A New Approach for ZCM Models Metamodelling based on Model Driven Architecture Concepts

Abdelaziz Mamouni[1*], Meriem Atifi[1], Younes Raki[2], Mohammed Oqaidi[3], Abdelaziz Marzak[4]
[1*,1,2,3,4] Faculty of Sciences Ben M'Sik, Hassan II University of Casablanca
Casablanca, Morocco
mamouni.abdelaziz@gmail.com
meryematif@gmail.com
raki.youness@gmail.com
moqaidi@gmail.com

ABSTRACT: *In this paper a new approach for Zakat Calculation Models (ZCM) metamodelling was proposed, using Model Driven Architecture (MDA) concepts. Also this paper was highlighted the three interesting contributions of our previous work in the context of the present paper especially for providing a better understanding of the proposal viz; the ZCM-based process called ZCMGProcess, the MOF-based metamodel called ZC2M, and the Metamodel-based tool called ZCMGenerator. This tool has been used to demonstrate the feasibility of the proposed approach and to facilitate its adoption. The approach adopted in this work was inspired on idea of Architecture Driven Modernization (ADM) and respects the MDA's Four-Tier architecture.*

Keywords: MDA, Zakat, ZC2M, ZCM, ADM, MOF, XML, ZCP, Metamodel

Received: 1 May 2017, Revised 14 June 2017, Accepted 25 June 2017

1. Introduction

Zakat was established by the Quran and the Sunna in the year II of the Hegira in Mecca. The purpose of this financial institution is to establish the charity and the spirit of solidarity within the Islamic society. Despite the fundamental role played this pillar in the religious prescriptions of Muslims, Zakat is neglected and largely ignored. Indeed, there are many who do not perform their Zakat properly and do not know neither how to calculate it nor those to whom it should be paid. In fact, this is mainly due to the multitude of Islamic schools of the jurisprudence. This plurality of schools mixed with the diversity and the complexity of the Zakatable wealth implies a great number of conditions, rules and methods, which makes the understanding and the calculating of Zakat more complex. In this sense and in order to mitigate this problem, several ZCP (Zakat Calculation Platform) platforms have been emerged in the last few years, each with its advantages and disadvantages. This panoply of platforms has the advantage of making the process of calculating Zakat extremely simple and user-friendly. However, these solutions are based on

different technologies and their features are often very limited. To overcome these limitations, this paper aims to resolve the issues related to the different existing platforms using the techniques of model driven engineering while satisfying the requirements and the expectations of the users, regardless of their school of belonging.

In this paper, we propose a new approach for ZCP platforms based on Model Driven Architecture (MDA) and Architecture Driven Modernization (ADM). The fruits of this approach have resulted in three major contributions. A first main contribution of this research is the proposal of a new model-based process called ZCMGProcess for the generation of the ZCM (Zakat Calculation Model) models adaptable to the new technologies. These models are extensible, reusable, portable, and can be communicated to any platform without taking into account technical specifications. A second contribution resides in proposing a metamodel called ZC2M (Zakat Calculation Meta-Model) containing a minimum set of concepts such as category, sub-category, wealth, Nissab and other specific concepts of the Zakat domain. It is the first metamodel of its kind to tackle the Zakat calculation problem. This meta-model is based on the MOF (Meta-Object Facility) and aims to define the key concepts used for modelling the ZCM models and to modernize the existing ZCP platforms. A third contribution resides in the development of a tool supporting our process called ZCMGenerator. It is implemented to carry out the application of the ZCMGProcess, in particular, to support automatic generation of the ZCM models. This tool has been used to demonstrate the feasibility of the proposed process and to facilitate its adoption.

The rest of this paper is organized as follows: In section II, we describe the proposed process. Section III present the proposed metamodel. Section IV describe and presents the proposed tool. Section V discuss and presents the results of our approach. Finally, Section VI concludes our work with references at the end.

2. ZCMG Process

In order to rationalize the modelling and generation of ZCM models, we have defined a new process called ZCMGProcess which allows us to guide the modelling of the models based on the requirements of the users. The objective is to define such a process based on the basic concepts of the MDA and ADM approaches. The architecture of the proposed process is divided into three main phases[1]. The first phase uses reverse engineering to generate the General PIM. The activity of this first phase has a single task, which carries out the PSM„³PIM transformation, in which the PSM models are evolved to the General PIM model, which represent similar features of the platforms studied. For this, we integrate the common features of all PSM models in a common model. The general independent platform model obtained describes all the business classes, their attributes and operations and their relationships in a UML class diagram. This model is not only allows us to visualize the static view of the ZCP platforms but it is also allows us to clearly shows the concepts required to understand the Zakat calculation and to defines all the elements needed to fulfil the ZCP platforms development. The second phase of our process uses restructuring engineering to enrich the General PIM model obtained at the end of the first phase. For this, we introduce the new requirements into the General PIM model by changing their structure. The result of this phase is a new improved model that we call ZC2M. For this purpose, the General PIM model obtained from the PSM models is restricted manually into another adapted model taking into account the features of the target ZCP platform which will meet better needs of their users. We define this new model by introducing the new requirements by changing their structure to meet the needs of the Zakat calculation as much as possible, taking into account the diversity of schools of jurisprudence and the variety of Zakatable wealth. The third activity in our process is the activity that eventually uses the obtainable improved model from the second phase for generating the ZCM models in the form of XML files. This third phase is consists of two main steps viz; the first step consists of modelling the new model according to our ZC2M meta-model based on the specified requirements and needs expressed by the users. The result of this first step is a PSM model representing a school of jurisprudence. In the second step, this PSM model is transformed to a model in the XML file format. This task is automated by using a supporting tool called ZCMGenerator[2] that we have implemented to carry out the application of the ZCMGProcess, in particular, to support automatic generation of ZCM models.

3. ZC2M Metamodel

In the context of MDA, a metamodel is expressed using the metametamodel MOF, which provides the language for creating metamodels, whereas UML defines the language for creating models corresponding to specific metamodels. ADM, in the context of MDA, creates the model of the source code of a platform for the purpose of importing it into an MDA-enabled development environment[3]. In order to provide a new metamodel for Zakat domain based on the MDA and ADM concepts, we have created and used the models related to existing ZCP platforms viz; Z2MWA (Zakat Maal Mobile Web Application)[4], PERISA (Personal Islamic Asset Management System)[5], M2A (Muslim Android Application)[6], and finally, MAAZS (Muslim

Android Application for Zakat Selangor)[7] platform. Each of these models is designed by using UML class diagram. We describe below the four PSM models representing the above ZCP platforms.

• The PSM model specific to the Z2MWA platform corresponds to a class diagram UML, in which each fundamental concept is represented using a class and each existing relation between concepts using an association. It contains three classes, User, Asset Data and Zakat Calculator. The User class has a method named addUser, whose purpose is to allow the addition of new users. The Zakat Calculator class has a method called calculateZakat to allow users to calculate the amount of Zakat that should be issued, in the form of Zakat per unit amount of the type of property and the calculation of the total amount of all assets owned. The Asset Data class has four methods, addAsset, deleteAsset, editAsset and viewZakatInformation, for allowing users to manage their Zakat information. It also contains an association named Access, which has the User class as the source and the Asset Data class as the target. Finally, it contains an aggregation relation between the Asset Data class and the Zakat Calculator class, where the latter is composed of one or more Asset Data classes.

• The PSM model representing the platform PERISA is similar to the Z2MWA platform model, it contains in addition to three classes, User, Zakat and Zakat Calculator, a fourth class named Zakat Counter which has an aggregation relationship with the Zakat class, where the Zakat Counter class is made up of one or more Zakat class, and includes two methods, searchCounter to allow registered users to search Zakat counter, and viewCounter to allow registered users to view the nearest Zakat counter;

• The model representing the M2A platform is similar to the PERISA platform model, it contains in addition to four classes, User, Zakat, Zakat Calculator and Zakat Counter, a fifth class named Skim Berkat class that allows users to calculate their monthly Zakat deduction, is made up of one or more Zakat class and includes one method named useSkimBerkat to allow user to have the ability to calculate their monthly Zakat deduction;

• The PSM model representing the platform MAAZS is similar to the M2A platform model, it contains in addition to five classes, User, Zakat, Zakat Calculator, Skim Berkat and Zakat Counter, a sixth class named Kifayah calculator class that allows users to calculate their Kifayah, it has an association relationship with the User class, and has two methods, displayKifayahInfo to display detail information about kifayah information including its limit, who eligible to received and pay Zakat, and calculateKifayah method to calculate kifayah and to decide whether in Maskeen or Fuqafaa category or not qualified to received Zakat.

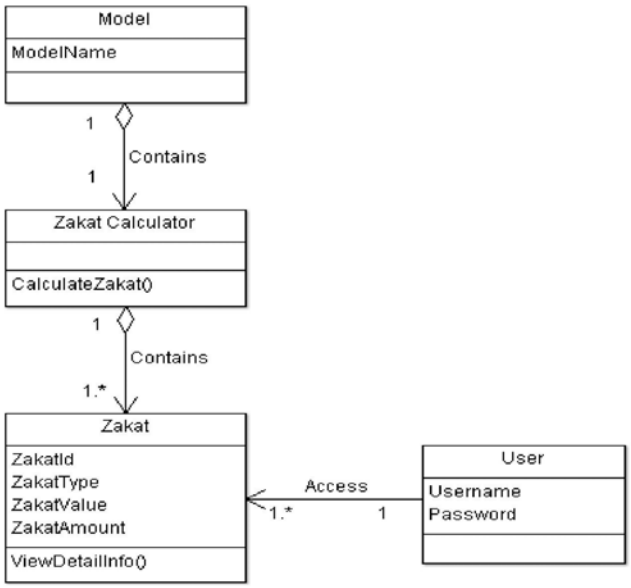

Figure 1. General PIM Model

These four models have been revisited and revised by identifying shared features and common functionality between their classes. Using these PSM models and from the comparison of their structures, we have proposed the General PIM model, which is given in the Figure 1. This model shows that all the PSM models share a common structure in the manner in which they

calculate the amount owed to be paid. Basically, these models consider two main classes, Zakat calculator to allow users to calculate their Zakat, and Zakat Information to allow users to view Zakat information, where the class “Zakat calculator” is made up of one or more “Zakat Information”. The general PIM model above is obtained by integrating the common features of all the PSM models in a common model. It includes all the common business classes, their attributes and operations and their relationships in a UML class diagram. This model can be expanded to cover all schools of jurisprudence and all categories of wealth. For this, we have provided a new improved model that will meet better needs of the ZCP platforms users. For the sake of providing a comprehensive metamodels for Zakat domain, we have used the concepts which are generally accepted by all schools of jurisprudence. The proposed metamodel (Fig. 2) is given after an in-depth study of the Zakat domain[8][9] and a detailed survey of the existing ZCP platforms[10]. It is divided into hierarchical elements and its structure is divided into three levels viz; model which represents a school of jurisprudence, category which can be divided into several sub-categories, and wealth. It has been defined by bearing in mind its use by developers who wish to design a new ZCP platform. It is considered the cornerstone of our tool, and will then be the basis used to properly instantiate the ZCM models in XML files in the next section.

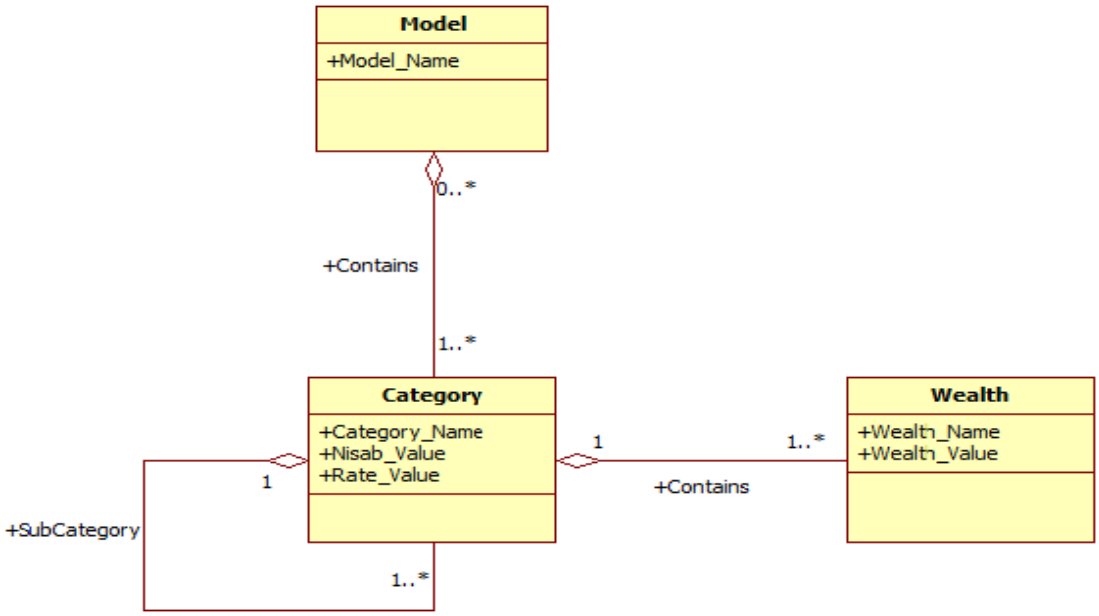

Figure 2. ZC2M Metamodel

4. ZCM Models Generation

4.1 Framework ZCM Generator

The aforementioned ZCMGProcess was inspired on idea of ADM [11] approach and is supported by a tool named ZCMGenerator that follows the model driven architecture paradigm for modelling and generating the ZCM models. The Framework ZCMGenerator was developed using the most advanced languages such as JEE and XML in web-based environment and was designed to be used in several technological platforms and to allow the reuse, portability and scalability of ZCM models. It makes the reuse of these models possible by describing with a metamodel the elements and their relations constituting such model. Thus, the portability and extensibility of these models are made possible by the ability to instantiate them in a format based on XML technology. In [12], we described the architecture technique, the features and the capability of this tool to enhance the portability and reusability of existing ZCP platforms. The technology used during ZCMGenerator development takes into account the possibility to add new schools of jurisprudence and new Zakatable wealth, while increasing the ZCMGenerator performance. To promote high-performance and high scalability of this framework, the design pattern strategy and XML technology were used. The ZCMGenerator is a web-based platform so that it can be easily accessible via Internet anywhere and anytime. English language has been chosen as the language to make it globally understood. It is user friendly, easy to use for various backgrounds of users, interactive and attractive as well as informative, applicable to professionals and also normal users, and equipped with detailed information on all categories of Zakat and all schools of jurisprudence. The developers can easily use this tool, consisting of predefined concepts relevant to Zakat domain, to select school of jurisprudence, categories,

sub- categories, and then assigns each value to the appropriate wealth based on the context of their particular case.

4.2 Case study

In this paper, we have chosen Maliki School and the Zakat on Livestock and Income to illustrate our approach. For generating the Livestock and Income model according to Maliki law, the users start modelling by choosing an Islamic school of their belonging, according to their choice, ZCMGenerator will help them to construct the remaining elements of their model. The

```xml
<?xml version="1.0" encoding="UTF-8"?>
<School name="Maliki">
  <Category name="Livestock">
    <SubCategory name="Sheep_Goats" Nissab ="40" Rate="1'>
      <Wealth name="Sheep"/>
        <value> <value/>
      <Wealth/>
      <Wealth name="Goats">
        <value> <value/>
      <Wealth/>
    </SubCategory>
    <SubCategory name="Camels" Nissab ="5" Rate="1">
      <Wealth name="Camels">
        <value> <value/>
      <Wealth/>
    </SubCategory>
    <SubCategory name="Cows_buffaloes" Nissab ="30" Rate="1">
      <Wealth name="Cows">
        <value> <value/>
      <Wealth/>
      <Wealth name="buffaloes">
        <value> <value/>
      <Wealth/>
    </SubCategory>
  </Category>
  <Category name="Income">
    <SubCategory name="Salaries" Nissab ="85g d'or" Rate="2.5%">
      <Wealth name="Yearly Gross Income">
        <value> <value/>
      <Wealth/>
      <Wealth name="Yearly Expenses">
        <value> <value/>
      <Wealth/>
    </SubCategory>
    <SubCategory name="Professionals" Nissab ="85g d'or" Rate="2.5%">
      <Wealth name="Professional Fees">
        <value> <value/>
      <Wealth/>
      <Wealth name="Yearly Expenses">
        <value> <value/>
      <Wealth/>
    </SubCategory>
  </Category>
</School>
```

Figure 3. The generated ZCM Model in XML Format

modelling of this model is carried out by respecting our ZC2M metamodel. The new model generated is strictly in accordance with the SCW (School-Category-Wealth) structure. This process makes it possible to transform the needs expressed by the users into a model. Each defined specification is associated with an element of the SCW structure. At the technical level, due to the importance of the XML technology in the field of software engineering, ZCMGenerator instantiates the new model in this format, which could be easily used to generate and communicate models between all the stakeholders that are responsible for the construction of ZCP platforms without considering the diversity of technologies. Fig. 3 below shows an example of the XML file generated by our framework, the model generated corresponds to the case of Zakat on Livestock and Income according to Maliki school law. For the validation of the results of the generation of ZCM models in XML format, we have used external tool that positively tested the conformity of these models to the standards of W3C (World Wide Web Consortium).

5. ZCM Model Metamodelling

Metamodelling is the modelling of models. In their most common use, metamodels describe the permitted structure to which models must adhere[13]. Metamodelling as a technology provides significant power to users, and it has been thoroughly explored in terms of modelling data, software, and languages. The layered approach to metamodelling is depicted in Fig. 4 below. The M0 layer represents the real system. A model represents this system at level M1 conforms to its metamodel defined at level M2. The metamodel itself conforms to the meta-metamodel at level M3. This meta-metamodel conforms to itself [14].

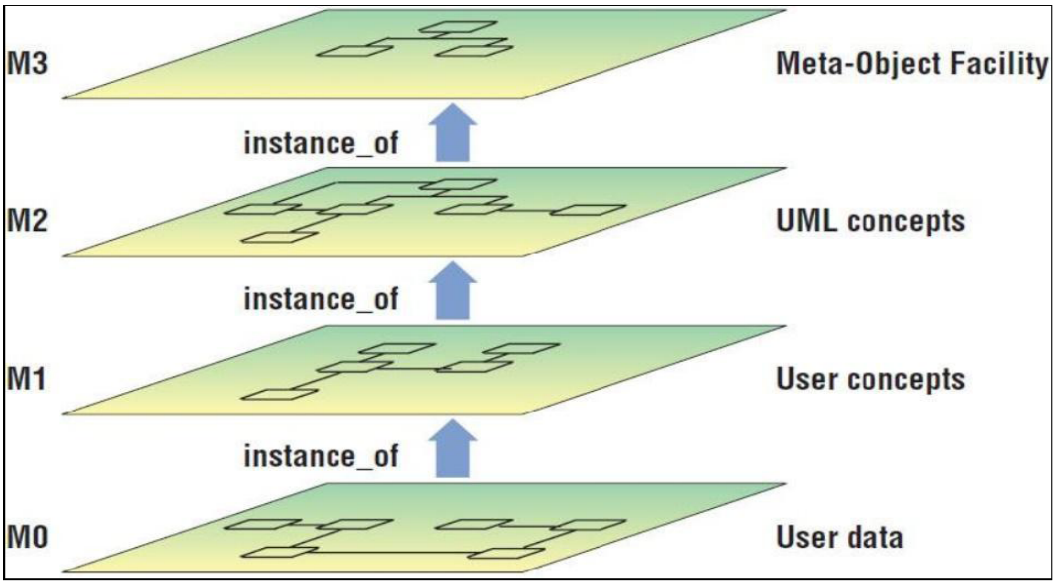

Figure 4. Traditional OMG's metamodelling infrastructure[14]

In the four-layer approach, artefacts in each layer conform to, and/or are abstracted by the more abstract layer adjacent. Thus, semantic artefacts in the M3 layer are abstracted in models from the M2 layer, which in turn conform to the metamodels from the M1 layer. As these layers of abstraction are traversed, the role of each abstraction layer changes. The purpose of metamodelling through MOF is to describe the models in these various layers using common modelling abstractions [13].

The approach adopted in this work respects the MDA's Four-Tier architecture that we have detailed above. Indeed, the models specific to existing ZCP platforms are assimilated to PSMs and belong to level M1. Similarly, the general model is assimilated to a PIM model and also belongs to the M1 level. The metamodel resulting from the definition of these PSMs and PIM in MOF belongs to the level M2. This metamodel helps the Stakeholder to express their needs in a model specific to their visions according to different schools (Hanafi, Shafii, Hanbali and Maliki), so that these models can be instantiated to ensure exchanges between all platforms and technologies, without taking into account technical specifications.

The models resulting from the instantiation of these models belong to level M0. Figure 5. below illustrates the four-level architecture applied to ZCM models.

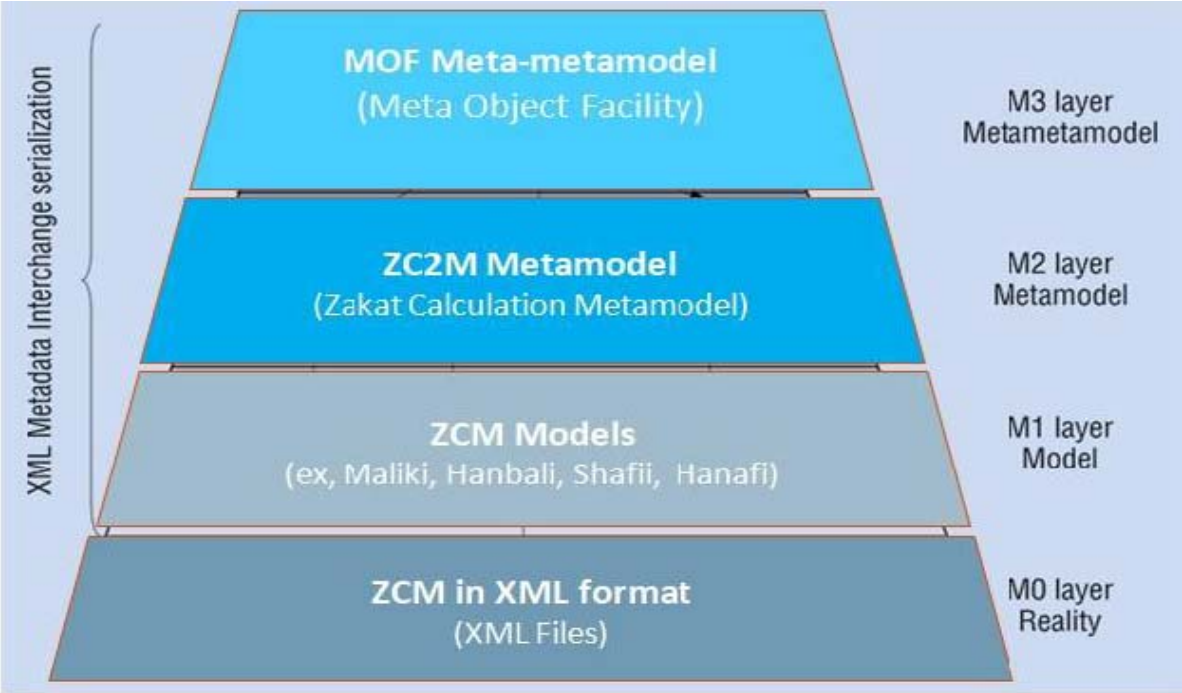

Figure 5. ZCM Models Metamodelling

6. Conclusion

This paper intends to provide an approach for ZCM models metamodelling using model driven architecture concepts and aims at guiding designers in the modelling of extensible, reusable, and portable models which can be communicated to any platform without taking into account technical specifications. To make the construction of these ZCM models feasible by using our approach, a modelling tool named ZCMGenerator was developed. This tool allows the users to design ZCM models according to our ZC2M metamodel and to generate them in form of XML files. Our tool uses XML technology, which could be easily, used to generate, and communicate models between all the stakeholders that is responsible for the construction of ZCP platforms without considering the diversity of technologies and platforms.

The approach adopted in this work respects the MDA's Four-Tier architecture, the models specific to existing ZCP platforms are assimilated to PSMs and belong to level M1. Similarly, the general model is assimilated to a PIM model and also belongs to the M1 level. The metamodel resulting from the definition of these PSMs and PIM in MOF belongs to the level M2. This metamodel helps the Stakeholder to express their needs in a model specific to their visions according to different schools (Hanafi, Shafii, Hanbali and Maliki), so that these models can be instantiated to ensure exchanges between all platforms and technologies, without taking into account technical specifications. The models resulting from the instantiation of these models belong to level M0.

References

[1] Ahmad, N. A. N., Akhbariee, N. I., Hafizuddeen, M. (2013). Requirements analysis of android application using activity theory: A case study. *In*: *International Conference of Information and Communication Technology*, *p*. 145–149.

[2] Akhbariee, N. I. (2013). Muslim Android Application for Zakat Selangor (MAAZS) Kuala Lumpur, Malysia.

[3] Fenty, E.M.A., Hulliyah, K., Ekafitri, M. (2014). Applying mobile application development life cycle in the development of Zakat Maal mobile web application using JQuery mobile framework. *In*: I*nternational Conference on Cyber and IT Service Management*. *p*. 89–92.

[4] Force, O.A.T. (2012). Why do we need standards for the modernization of existing systems.

[5] Génova, G. (2005). What is a metamodel, the OMG's metamodeling infrastructure. *Softw. Syst. Model*. 4. *p*. 171–188.

[6] Mamouni, A., Marzak, A. (2017). ZCP Modernization by Recovering ZCM Models from Existing Platforms. *In:* International Conference on Compute and Data Analysis, (New York, NY, USA: ACM). p. 252–256.

[7] Mamouni, A., Marzak, A. (2017). ZCP Platforms: Survey, Issues and Challenges. *Int. J. Sci. Eng. Res*. 8. p. 481-489.

[8] Mamouni, A., Marzak, A., AlHaddad, Z., Boukouchi, Y. (2016). ZCPGenerator framework: Zakat calculation platforms generator based on MDA approach. *In*: 2nd International Conference on Cloud Computing Technologies and Applications (CloudTech). p. 185–190.

[9] Mamouni, A., Marzak, A., Belangour, A., Azouazi, M., Youness, R., Haddad, Z. A. (2017). ZCMGenerator: Generation of ZCM Models from ZC2M Metamodel Based on MDA and ADM Approaches. *Trans. Mach. Learn. Artif. Intell*, V (5).

[10] Mamouni, A., Marzak, A., Belangour, A., Alhaddad, Z. (2015). A meta-model for automation of the deduction of judgments relating to Zakat. *Int. J. Comput. Trends Technol*. 30. *p*. 1-10

[11] Mamouni, A., Marzak, A., AlHaddad, Z. (2015). Vers une déduction automatique dans l'Islam: Cas de la zakat. *Int. J. Comput. Trends Technol*. 24. p. 29-35.

[12] Nor, A.H. (2007). personal islamic asset management system using object-oriented approach.

[13] Object Management Group, http://www.omg.org

[14] Sprinkle, J., Rumpe, B., Vangheluwe, H., Karsai, G.: Metamodelling. Model-Based Eng. *Embed. Real-Time Syst*. (2011). 3. p. 57–76.