ISSN: 1646-9895

risti

Revista Ibérica de Sistemas e Tecnologias de Informação
Iberian Journal of Information Systems and Technologies

A b r i l 1 9 • A p r i l 1 9

Nº E19

risti
Revista Ibérica de Sistemas e Tecnologias de Informação
Iberian Journal of Information Systems and Technologies

Edição / Edition
Nº. E19, 04/2019

ISSN: 1646-9895

Indexação / Indexing
Academic Journals Database, CiteFactor, Dialnet, DOAJ, DOI, EBSCO, GALE, Index-Copernicus, Index of Information Systems Journals, Latindex, ProQuest, QUALIS, SCImago, SCOPUS, SIS, Ulrich's.

Propriedade e Publicação / Ownership and Publication
AISTI – Associação Ibérica de Sistemas e Tecnologias de Informação
Rua Quinta do Roseiral 76, 4435-209 Rio Tinto, Portugal
E-mail: aistic@gmail.com
Web: http://www.aisti.eu

risti
Revista Ibérica de Sistemas e Tecnologias de Informação
Iberian Journal of Information Systems and Technologies

Recebido/Submission: 24/11/2018
Aceitação/Acceptance: 22/01/2019

An approach to Big Data Modeling for Key-Value NoSQL Databases

Diana Martinez-Mosquera[1], Sergio Lujan-Mora[2], Rosa Navarrete[3], Tannia Cecilia Mayorga[4], Henry Rodrigo Vivanco Herrera[5]

sdmm1@alu.ua.es, sergio.lujan@ua.es, rosa.navarrete@epn.edu.ec, tmayorga@uisrael.edu.ec, hvivanco@uisrael.edu.ec

[1,2] University of Alicante, 03690, San Vicente del Raspeig, Spain.

[3] Escuela Politécnica Nacional, 170525 , Quito, Ecuador.

[4,5] Universidad Tecnológica Israel, 170522, Quito, Ecuador.

Pages:519–530

Abstract: The scientific community has a special interest in providing solutions to deal with a huge amount of data generated by the Internet, mobile devices, and sensors, among others. One of the foremost research approaches has been in not only SQL (NoSQL) databases, mainly used to handle Big Data. A state of the art review presented in this article let us argue on the need to define modeling techniques to depict how data will be structured in NoSQL databases. The majority of studies have focused on structured data and column oriented databases; thus, we propose an approach for semi-structured data at conceptual and logical modeling levels, using the Unified Modeling Language and key-value databases. The logical model is attained from a class diagram, with the use of transformation rules based on some aspects of the Query View Transformation. Furthermore, our proposal presents a case study concerning data in security log files.

***Keywords*:** big data; key-value; modeling; UML; semi-structured.

1. Introduction

The permanent growing of data generated and collected by companies is in the range of terabytes of information; this high amount of data is known as Big Data (Martinez-Mosquera, Lujan-Mora & Parra, 2017). Big Data is a concept widely used to describe a big amount of data that comply with some specific features like volume, velocity, variety, variability, value, among others (Qaiyum, Aziz & Jaafar, 2016). Big Data researchers have commendably focused in different approaches, one of them about not only SQL (NoSQL) databases, since relational database paradigms are not adequate and NoSQL, by contrast, handles Big Data with high performance (Abdelhedi et al., 2016).

Although this type of database was designed to operate without schema, there is an underlying need for defining the data structure in the database (Abdelhedi et al., 2016; Santos & Costa, 2016). The reason is modeling helps to design the data processing flow (Da Silva et al., 2014). Many researches have proposed Big Data modeling, but the major part

are focused at a conceptual level and structured data. With this motivation, we propose an approach based on the Model Driven Architecture (MDA) (MDA Specifications) to model semi-structured Big Data at a logical level for a key-value NoSQL database. MDA is an architecture funded by the Object Management Group (OMG) and provides the guidelines for structuring models in different levels (Lano, 2005; MDA Specifications, n. d.).

Our main challenge is modeling semi-structured data extending the Unified Modeling Language (UML). Semi-structured data contains metadata to describe its structure, it is not organized, it is generally plain text, and it is not stored in databases (Khalifa et al., 2016). On the other hand, the high number of sources that generate semi-structured data, such as security and web logs, data from sensors, JavaScript Object Notation (JSON) files, among others, makes important to model this type of data. Despite that, there is a minimal research effort in this topic. Furthermore, Big Data sources must be analyzed, managed, organized for finally support business decisions (Barbierato, Gribaudo & Iacono, 2013).

In this article, we present three different type of diagrams for modeling semi-structured Big Data:

1. Conceptual Diagram, a deployment diagram based on the UML to depict the physical resources used for the storage of the collected data from the source to the target.
2. Intermediate Logical Diagram, a class diagram based on UML as an intermediate procedure to apply the transformation rules in order to derive the logical modeling diagram for a key-value database.
3. Key-Value Logical Diagram, a class diagram based on UML to depict the key-value logical model for a NoSQL database.

We have employed the Query View Transformation (QVT) standard to define the transformation rules (Kurtev, 2007).

To provide a better understanding, we present as a case study the modeling of security log files for a key-value NoSQL database. Following the MDA, the case of study represents the Platform Specific Model (PSM) derived from the Platform Independent Model (PIM) generated earlier. As result, we demonstrate how to transform a semi-structured Big Data conceptual model into a Big Data logical model based on a key-value approach.

This paper is organized as follows. Section 2 presents the state of the art about Big Data modeling using UML, summarizes the approaches from different authors and discusses the main results attained from them researches. Section 3 explains the basic concepts used for our research. Section 4 describes the method used to conduct our proposal, detailing concepts related to our research and the procedure to achieve our goal. Section 5 details the case study to explain with an example how our proposal can be applied. Finally, section 6 portrays our findings, conclusions and planned future works.

2. State of the Art

Proposals related to Big Data modeling are described in several relevant studies. In this section, we summarize their main findings related to our matter of interest.

Abdelhedi et al. (2016) propose an approach to generate a column-oriented NoSQL model from a conceptual model, using the QVT standard to define the transformation rules from one to the other. As a case study, they present an example related to health care domain. The study is based on the MDA and the PSM and it is addressed for Cassandra and HBase. Cassandra is a distributed NoSQL database based on a key-value and column oriented storage model. HBase is also a NoSQL database used for real-time read/write random-access to very large databases. In contrast to our proposal, they are focused on structured data and column-oriented databases and we are interested into semi-structured data and key-value databases; moreover, we have taken as case study the security domain and the CouchDB database as PSM.

Feng, Zhang & Zhou (2015) describe how to obtain data models for Cassandra from UML class diagrams; the source metamodel is built by simplifying the UML class diagram definition and the target data model is produced by studying the Cassandra structure. This work also presents the conversion from a relational database model to a NoSQL database model. Consequently, this research explains how to transform structured data at the logical level and not the semi-structured data as our research aims. In addition, and while we propose the use of QVT, they do not define any transformation rules.

Da Silva et al. (2014) present a proposal for modeling real time platforms with the use of a UML profile called Modeling and Analysis of Real Time Embedded Systems (MARTE). They illustrate the transformation from a relational database deployment model to a key-value database deployment model, mainly representing the used physical resources. As a case study, they describe a project platform called Juniper. In contrast to our study, this work does not involve the logical level and instead of, only the physical level is presented.

Jutla, Bodorik & Ali (2013) extend UML with ribbon icons focused on Big Data privacy services. According Microsoft, ribbons are the modern way to use commands with a minimum number of clicks. Their work covers conceptual modeling level and the extensions are useful for use case diagrams. This research does not support logical modeling level, since it was just designed to help software developers to represent their requirements. By contrast, to our work, we present the logical level modeling for semi-structured Big Data using deployment and class diagrams.

Santos & Costa (2016) present an automatic transformation of a multidimensional schema into a tabular schema for the Hive database. Hive is constituted by a set of tables and it uses a column-oriented model. Multidimensional modeling is used in traditional data warehouses (Lujan-Mora, 2006). In order to conduct the proposed transformation, the authors define a set of rules at the logical level. Our work differs in several aspects, we raise the transformation for semi-structured Big Data in the source and the target is a key-value database; furthermore, we have used QVT for defining the transformation rules and our proposal is based on MDA.

Sousa et al., (2017) propose transformation definitions from Action Language for Foundational UML (ALF) to MapReduce model by using Atlas Transformation Language (ATL). It is also based on MDA. In addition, the work presents a case study through the modeling of a word count application. The key difference with our approach is the transformation language and the data type, because we propose QVT and semi-structured Big Data.

As conclusion, the major part of the analyzed works deals with structured data and the targets are column-oriented databases. We did not find research about modeling semi-structured data for key-value NoSQL databases; thus, in this work we propose to handle this type of data. Furthermore, we know Big Data is generally unstructured and they are not stored in relational databases.

3. Basic Concepts

Data modeling is a technique which main goal is to depict the data element characteristics into a diagram. The graphical representation describes how the data are used in a process. Data modeling has three main levels conceptual, logical and physical. Thus, the highest level representation about the data is done at the conceptual level. The logical structure of the data is described at the logical level and the specific data model is depicted at the physical level (Martinez-Mosquera et al., 2017). Conceptual, logical and physical modeling levels can involve PIM and PSM data models within MDA architecture (Abdelhedi et al., 2016).

UML is a standard for modeling business and similar processes. The concepts of UML are grouped into three major sub-parts:

1. Structure modeling includes classes, components, composites and deployments.
2. Behavior modeling is composed by action, activities, common behaviors, interactions, state machines and use cases.
3. Supplementary modeling is formed by auxiliary constructions and profiles (UML ISO IEC 19505-2, n. d.).

All of these sub-parts can be combined as per the creator preferences (Gomez et al., 2016).

To achieve our goal, we use structure modeling with deployment and class diagrams. A UML deployment diagram depicts artifacts and nodes according to deployment relationships, in other words the system physical architecture (ISO IEC 19505-2, n. d.); an artifact is defined as a physical entity into a node. The class diagrams deal with the basic modeling concepts of UML classes and their relationships. This study begins with a deployment diagram at the conceptual level, representing the physical resources used for semi-structured Big Data and concludes with a logical model based on UML for a key-value NoSQL database as a target (Martinez-Mosquera et al., 2017).

In Big Data, three types of data can be considered: structured data, which are data arranged by rows and columns in a relational database; semi-structured, referred to data with some of structure but not organized in relational databases; and unstructured, related to data without any format. In this paper, we have focused our effort to deal with semi-structured data, due to the limited research about this type of data, as shown in Section 2 State of the Art.

The OMG MDA provides two levels of models: PIM and PSM. PIM defines business functionality and behavior independent of a technology; this enables us to model exactly the business rules in accordance with the customer's needs. When a model of a system is defined in terms of a specific platform, it is PSM (MDA Specifications, n. d.). The MDA approach starts with PIM, then it is transformed into PSM and PSM is finally

transformed to code (Rahmani et al., 2006). Among these models, we have selected PIM to describe semi-structured data separated from specific platforms at the conceptual and logical modeling level.

4. Method

When we talk about Big Data, traditional data models as relational model are unusual. In addition, new systems have been deployed for this goal, these are known as NoSQL. NoSQL systems are classified as follows: key-value stores, document stores, extensible record/column oriented stores and graph oriented (Iribarne et al., 2017). Our approach is focused into key-value stores, since it is suitable for semi-structured data. In summary, the Fig. 1 presents the flow of our proposal.

4.1. Modeling Semi-structured Big Data

In this section, we present the process to implement our proposal in order to obtain the key-value model. We have divided the modeling into three subsections: deployment diagram, class diagram, and key-value model. At the three phases, we have used UML as modeling language.

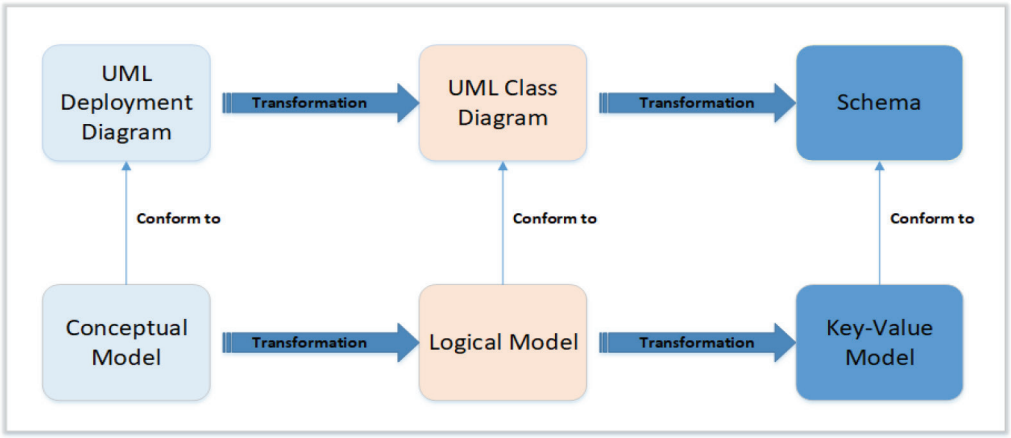

Figure 1 – Process for Modeling Semi-Structured Big Data

Deployment Diagram

As we propose the use of PIM as MDA, we map the main physical entities to UML concepts separately of any specific platform. Table 1 contains the proposed details to follow; they are based on (Da Silva et al., 2014). Storages, software applications and operating systems will be represented by nodes, and semi-structured data with artifacts. This map will allow us depicting the deployment diagram for semi-structured Big Data at the source and the target.

With the provided UML concepts, we can represent our proposed modeling at a conceptual level. We consider a UML deployment diagram as a set of nodes N={Node 1, Node 2 ...

Node n}, where 1 is the identifier for the first node and n for the last one, each node is composed by artifacts Nodei={Artifact 1, Artifact 2 ... Artifact n}. A string identifies every node and artifact. Fig. 2 presents an example of the deployment diagram for the source and the target of the semi-structured Big Data. At the source, the used resources for extracting the semi-structured data are represented by a storage infrastructure; at the target, another storage infrastructure is represented by contains operating system, application software and the elements to store the data.

A deployment diagram was necessary to know how the semi-structured Big Data can be extracted from the source where they are arisen. There are many tools for this purpose; independently of the chosen tool, it must be represented with this conceptual modeling. The next procedure is to build the class diagram at a logical level as we planned previously.

Physical entity	UML concept
operating system	node
storage	node
semi-structured data	artifact

Table 1 – Map of physical entities to UML concepts

Class Diagram

In order to transform the UML deployment diagram into a key-value model, an intermediate procedure has been necessary. We generated a UML class diagram for representing the logical modeling for semi-structured Big Data at the source. Semi-structured Big Data does not have a predefined structure, but the different fields can be easily identified, since they have metadata or they are delimited. We represented the UML class diagram as a set of classes C={C1, C2... Cn}, each class contains a set of instances I= {I1, I2...In}, every class and instance is identified by a string and belongs to a specific type.

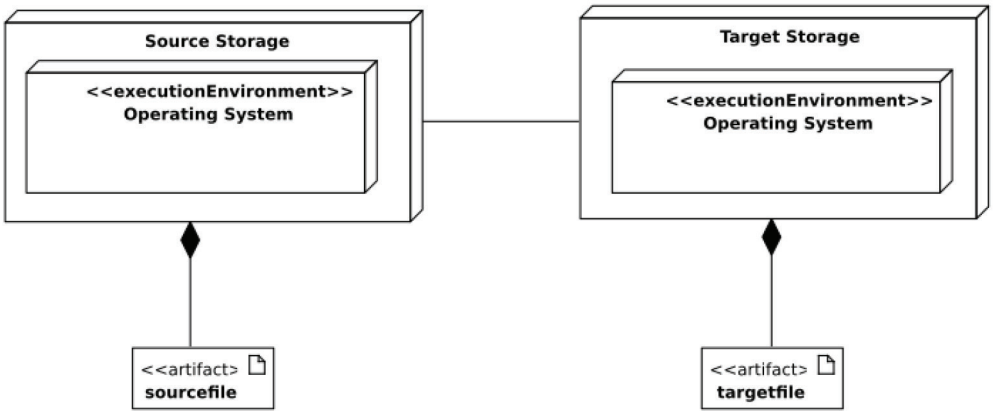

Figure 2 – Deployment Diagram

The Fig. 3 presents the UML class diagram for the semi-structured Big Data at source. A package depicts the file that includes the data, and the class contains the field. At this stage, the QVT standard was used to define transformation rules in order to attain the key-value logical model. From the QVT specification, a transformation allows relations between model elements of source alter to target models. A relation specifies how two types of object diagrams, called domain patterns, relate to each other. The operational mappings are summarized into the Fig 4. Source domain is called SourceFile and target domain is called Table. Domain patterns are the file in the source and key and value in the target.

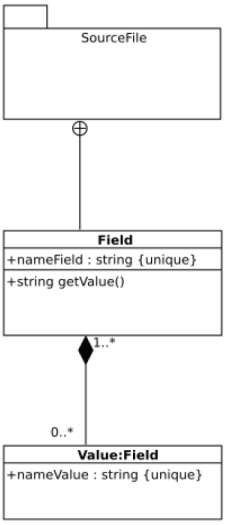

Figure 3 – Source Class Diagram

A key-value NoSQL database stores, as the name suggests, a key and a value. Usually the key data type is a string and the value is a list or an array. However, it is considered as straightforward, since it enables to store a huge amount of data. The transformation rules, to gather the key-value logical model, are based on (Abdelhedi et al., 2016) and are outlined as follows:

- Package is converted to a single table because the key-value model supports only one table.
- Class is converted to key. Semi-structured data, usually includes field names separated by a delimiter, every field (class) will be stored as a key in the NoSQL database. Here, we propose the use of two identifiers, one of them to identify the row and another for determining the delimiter type.
- Instance is converted to value. The records for every field will be stored in the value column of the NoSQL database as a data set.

Key-Value Model In the key-value model, there is a single table composed by two columns, key and value {K,V}, key is a class that contains field name as an attribute and the getValues() operation to take the values from the Values class K={K1, K2… Kn}. Values class includes the dataset of every key Ki ={Vi}.

Fig. 5 presents the attained key-value model for semi-structured Big Data, after applying the transformation rules to the logical modeling in the UML class diagram. A single table is presented, that can be associated with many keys and every key with many values.

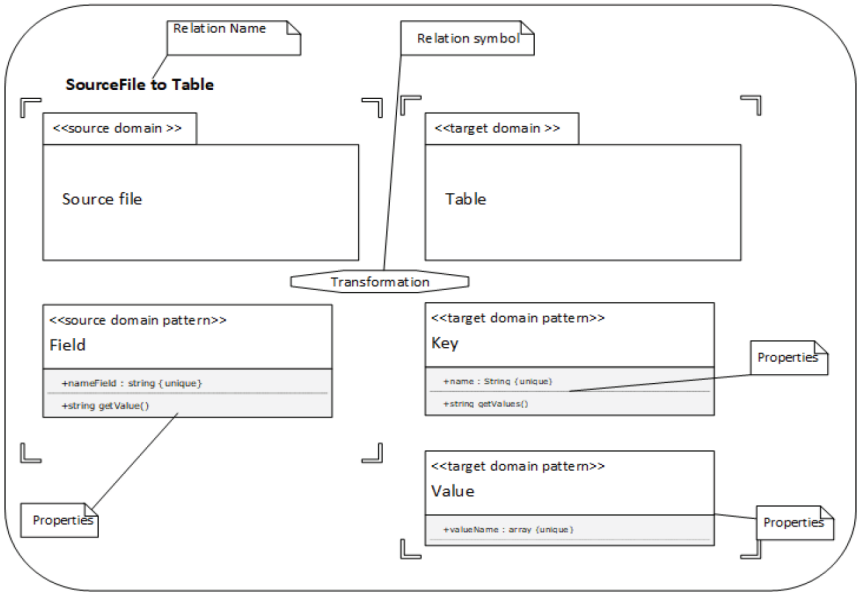

Figure 4 – QVT Relationship

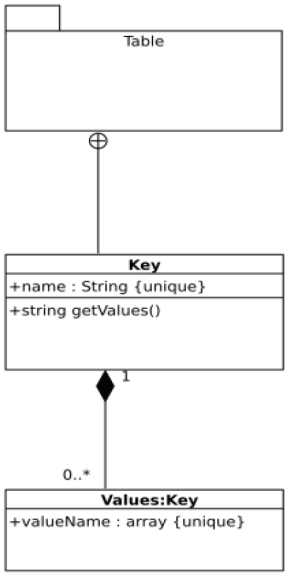

Figure 5 – Key-value Model

5. Case Study

In this section, we present a case study to show how to apply our proposal. At the source, we collected records from a network firewall, since event logs, that generally contain metadata to describe their structure, are considered as semi-structured data (Khalifa et al., 2016). Firewall records can be classified as Big Data; they comply at least with the features volume, velocity, variety, veracity and value (Qaiyum et al., 2016).

We will specify the tools and database types used for the case study. The source is a firewall that generates record files in real time, but this study is still adapted only for batch processing. Through the Flume tool for extracting log records, these data will be stored into a key-value database in CouchDB. Flume is a tool for collecting web or database log records. Flume comprises of agents that are installed into the source, the spooling directory and the target, the storage place (Sameer & Madhu, 2014). Apache CouchDB is a NoSQL database, which supports several data formats, for instance JSON and the Atom Publishing Protocol (AtomPub) (Han et al., 2011).

Fig. 6 presents a fragment of the log file extracted from the firewall. There are five field names and two rows with the collected values for every field; the delimiter parameter is a comma.

```
Severity, Event Name, Start Time, Source, Destination
Critical, Non Compliant DNS,12:24:51 07 Nov 2017,11.12.13.14,sea10.ff.avast.com
Critical,Non Compliant DNS,12:24:51 07 Nov 2017,11.12.13.15,ed.73.1732.ip4.static.sl-reverse.com
```

Figure 6 – A Fragment of the Security Log File

Fig. 7 presents the deployment diagram for the referred scenario, two nodes where the operating systems and Flume agents are installed. This figure is based on Fig. 2 that depicts the source and the target for semi-structured Big Data. Nodes are the firewall and the storage physical units, the Flumes agents and the operating systems are nodes contained into the main node. The log file is an artifact related to the Firewall node.

Associations show the relationship between elements in UML. For Fig. 6, one or many log files can be associated with a firewall, and many firewalls can be associated with a storage.

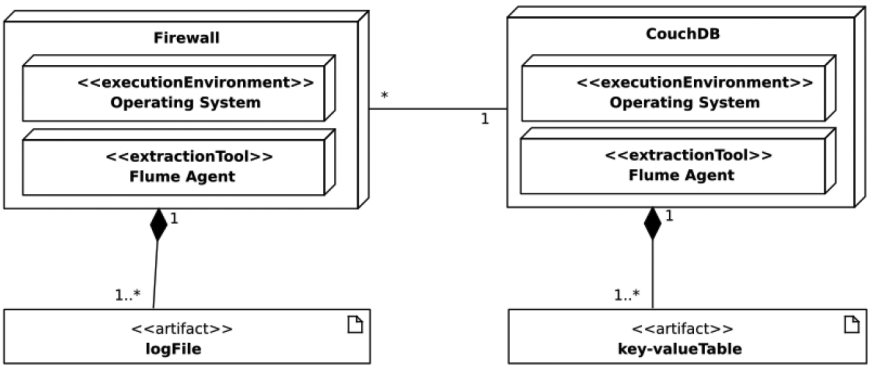

Figure 7 – UML Deployment Diagram for Security Logs

Fig. 8 presents the UML class diagram for this log file, as we early mentioned, this intermediate procedure is necessary before conducting the transformation. We have depicted the log file, where the semi-structured data comes from, as a package, the fields associated to the log and the concerned attributes and operations as classes and the field values as instances. Here, none or many values belong to one field and a log file can contain one or many fields.

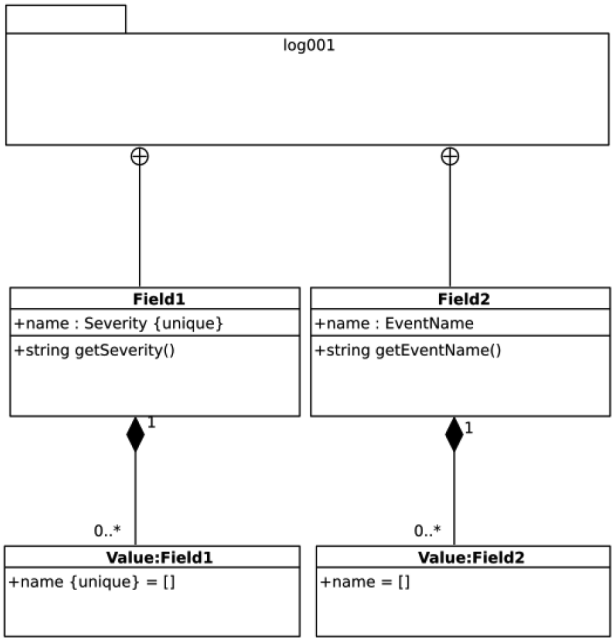

Figure 8 – UML Class Diagram for Security Logs

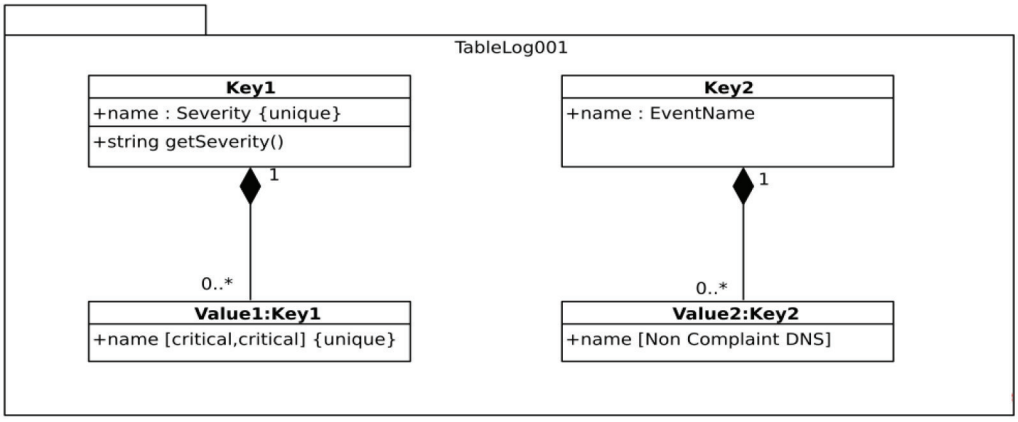

Figure 9 – Key-value Modeling for Security Logs

To obtain the key-value logical model, we applied the previously predefined QVT rules: package is a single table, class is converted to key and instance is converted to value. Fig. 9 presents the model into a class diagram composed by packages, classes and instances. As it can be seen, the conversion is easy and can be replicated in another scenario.

6. Conclusions

In this paper, we presented an approach based on MDA supported on PIM to modeling semi-structured Big Data from the conceptual to a logical level for a key-value NoSQL database. We departed from a UML deployment diagram that depicts the physical resources at the conceptual level, then we created a UML class diagram as an intermediate step to apply the proposed transformation rules based on the QVT standard and finally to implement the modeling at a logical level for a key-value model. From the specified PIM model, we presented the PSM model for a case study concerned to security log files, using the data extraction tool Flume and the key-value database CouchDB. All the diagrams were performed with the use of a Visual Paradigm Community version, achieving our main goal to present how to model semi-structured Big Data for key-value NoSQL database.

As future work, we intend to assess our proposal through a modeling framework, with the purpose to perform all the transformation with the tool and finally to obtain the database schema. We also plan to model real-time semi-structured Big Data and evaluating the NoSQL database types to know the better alternative. We expect to present novel approaches for modeling semi-structured and unstructured Big Data at the conceptual, logical and physical levels; automatizing the model transformation with QVT and incorporating the use of OMG Object Constraint Language (OCL) standard for defining detailed transformation processes. In addition, we will extend UML through a profile with stereotypes to depict a particular diagram.

References

Abdelhedi, F., Brahim, A. A., Atigui, F., & Zurfiuh, G. (2016). Big Data and Knowledge Management: How to Implement Conceptual Models in NoSQL Systems. *In 8th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, 235–240. DOI: 10.5220/0006082302350240

Barbierato, E., Gribaudo, M., & Iacono, M. (2013). A performance modeling language for Big Data architectures. *In 27th European Conference on Modelling and Simulation,* 511–517.

Da Silva, M. A. A., Sadovykh, A., Bagnato, A., Cheptsov, A., & Adam, L. (2014). JUNIPER: towards modeling approach enabling efficient platform for heterogeneous big data analysis. *In 10th Central and Eastern European Software Engineering Conference*, 12. DOI: 10.1145/2687233.2687252

Gomez, A., Merseguer, J., Nitto, D., & Tamburri, D. A. (2016). Towards a UML pro le for data intensive applications. *In 2nd International Workshop on Quality-Aware Devops*, 8–23. DOI: 10.1145/2945408.2945412

Han, J., Haihong, E., Le, G., & Du, J. (2011). Survey on NoSQL database. *In 6th International Conference on Pervasive Computing and Applications*, 363–366. DOI: 10.1109/ICPCA.2011.6106531

Iribarne, L., Asensio, J. A., Padilla, N., & Criado, J. (2017). Modeling Big databased systems through ontological trading. *Software*: *Practice and Experience*, *47*(11), 1561–1596. DOI: 10.1002/spe.2488

Khalifa, S., Elshater, Y., Sundaravarathan, K., Bhat, A., Martin, P., Imam, F., & Statchuk, C. (2016). The six pillars for building big data analytics ecosystems. *ACM Computing Surveys*, *49*(2), 33. DOI: 10.1145/2963143

Kurtev, I. (2007). State of the art of QVT: A model transformation language standard. *In International Symposium on Applications of Graph Transformations with Industrial Relevance*, 377–393. DOI: 10.1007/978-3-540-89020-1_26

Lano, K. (2005). *Advanced systems design with Java, UML and MDA*. *In Elseiver Butterworth-Heneimann*. Great Britain: Elsevier.

Lujan-Mora, S., Trujillo, J., & Song, II_Yeol. (2006). A UML profile for multidimensional modeling in data warehouses. *Data and Knowledge Engineering*, *59*(2), 725–769. DOI: 10.1016/j.datak.2005.11.004.

Martinez-Mosquera, D., Lujan-Mora, S., & Parra, F. (2017). Modeling data cleaning techniques for big data. *In WWW/Internet and Applied Computing*, 310–313.

Martinez-Mosquera, D., Lujan-Mora, S., & Recalde, H. (2017). Conceptual Modeling of Big Data extract processes with UML. *In 2nd International Conference on Information Systems and Computer Science*. DOI: 10.1109/INCISCOS.2017.18

MDA Specifications. (n.d.).

Qaiyum, S., Aziz, I. A., & Jaafar, J. B. (2016). Analysis of Big Data and Quality-of-Experience in High-Density Wireless Network. *In 3rd International Conference Computer and Information Sciences*, 287–292. DOI: 10.1109/ICCOINS.2016.7783229

Unified Modeling Language (UML), ISO IEC 19505-2. (n. d.). *UML*. Retrieved from https://www.uml.org/