JISBD 2017 (La Laguna)

Ruiz, F. (Ed.), Actas de las XXII Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2017). La Laguna (Tenerife), septiembre de 2017.

Las XX Jornadas de Ingeniería del Software y Bases de Datos (JISBD 2015) se han celebrado en La Laguna del 19 al 21 de julio de 2017, como parte de las Jornadas SISTEDES.
El programa de JISBD 2017 se ha organizado en torno a sesiones temáticas o *tracks*. A continuación se detalla el contenido de las actas:

- Track RCP – Requisitos, Calidad y Pruebas

Framework for modelling and implementing secure NoSQL document databases

Carlos Blanco [1,2], Jesus Peral [3], Juan Trujillo [3], and Eduardo Fernández-Medina [1]

[1]GSyA research group. Institute of Information Technologies and Systems.
Information Systems and Technologies Department.
University of Castilla-La Mancha. Spain.
Eduardo.Fdezmedina@uclm.es
[2]ISTR research group. Dep. of Computer Science and Electronics.
University of Cantabria. Spain.
Carlos.Blanco@unican.es
[3]LUCENTIA research group. Languages and Computing Systems Department.
University of Alicante. Spain.
jperal@dlsi.ua.esjtrujillo@dlsi.ua.es

Abstract. The great amount of data managed by Big Data technologies have to be correctly assured in order to protect critical enterprise and personal information. Nevertheless, current security solutions for Big Data technologies such as NoSQL databases do not take into account the special characteristics of these technologies.
In this paper, we focus on assuring NoSQL document databases proposing a framework composed of three stages: (1) the source data set is analysed by using Natural Language Processing techniques and ontological resources in order to detect sensitive data. (2) we define a metamodel for document NoSQL databases that allows designer to specify both structural and security aspects. (3) this model is implemented into a specific document database tool, MongoDB. Finally, we apply the framework proposed to a case study with a data set of medical domain.

Keywords: Big Data, NoSQL, model, security, Natural Language Processing

1 Introduction

Nowadays, the exchange of a great volume of data is a usual activity in several domains (scientific, medical, citizens, etc.). These data include critical enterprise information and personal data which could be exposed if they are not correctly assured [7]. Nevertheless, we do not have (or we do not know how to correctly apply) security policies to assure (confidentiality, privacy, integrity, etc.) in Big Data domains [9, 6]. Furthermore, the democratization of Big Data is increasing generating more security and privacy problems that need new strategies and solutions [10].

Within this Big Data ecosystem new technologies have arised such as document, columnar and graph NoSQL databases. Nevertheless, current Big Data technologies and tools are not designed for managing the variety, velocity and complexity of these massive data sets incorporating adequate security and privacy constraints [5, 9].

The main objective of our research deals with incorporating security in Big Data technologies, focusing in this work on a certain technology, document NoSQL databases. We consider that to obtain a robust secure solution security constraints should be included at early development stages in order to be considered in design decisions. Then, they could be correctly incorporated into the final implementation.

In this way, this work presents a framework for modelling and implementing document NoSQL databases considering security issues and also apply our proposal to a case study.

The framework proposed consists on three stages: (1) The analysis of the source data set to detect sensitive data by applying Natural Language Processing (NLP) and ontological techniques; (2) The definition of a metamodel for document NoSQL databases which includes both structural and security aspects; (3) The implementation in an specific document database tool, MongoDB, considering security constraints.

The paper is organized as follows. In section 2 the related work about data integration and security are shown. Following this, the framework proposed for modelling secure document databases including data set analysis, modelling and implementation is defined. In section 4 our framework is applied to a case study of medical domain. Finally, conclusions and future work are presented.

2 Related work

Regarding to security, some works to incorporate security in Big Data have been proposed, mainly in Hadoop ecosystem [8]. Nevertheless, these proposals does not correctly fit with the special characteristics of Big Data technologies (document, columnar, graph NoSQL databases) and do not consider security at the modeling stages [6, 5, 10].

There are relevant contributions concerning a complete secure development of information systems. Although they do not focus specifically on Big Data and their specific security problems, they present interesting ideas, such as proposals to include security constraints from the earliest development stages, to extend UML with security aspects, to use the model-driven approach, etc. Some of the most relevant proposals are described below. TROPOS is a methodology for software development based on the intentional goals of agents that provides an extension called Secure TROPOS [2]. Mokum [11], which is an active object-oriented knowledge-based system for modelling, permits the specification of security and integrity constraints, along with automatic code generation. UMLsec [4] defines and evaluates security specifications using formal semantics (labels, stereotypes, etc.). It focuses mainly on accessing control policies and in the specification

of confidentiality and integrity requirements. Model-driven Security (MDS) [1] applies the model-driven approach to include security properties in high-level system models and to automatically generate secure system architectures.

3 Framework proposed

This section describes the framework proposed in this paper. As Figure 1 shows, this framework is composed of three stages. First, we carry out an analysis of the data set in order to automatically detect and tag sensitive data by applying NLP and ontological techniques. Then, designer use that information as recommendations to model the structural and security aspects of the database. In this stage, designer decides which NoSQL database technology to use (document, columnar or graph) and uses the metamodel proposed. Finally, that model is implemented into a specific tool according to the mappings defined in our methodology.

In this paper we have defined the path marked in Figure 1 (data set analysis, modelling for document NoSQL database technology and implementation into MongoDB). The remainder parts of the figure shows how we plan to expand our architecture.

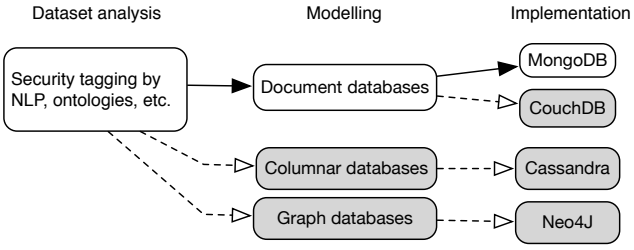

Fig. 1. Framework proposed.

3.1 Data-Set analysis

Firstly, starting from a data set (for instance in a CSV format), each field is analysed searching for sensitive information.

One of the contributions of this proposal is the establishment of the security privileges needed to access each field of the data set. It is carried out by using NLP and lexical and ontological resources. In our framework we have used the lexical resource WordNet 3.1[1]. By analysing the values of each field we establish two kinds of security constraints.

[1] http://wordnetweb.princeton.edu/perl/webwn (visited on February, 2017).

1. Security constraints. There are fields in which all the information is sensible at the same security level, that is, it does not depend of their specific values. For instance, an address field is sensible and a certain security level (for instance, SL=2) could be required for queries. But this level is the same for all the address, there are not address more sensible than others.
2. Fine-grain security constraints. Nevertheless, there are special cases in which are needed to define fine-grain security constraints to establish higher security privileges for certain values of the field. For instance, to query a field representing diseases could require an specific security level in general (for instance, SL=2), but for certain values that represent terminal diseases could require a higher security level (for instance, SL=3). In this way, an user with security level of 2 solely could see diseases not related with terminal diseases.

All the information obtained in this stage will be used in the next stage as recommendations for the designer to model the data set.

3.2 Modelling

In this step, designer models the data set according to a metamodel. In this paper we have defined a metamodel focused on a kind of NoSQL databases, document databases (Figure 2).

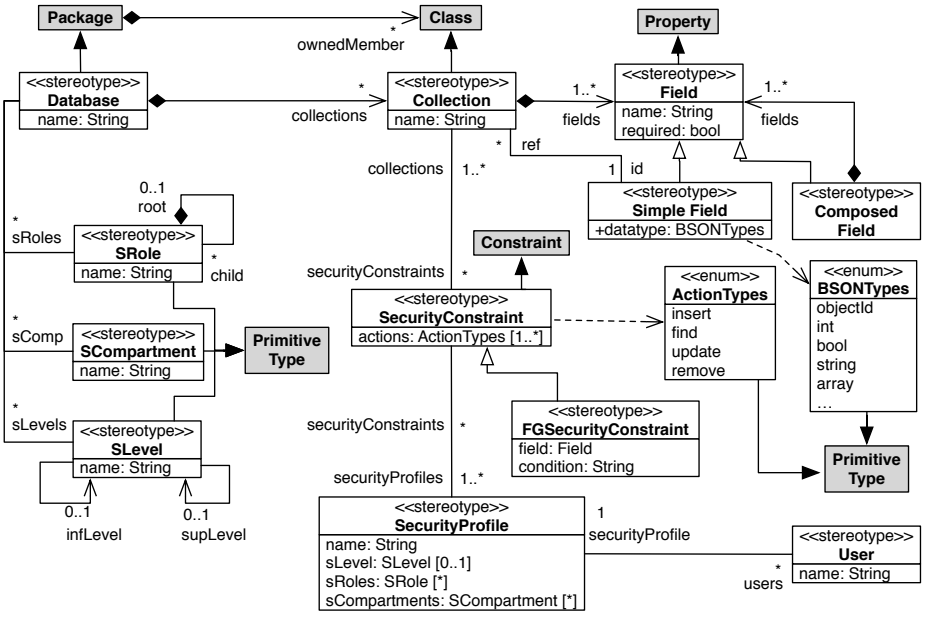

Fig. 2. Metamodel for secure document databases.

This metamodel extends UML class diagram for allowing the specification of both structural and security aspects related with document databases. It permits to model structural aspects such as Databases (as Packages), Collections (as Classes) and Fields (as Properties): that can be simple fields (with a BSON data type) or fields composed of several simple or composed fields.

The security configuration of the system which we want to model is defined by using three points of view: a hierarchical structure of Security Roles (SRole); a list of Security Levels (SLevel) with the clearance levels of the users; and a set of horizontal Security Compartments or groups (SCompartment). Once this configuration has been established, sets of certain security configurations composed of roles, levels and compartments can be defined as instances of security profiles (SecurityProfile) that after that, are associated to system Users.

Once security configuration has been established, we can define security rules (SecurityConstraint) associated with structural elements. Each rule indicates the actions (insert, find, update and remove) that certain subjects (SecurityProfile) can carry out over certain objects (Collections). Furthermore, we can define fine grain security rules (FGSecurityConstraint) which affect to specific fields of a collection. This kind of rules allow us to establish different security privileges (SecurityProfile) when the values of a field satisfy a condition.

3.3 Implementation

The last stage is the implementation of the modelled database into a specific document database management tool (such as MongoDB, CouchDB, etc.). Our model represents all the concepts needed for its implementation in different tools, but we have to define how to map the model of each destination tool. Once we have defined these mappings, this process can be integrated in a Model Driven Engineering approach and automated by defining a set of model-to-text transformations for each destination tool.

In this paper, MongoDB has been considered as the destination tool and the mapping necessary to obtain a secure implementation from model has been defined. This mapping is composed of three stages and will be further detailed in case study (Section 4):

1. **Database structure.** First, we generate scripts to create de collections and fields defined, considering their datatypes and structural constraints such as mandatory fields. In MongoDB we use the commands "createCollection" and "validator" for datatypes and constraints.
2. **Security configuration.** The majority of database tools uses a Role Based Access Control (RBAC) policy. Since our model is richer and includes roles, levels and compartments, we have to implement that information by using role hierarchies. The security constraints defined in the model are implemented attached to the roles definition as privileges over certain objects (collections). It is implemented in MongoDB by defining "role" with "privileges" over "resources" and "actions".

3. **Fine-grain security constraints.** This kind of constraints have to analyse the values returned by a query in order to hide the most sensible ones. These constraints used to be implemented by adding middleware to control user queries. MongoDB uses a trusted middleware that processes the aggregation pipeline with query result and allows us to define aggregation rules that checks conditions and users privileges in order to hide sensitive information. We use "redact" predicates showing (descend) or hiding (prune) information according to the evaluation of a condition.

4 Case study

4.1 Description

The data set used in this experimentation is a synopsis that represents the clinical care at 130 hospitals between the years 1999 and 2008. This data set is available in the UCI Machine Learning Repository [3].

4.2 Data Set analysis

At this stage we will work with the data set previously defined in CSV format. This phase of analysis is reusable, regardless of the technology (for instance, document, columnar or graph databases) used in the following phases.

In our example, considering our data model, we analyse the different fields in order to establish the security constraints and fine-grain constraints defined in section 3.1.

Following we present an example of each type of security constraint. We assume that we analyse the field treatment. In advance, we ignore what is the kind of information stored, but we know the values. After consulting on WordNet, each of the terms of the field is determined that refers to the concept "medicament" (each particular word is searched, its hyperonyms and hyponyms). Previously the security levels of the key concepts of information which are to be handled have been defined within the ontology. In our example, "medicament" is assigned with a level 2 of security. Therefore, we define a security constraint for the treatment field by setting a security level 2. In Figure 3 a fragment of the ontologies extracted from the fields treatment and medical_specialty is shown.

It is noteworthy that the NLP techniques are suitable to be applied on text fields. After undertaking a lexical-morphological analysis of the text (POS tagging) and a partial syntactic analysis (partial parsing) the main concepts of the text are detected. In addition, these techniques can solve own problems of language, such as ellipses, anaphoric references, ambiguities, etc. present in the text. On these key concepts extracted, the process defined above is carried out by assigning a security level to a text field.

In order to show the fine-grain constraints, we analyse the field medical_specialty. This field identifies a specialty of the admitting physician corresponding to 84 distinct values, for example , "cardiology", "pediatrics", "neurology", "oncology", etc. These values are searched in WordNet and it is determined that they

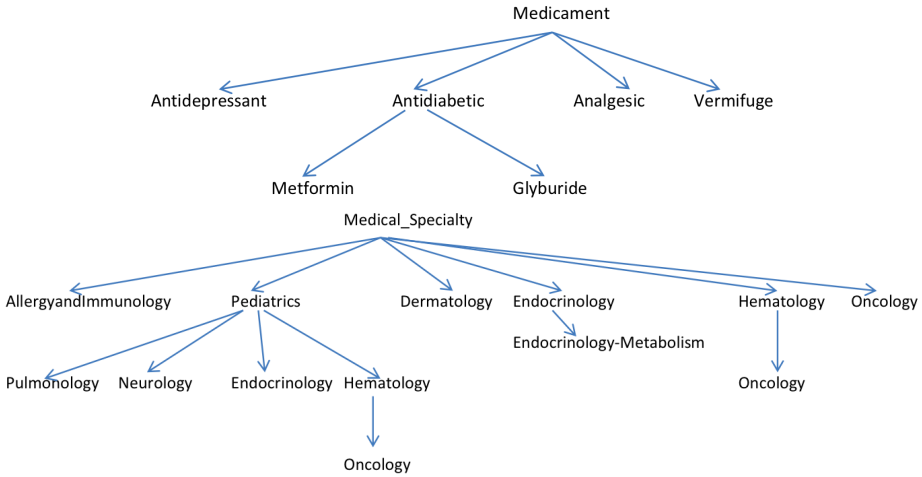

Fig. 3. A fragment of treatment and specialty ontologies.

are concepts related to "medical specialty", which is sensitive information. In our example, "medical specialty" is assigned with a level of security of 2. When analysing the data, we can appreciate that some values belong to the specialty of "oncology" which is more sensitive (level 3). Therefore it is marked to be taken into account and creating a new rule of this type into the model (Figure 3).

The final result of the analysis, after applying the constraints set, is as follows:

Patient fields. Race: Level 3; Address: Level 2; Remaining fields: Level 1.

Admission related fields. Admission type and treatment: Level 2; Medical specialty: Level 2 (and if it is oncology: Level 3); Diagnosis: Level 2; Remaining fields: Level 1.

4.3 Modelling

Figure 4 shows the model designed for our case study. It has been modelled according to the metamodel defined in this paper (Figure 2).

Fields from the original data set have been grouped by two documents represented as two collections: Patient collection, for patient id, name, race, etc. and Admission collection, for information related with encounters, diagnosis and treatments. In this example we have used the two kinds of references between documents: (1) a normalized reference between the patient id in Patient and Admission collections; and (2) a denormalized reference with embedded data of treatment (pairs of medicaments and doses) in Admission collection.

We consider a security configuration composed of roles for administration and health (SRole Admin and Health). We grant access to Patient collection for Admin role and access to Admission collection for Health role.

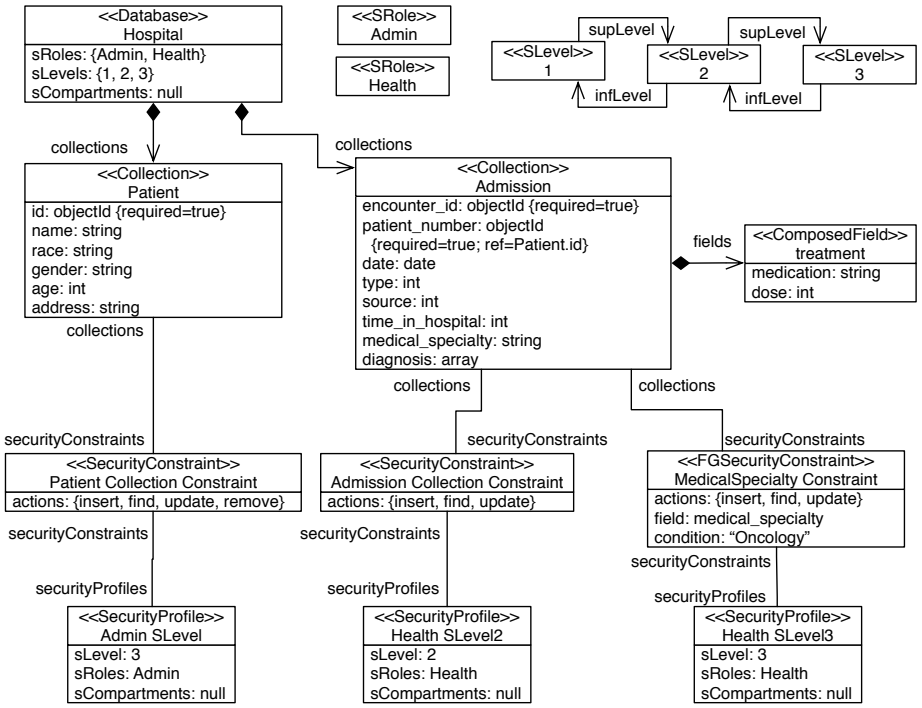

Fig. 4. Case study: model.

For establishing security constraints we take into account the security recommendations obtained in the previous analysis. Then, we also define three security levels: 1, 2 and 3, being level 3 the most restrictive.

In this point, each field of the data set has been tagged with a security level. We analyse these tags to define the security level of the collection as the higher value of their fields. Then, we define a security constraint associated to the collection and the corresponding security profile. For instance, fields of Admission collection require a security level of 1 unless type, diagnosis, medical specialty and treatment (level 2). Then, we establish the security level of the Patient collection as the most restrictive one (level 2). This is represented as a SecurityConstraint (named “Admission Collection Constraint”) attached to Patient collection and to a Security Profile (named “Health SLevel2”) that indicates the security level of 2 and the security role Health. This rule indicates in its actions property that the actions granted are insert, find and update.

Additionally, we can define fine-grain security constraints to increase the security privileges needed to access certain values of a field. For instance, the field “medical specialty” of Admission collection was tagged as level 2. Nevertheless, some values of this field are related with “oncology” and require a higher security level (level 3). This rule has been modelled as an additional security constraint

(named "MedicalSpecialty Constraint") attached with the collection Admission and with a security profile (named "Health SLevel3") that indicates a Health role and a security level of 3. This rule includes a property "condition" in which is indicated that this rule is applied if medical specialty is "oncology".

4.4 Implementation

This section shows how the system previously modelled could be implemented into an specific document database management tool, MongoDB. In Figure 5 the DB structure, the documents, the security configuration, and the fine-grain security configuration can be seen.

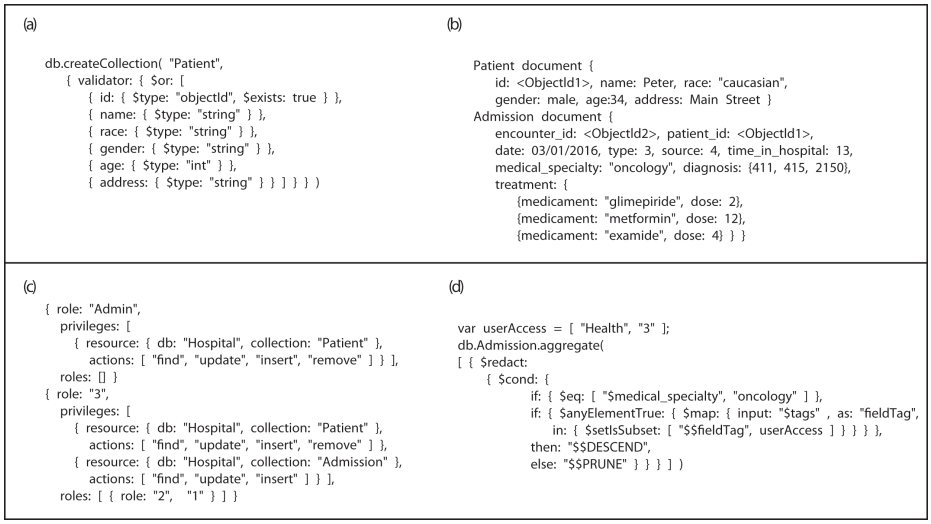

Fig. 5. Case study: implementation.

Database structure. The structural part of the database is implemented by defining collections and their fields, indicating datatypes and restrictions. Documents of our collections Patient and Admission are shown in Figure 5.

Security configuration. Since MongoDB use a role based access control (RBAC) mechanism, the security configuration (roles, levels and compartments) has to be established by defining roles and their attached security privileges (actions granted over certain objects). The figure shows and example for the roles "Admin" and "3" (that represent the security level 3). They are granted to do certain actions over the collection Patient.

Fine-grain security constraint can be implemented in MongoDB by using the aggregation pipeline and "redact" predicates. These results answer user

queries passing through trusted middleware. This predicates allows showing (descend) or hiding (prune) information according to the evaluation of a condition. An example in which oncological medical specialties are shown to roles "Health" and security level of 3.

5 Conclusions

The large amount of data handled by Big Data technologies can expose sensitive enterprise or personal information if it is not correctly assured (for unauthorized accesses, data aggregation, etc.). Current Big Data technologies, such as NoSQL document, columnar or graph databases, do not allow to adequately incorporate security constraints.

In this paper, we have presented a framework for modelling and implementing secure NoSQL document databases. The framework is composed of three stages: (1) analysis of the data set in order to automatically detect and tag sensible data by applying NLP and ontological techniques. (2) Modelling the structural and security aspects of the database; in this stage, the NoSQL database technology to use (document, columnar or graph) is defined and the metamodel proposed in our framework for document databases is used. (3) Implementation of the model into a specific tool according to the mappings defined in our framework. Here, we have considered MongoDB implementation.

Finally, a case study is presented with a data set of medical domain in which the analysis of the data is displayed to establish different levels of security, modeling and its concrete implementation in MongoDB.

As future work, we want to automate our process for document databases by (1) implementing the mappings from model to implementation with model-to-text (M2T) transformations and (2) extracting database model from the data set to use it as base model, that will be modified by the designer in order to include security rules. Furthermore, we plan to extend our framework to columnar and graph NoSQL database technologies.

References

1. D. Basin, J. Doser, and T. Lodderstedt. Model driven security: from uml models to access control infrastructures. *ACM Transactions on Software Engineering and Methodology*, 15(1):39–91, 2006.
2. L Compagna, P.E Khoury, A Krausová, F Massacci, and N Zannone. How to integrate legal requirements into a requirements engineering methodology for the development of security and privacy patterns. *Artif. Intell. Law*, 17(1):1–30, 2009.
3. Andrew Frank and Arthur Asuncion. Uci machine learning repository [http://archive. ics. uci. edu/ml]. irvine, ca: University of california. *School of Information and Computer Science*, 213, 2010.
4. J Jurjens and H Schmidt. Umlsec4uml2 - adopting umlsec to support uml2. Technical report, Technical Reports in Computer Science. Technische Universitat Dortmund, http://hdl.handle.net/2003/27602, 2011.

5. Nir Kshetri. Big data's impact on privacy, security and consumer welfare. *Telecommunications Policy*, 38(11):1134–1145, 2014.
6. Katina Michael and Keith Miller. Big data: New opportunities and new challenges [guest editors' introduction]. *Computer*, 46(6):22–24, 2013.
7. RENCI/NCDS. Security and privacy in the era of big data. Report, 2014.
8. B Saraladevi, N Pazhaniraja, P Victer Paul, MS Saleem Basha, and P Dhavachelvan. Big data and hadoop-a study in security perspective. *Procedia Computer Science*, 50:596–601, 2015.
9. Bhavani Thuraisingham. Big data security and privacy, 2015.
10. Raghav Toshniwal, Kanishka Ghosh Dastidar, and Asoke Nath. Big data security issues and challenges. *International Journal of Innovative Research in Advanced Engineering (IJIRAE)*, 2(2):15–20, 2015.
11. R.P. van de Riet. Twenty-five years of mokum: For 25 years of data and knowledge engineering: Correctness by design in relation to mde and correct protocols in cyberspace. *Data & Knowledge Engineering*, 67(2):293–329, 2008.