD) consists of a set of tables. A table is used to group objects of variable size as lines; each of them is identified by a unique identifier (Id) whose type is denoted "key-line". Generally, we group in a table strongly bound objects; for example, employees, the services they belong to and the projects in which they participate. By default, we will store the database in a single table marked T; but this setting can be changed by the data administrator. The table T is associated with a set of families of columns {f_11, ..., f_q}. The schema of a family f is a triplet (N, COL, Id) where:

- f.N is the name of the family.

Retrieval Number: A1619109119/2019©BEIESP
DOI: 10.35940/ijeat.A1619.109119
Journal Website: www.ijeat.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

- f.COL = $\{col_1, ..., col_q\}$ is a set of q columns present in each line of T described by f. The schema of a col column is a triplet (N, T, TS) where "col.N" represents the name of the column, "col.T" its type and "col.TS" the TimeStamp (timestamp). In this article, we do not consider the timestamp of the data.
- f.Id is a unique identifier of the family of columns of type "key-line".

The following figure illustrates an example of transform ATL code to the NoSQL Cassandra, MongoDB databases.

```
rule TransformationGenericModelToCassandraModel {

From l: LogicalPIM!DataBase,
 To
    c: CassandraPSM!KeySpace (ColumnsFamily()<-KeySpace{name := self.name;columnsfamily:=self.tabl)
}
rule Table2Columns-Family{
 From
   t: Table!
 To
   cf:Columns-Family! (ColumnsFamily()<-LogicalPIM :CassandraPSM::toColumnsFamily )
}

rule Attribute2Column{
 From
   A: Attribute!
 To
   c:Column! (Attribute()<- CassandraPSM::Column {name:=self.name;type:=self.typea -> map toType(); })

   A: CassandraPSM::Type{if(self.typea = "Rid"){type:='Int';}endif; type:=self.typea;
}
```

Fig. 8.An extract of ATL transformations.

VI. EXPERIENCES AND EVALUATION

In this section, we briefly present the techniques we used to set up the experimental environment. Since our approach is based on MDA, we need an infrastructure adapted to metamodeling, modeling, and M2M (Model-To-Model) transformations. EMF provides a set of tools for introducing a model-driven development approach within the Eclipse environment. These tools provide three main features. The first is the definition of a metamodel representing the concepts used by the user, the second is the creation of models instantiating this metamodel and the third is the transformation from model to model and from model to text.

A. Configuration

In this section, we discuss the techniques we used to implement the approach presented in Figure XX. Version 5 of the Hadoop Cloudera distribution (CDH 5, Cloudera Distribution Including Apache Hadoop) was used to install Hadoop and HBase. The use of CDH5 is done on VirtualBox (v 4.3.20) with Cloudera QuickStart Virtual Machine 6. This is a preconfigured solution that facilitates the installation of Hadoop and several of its subprojects, such as HBase, Carstoiu et al. [30], Hive, Thusoo et al. [31] and Impala 7. For the sake of efficiency, we chose this solution which contains all the necessary components for the implementation of our database. Under HBase, several solutions for implementing a database are possible. We have opted for those that best fit the logic model that has been proposed; in particular, we have chosen to define a single family of columns per row of the table.

B. Experiments

To carry out our tests, we used two datasets, to better measure the transformation execution time to the four NoSQL systems: Cassandra the key/value oriented database, HBASE the column-oriented database, MongoDB the database document-oriented, and Neo4J the graph-oriented database. The results are shown in Figure XX below:

Table-I: transformations runtime.

Database	Cassandra	HBas e	MongoD B	Neo 4J
Transformati on time	763	1045	1232	2509

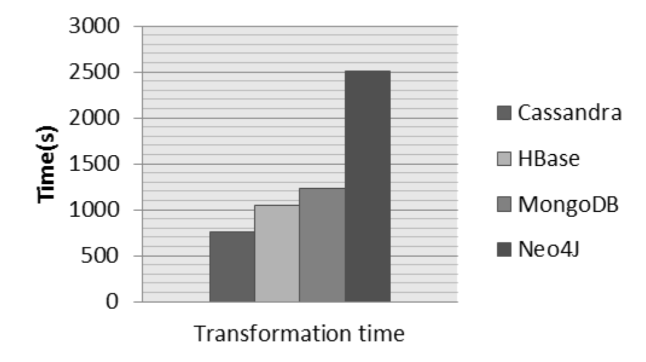

Fig. 9.NoSQL systems transformations time

VII. DISCUSSION

The analysis of the results obtained shows that the transformation time is the model of the database, the Neo4J database is very time-consuming compared to other systems, thanks to the graphical architecture of this database, the model key/value requires a smaller time compared to the other solution since all the data is stored in key/value format, but we know that it is not designed for the majority of the domains except some cases of uses like the files log.

Also, HBase and Cassandra use a selectable replication factor, while MongoDB uses a master-slave replication factor. An additional difference between HBase and MongoDB and Cassandra is that HBase and Cassandra have triggers, but MongoDB has no triggers. Besides, HBase does not have a secondary index while MongoDB has secondary indexes and Cassandra has restricted secondary indexes. HBase is a base apart because it is intimately linked to Hadoop, of which it is a sub-project. It is also installed on its HDFS distributed file system. Intended for high volumes, HBase gives more priority to querying data consistency. To Conclude HBase, MongoDB Neo4J, and Cassandra are four NoSQL database systems or nonrelational systems. HBase and Cassandra are c / e-oriented and column-oriented databases, while MongoDB is a document-oriented database. Neo4J is graph-oriented. This is the difference between HBase, MongoDB, Cassandra, and Neo4J. They are used for various applications such as Big Data, Content Management, Mobile and Social Infrastructure, and Data Hubs. So the choice of application types we have presented a schema transformation approach to its four NoSQL solutions. The choice of the most appropriate DBMS category for a given application is related to the nature of the processing (queries) applied to the data. But this choice is not exclusive since, in each category, the DBMS can provide all types of treatments, sometimes at the cost of some heaviness or more extensive programming.

VIII. CONCLUSION

Companies are developing applications that manipulate massive databases, and we are seeing that decision-making systems are increasingly integrating these new data sources. Given the large volume, variety, and data velocity we entered the era of Big Data. And since most of today's BI tools are designed to handle structured data. In our research project, we aim to consolidate a BI system for Big Data. In continuous efforts, this paper is a progress report of our first contribution that aims to apply the techniques of model engineering to propose a universal approach to deal with Big Data is to help decision-makers make decisions.

REFERENCES

1. N. Sawant and H. (Software engineer) Shah, Big data application architecture Q & A a problem-solution approach. Apress, 2013. https://doi.org/10.1007/978-1-4302-6293-0
2. A., Erraissi A., Belangour A. Capturing Hadoop Storage Big Data Layer Meta-Concepts. In: Ezziyyani M. (eds) Advanced Intelligent Systems for Sustainable Development (AI2SD'2018). AI2SD 2018. Advances in Intelligent Systems and Computing, vol 915. Springer, Cham, 2019. https://doi.org/10.1007/978-3-030-11928-7_37
3. Angadi, A. B., Angadi, A. B., & Gull, K. C. (2013). Growth of New Databases & Analysis of NoSQL Datastores. International Journal of Advanced Research in Computer Science and Software Engineering, 3, 1307-1319.
4. Banker, Kyle. MongoDB in action. Manning Publications Co., 2011.
5. Vora, Mehul Nalin. "Hadoop-HBase for large-scale data." Proceedings of 2011 International Conference on Computer Science and Network Technology. Vol. 1. IEEE, 2011.
6. Cassandra, Apache. "Apache cassandra." Website. Available online at http://planetcassandra. org/what-is-apache-cassandra (2014): 13.
7. Vukotic, Aleksa, et al. Neo4j in action. Manning Publications Co., 2014.
8. Fatima Kalna, Abdessamad Belangour, A Meta-model for Diverse Data Sources in Business Intelligence, American Journal of Embedded Systems and Applications (AJESA), 2019.
9. Darmont, Jérôme, Omar Boussaid, Jean-Christian Ralaivao, and Kamel Aouiche. "An architecture framework for complex data warehouses." arXiv preprint arXiv:0707.1534 (2007).
10. Pedersen, Torben Bach, and Christian S. Jensen. "Multidimensional data modeling for complex data." In Proceedings 15th International Conference on Data Engineering (Cat. No. 99CB36337), pp. 336-345. IEEE.
11. Tanasescu, Adrian, Omar Boussaid, and Fadila Bentayeb. "Preparing complex data for warehousing." In The 3rd ACS/IEEE International Conference onComputer Systems and Applications, 2005., p. 30. IEEE.
12. Midouni, Sid Ahmed Djallal, Jérôme Darmont, and Fadila Bentayeb. "Approche de modélisation multidimensionnelle des données complexes: Application aux données médicales." 2009.
13. Chevalier, Max, Mohammed El Malki, Arlind Kopliku, Olivier Teste, and Ronan Tournier. "Entrepôts de données multidimensionnelles NoSQL." 2015.
14. Li, Chongxin. "Transforming relational database into HBase: A case study." In 2010 IEEE international conference on software engineering and service sciences, pp. 683-687. IEEE, 2010.
15. Li, Yan, Ping Gu, and Chao Zhang. "Transforming UML class diagrams into HBase based on meta-model." In 2014 International Conference on Information Science, Electronics and Electrical Engineering, vol. 2, pp. 720-724. IEEE, 2014.
16. M. Banane, and Abdessamad Belangour. « An Evaluation and Comparative study of massive RDF Data management approaches based on Big Data Technologies». International Journal of Emerging Trends in Engineering Research. 7, nº 7 (2019): 48 – 53, 2019.
17. M. Banane, and Abdessamad Belangour. « RDFMongo: A MongoDB Distributed and Scalable RDF management system based on Meta-model». International Journal of Advanced Trends in Computer Science and Engineering 8, nº 3 (2019): 734 – 741.
18. M. Banane, and Abdessamad Belangour. « Querying massive RDF data using Spark». International Journal of Advanced Trends in Computer Science and Engineering 8, nº 4 (2019): 1481 - 1486.
19. M. Banane, and Abdessamad Belangour. "New Approach based on Model Driven Engineering for Processing Complex SPARQL Queries on Hive." International Journal of Advanced Computer Science and Applications (IJACSA) 10, no. 4 (2019).
20. Erraissi, Allae, and Abdessamad Belangour. « Hadoop Storage Big Data Layer: Meta-Modeling of Key Concepts and Features ». International Journal of Advanced Trends in Computer Science and Engineering 8, nº 3, 646 53, 2019. https://doi.org/10.30534/ijatcse/2019/49832019
21. Erraissi, Allae, and Abdessamad Belangour. « Meta-Modeling of Big Data Management Layer ». International Journal of Emerging Trends in Engineering Research 7, no 7, 36-43, 2019. https://doi.org/10.30534/ijeter/2019/01772019.
22. Erraissi, A., & Belangour, A. Data sources and ingestion big data layers: meta-modeling of key concepts and features. International Journal of Engineering & Technology, 7(4), 3607–3612, 2018. https://doi.org/10.14419/ijet.v7i4.21742
23. Erraissi, Allae, and Abdessamad Belangour. « Meta-Modeling of Big Data Visualization Layer Using On-Line Analytical Processing (OLAP) ». International Journal of Advanced Trends in Computer Science and Engineering 8, nº 4, 990-98, 2019. https://doi.org/10.30534/ijatcse/2019/02842019.
24. A. Erraissi and A. Belangour, "Meta-modeling of Zookeeper and MapReduce processing," 2018 International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS), Kenitra, Morocco, pp. 1-5, 2018. https://doi.org/10.1109/ICECOCS.2018.8610630

25. Fatima Kalna, Allae Erraissi, Mouad Banane, Belangour "A Scalable Business Intelligence Decision-Making System in The Era of Big Data" International Journal of Innovative Technology and Exploring Engineering.
26. Kleppe, Anneke G., et al. MDA explained: the model driven architecture: practice and promise. Addison-Wesley Professional, 2003.
27. Vora, Mehul Nalin. "Hadoop-HBase for large-scale data." In Proceedings of 2011 International Conference on Computer Science and Network Technology, vol. 1, pp. 601-605. IEEE, 2011.
28. Chang, Fay, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes, and Robert E. Gruber. "Bigtable: A distributed storage system for structured data." ACM Transactions on Computer Systems (TOCS) 26, no. 2 (2008): 4.
29. Carstoiu, Dorin, Elena Lepadatu, and Mihai Gaspar. "Hbase-non sql database, performances evaluation." In in Computer Science, Master in Computer Science, and PhD in Computer Science. 2010.
30. Carstoiu, Dorin, Elena Lepadatu, and Mihai Gaspar. "Hbase-non sql database, performances evaluation." In in Computer Science, Master in Computer Science, and PhD in Computer Science. 2010.
31. Thusoo, Ashish, Joydeep Sen Sarma, Namit Jain, Zheng Shao, Prasad Chakka, Suresh Anthony, Hao Liu, Pete Wyckoff, and Raghotham Murthy. "Hive: a warehousing solution over a map-reduce framework." Proceedings of the VLDB Endowment 2, no. 2 (2009): 1626-1629.

AUTHORS PROFILE

Fatima Kalna is a Business Intelligence Engineer and a Ph.D. at the Faculty of Sciences Ben M'Sik, Laboratory of Information Technology and Modeling (LTIM), Hassan II University of Casablanca, Morocco. His research fields: Business Intelligence (BI), Decision-making systems, Model-Driven Engineering, and Big Data.

Abdessamad Belangour is a Full Professor at the Faculty of Sciences at the Hassan II University, Casablanca, Morocco. He is mainly working on Model-Driven Engineering approaches and their applications on new emerging technologies such as Big Data, Business Intelligence, Cloud Computing, Internet of Things, Real-time embedded systems, etc.

Mouad Banane is a Ph.D. at the Faculty of Sciences Ben M'Sik, Laboratory of Information Technology and Modeling (LTIM), Hassan II University of Casablanca, Morocco. His research fields: Model-Driven Engineering, Semantic Web, and Big Data.

Allae Erraissi is a Ph.D. on computer science at the Faculty of Sciences Ben M'Sik at the Hassan II University, Casablanca, Morocco. He won his Master Degree in Information Sciences and Engineering from the same University in 2016 and is currently working as a Mathematics teacher in a High school in Casablanca, Morocco. His main interests are the new technologies namely Model-driven engineering, Cloud Computing, and Big Data.