SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM
TMLAI TRANSACTIONS ON MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE
VOLUME 5 NO 4, 2017
ISSN 2054-7390

ZCMGenerator: Generation of ZCM Models from ZC2M Metamodel Based on MDA and ADM Approaches

Abdelaziz Mamouni, Abdelaziz Marzak, Abdessamad Belangour, Mohamed Azouazi, Raki Youness, Zayed Al haddad
Faculty of Sciences Ben M'sik, Laboratory of Information Technology and Modeling, Casablanca, Morocco, Hassan II Universty of Casablanca, UH2C
Cdt Driss El Harti, B.P 7955, Sidi Othman, Casablanca-Morocco
mamouni.abdelaziz@gmail.com, marzak@hotmail.com, belangour@gmail.com, azouazii@gmail.com, raki.youness@gmail.com, alhaddadtri@gmail.com

ABSTRACT

Zakat Calculation Models (ZCM) consist of sequences of specific concepts of the Zakat domain such as category, sub-category, wealth and rate. The content of a ZCM model depends on the school of jurisprudence and the type of Zakatable wealth. Correspondingly, creating a ZCM model involves four steps (analyzing, modelling, developing and generating), the last step consists of generating ZCM model in a structured XML format in accordance with our ZC2M (Zakat Calculation Meta-Model) metamodel which we have already proposed in our previous work. This paper proposes ZCMGenerator, a metamodel-based framework for modelling and generating of the ZCM models adaptable to the new technologies and specific to each school of jurisprudence. These models are extensible, reusable, portable, and can be communicated to any platform without taking into account technical specifications. This paper also presents some cases study scenarios emphasizing the benefits enabled by the proposed framework and a detailed comparison between ZCMGenerator and some existing ZCP (Zakat Calculation Platforms) platforms. Experimental results show that, compared with traditional ZCP platforms, ZCMGenerator is more scalable and high-performance.

Keywords-component; ADM; MDA; ZCM; ZC2M; Zakat; MOF; Metamodel

1 Introduction

Zakat calculation platform or ZCP for short is a platform created to support the understanding and calculating of Zakat. These platforms are becoming an important tool and play a crucial role in Muslims life. Over the past few years, several ZCP have been developed due to the high demands on such Islamic platforms. One of the key aims of these platforms is to support Muslims in calculating their Zakat easily, correctly, quickly and accurately. Actually, they have made the entire process of calculating Zakat extremely simple and straightforward. However, this multitude of platforms, although it has advantages, it also presents a number of disadvantages, for instance, the vast majority of them were developed in traditional manner, most of them have lack of portability, reusability and interoperability. Moreover, they were developed to run solely on one technological platform. Furthermore, they are inflexible and difficult to maintain and change. In order to go some way to overcome these problems, the ZCMGenerator

DOI: 10.14738/tmlai.54.3188
Publication Date: 15th August, 2017
URL: http://dx.doi.org/10.14738/tmlai.54.3188

framework has been developed to fulfil the Zakat calculation requirements by facilitating ZCM models modelling and ensuring reuse and portability. To develop and implement this framework, we have already proposed in [1] a new ADM-based process named ZCMGProcess. The architecture of this process is depicted in Figure 1 which is divided into three main phases. The first phase consists in understanding, describing and comparing the ZCP models structure, where the goal is to generate the General PIM. The second phase aims to enrich this PIM model, where the goal is to generate a new improved model. The third phase aims to generate the ZCM models in XML files. These phases are in accordance with the three stages defined by the ADM (Architecture Driven Modernization) approach that are reverse engineering, restructuring, and forward engineering.

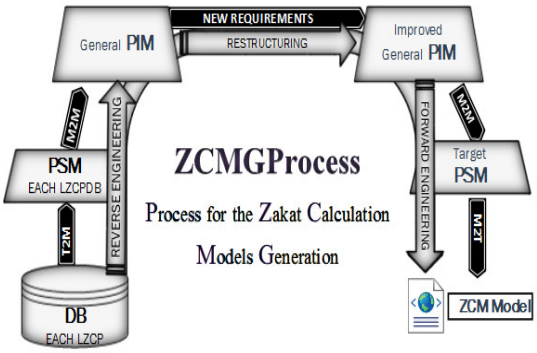

Figure 1. Architecture of ZCMGProcess

The aforementioned ZCMGProcess was inspired on idea of ADM approach. It aims at guiding designers in the modelling of extensible, reusable and portable Zakat calculation models using MDA (Model Driven Architecture) concepts. It not only takes advantage of the MDA approach but also integrates other important aspect of portability, interoperability and reusability that is XML technology. Indeed, the XML brings a number of powerful capabilities to information modelling such as heterogeneity, extensibility, flexibility, internationalization and many others [2]. This process has been defined by bearing in mind its use by developers who wish to design a new ZCP platform and to produce the extensible, reusable and portable models which can be communicated to any platform without taking into account technical specifications.

Then, we have proposed in [3] a new metamodel called ZC2M (Zakat Calculation Meta-Model) (Fig. 2) which is a generic metamodel enabling, among other things, to take into account the specific calculation rules of each school of jurisprudence when determining the Zakat amount. For the sake of providing a comprehensive metamodel, we have used the concepts, which are generally accepted by all schools of jurisprudence. It is given after an in-depth study of the Zakat domain [4] and a survey of the existing ZCP platforms. It 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 platforms. It is the first metamodel of its kind to tackle the Zakat calculation problem. Its structure is divided into hierarchical elements viz; Model, Category and Wealth, whose categories can be divided into several sub-categories.

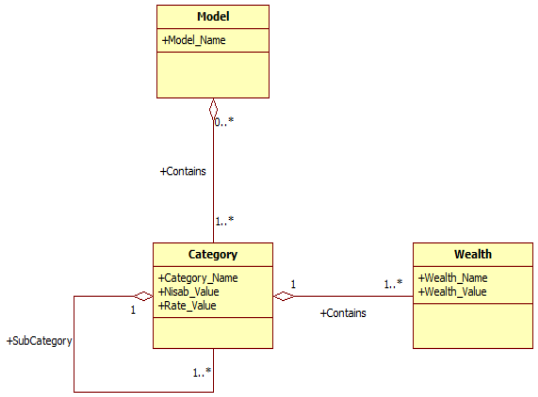

Figure 2. ZC2M Metamodel

Finally, in [5] we have described the architecture technique, the features and the capability of ZCMGenerator to enhance the portability and reusability of existing platforms. This framework uses XML technology, which could be easily used to generate and communicate ZCM models between all the stakeholders whose are responsible for developing the ZCP platforms without considering the diversity of technological platforms. Figure 3 below describes the four steps involved in this framework.

- Analyzing: in this step, we define and describe the requirements of the users;
- Modelling: the previous requirements are transformed into a model under the following structure: schools, categories, sub-categories and wealth;
- Developing: this step consists of developing the Zakat Calculation Functions (ZCF), which allow associating the values of wealth in the XML files in order to Zakat calculation;
- Generating: this step allows generating the ZCM models in XML format in accordance with our ZC2M metamodel. It allows also generating the ZCR reports using the generated ZCM models.

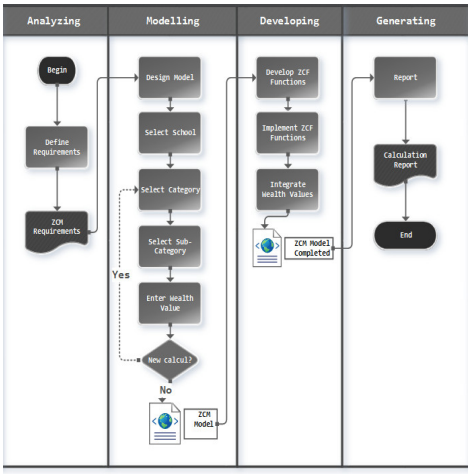

Figure 3. ZCMGenerator business processes

This paper discusses the design, implementation and performance of ZCMGenerator framework. This later is a metamodel based tool for modelling and generating of ZCM models based on the requirements expressed by users. In fact, ZCMGenerator is more than one tool of generation of ZCM models, it offers a

lot of functionality to calculate the amount owed to pay, to generate the ZCR reports and to send these reports and models by email. Another key concept of ZCMGenerator is the strong association between ZCF functions and the generated XML files. It permits through the mechanisms of transformation to define and generate the ZCM models in XML format and the ZCR reports in XHTM format. It integrates the XML technology in order to ensure the portability, reusability and interoperability.

The rest of this paper is organized as follows. Section II describes detailed analysis of various existing ZCP platforms. Section III describes in details functional and technical characteristics, business processes and development process of the ZCMGenerator. Section IV presents discussion and comparative study. Finally, section V discusses our conclusions and future work.

2 Related Work

During the past few years there has been a remarkable evolution in the world of ZCP platforms. These later are a great source for understanding and calculating Zakat. However, most of them are still having their weaknesses and limitations. Nowadays, the design and the development of a ZCP platform impose new requirements as a result of the diversity of technologies with specifics characteristic. In this context, several approaches have been proposed and used to design and develop such platform.

As technology evolves, several approaches have been proposed and multiple ZCP have been developed. In [6], a platform that calculates the Zakat for mobile phone users with a GPRS connection has been developed using J2ME technology. The IZAMD platform is based on client-server approach. The server side is just various web pages to get the prices of gold, silver and stocks. The mobile device side allows setting the Zakat configuration. This platform covers all school of jurisprudence. However, it is limited to the specific categories of wealth.

Al-Riyami et al. [7] have applied expert system technology in the field of Zakat to assist Muslims in the decision making of identifying the rules of making Zakat and to assist in complex calculations. The ZES platform was developed using a freeware rule-based shell called eXpertise2Go. In this platform, the user has the freedom of choice, either one type or as much as he desires to calculate in an understandable and clear manner. However, it does not state the source of the information.

Mohammad et al. [8] have also developed an expert system for the domain of Zakat. The ESs platform consists of two main components, the knowledge base uses facts to represent knowledge and the inference engine executes rules upon these facts to provide responses. This platform is very helpful in calculating Zakat. However, it cannot answer to questions of all Muslims belonging to different Islamic schools.

Harun et al. [9] have proposed an ontology-based approach for developing the Zakat management system. The proposed ontology has been implemented in OWL and modelled with the Protégé tool. It was designed for interoperability of systems and will make the process of developing the Zakat management system faster. However, it only covers two main processes that are collection and distribution.

Abdul Hamid & Kasirun [10] have been developed Personal Islamic Asset Management System (PERISA) using object-oriented approach by implementing Rational Unified Process (RUP) model. This platform is user friendly and easy to use for various backgrounds of users. However, its features can still be improved by integrating online payment, registration and online financial report.

Fenty et al. [11] have applied mobile application development life cycle approach in the development of Z2MWA platform using JQuery Mobile Framework for mobile users. This platform can be accessed by any platform and delivers speed, stability and an excellent cross-browser experience for web mobile visitor. However, it is limited to one school of jurisprudence and does not cover all categories of wealth.

Noorul et al. [12] have developed Muslim Android Application for Zakat Selangor (MAAZS) using agile development approach, RUP model, PHP and JQuery language for Android mobile platform. This platform is useful for Muslims users as it combine the Zakat information, Zakat calculation, Skim Berkat which is Zakat monthly deduction, Fidyah and Kifayah calculation in one application. However, it is developed especially for Selangor residents.

Ahmad et al. [13] have proposed the requirements analysis approach using the activity theory to develop a Muslim Android Application (M2A) basing on Android platform. This platform is equipped with Zakat calculator, information about Zakat, Skim Berkat and a special feature to find the nearest Zakat counter. However, it is limited to one school of jurisprudence.

Imam & Usman [14] have developed SICZ platform using Waterfall model. This later is done in a systematic and sequential approach starting from system level requirements and then headed to the stage of the analysis, design, coding, and implementation. This platform was developed to help the community in general and in particular the users of Android-based Smartphone in term of calculating Zakat. However, it does not cover all categories of wealth.

Atunnisa et al. [15] have developed ABACZ platform using prototype method. This later is an attractive idea for complicated and large systems for which there is no manual process or existing system to help determining the requirements. However, it is usually not complete and many of the details are not built in it. This platform can present the calculation of Zakat Al-Mal with easy to use and practical. However, it does not cover all types of wealth and all schools of jurisprudence.

We have now covered the main characteristics, limitations and advantages of the existing ZCP platforms. All these platforms are very useful and helpful to perform property calculations. However, our analysis shows that although multiple ZCP using different approaches have already been proposed which help in calculating the amount owed to pay. These ZCP do not provide complete requirements of Zakat calculation. For example, the most of these platforms are limited to one school of jurisprudence and are poorly implemented with limited calculation categories. Moreover, the vast majority of them do not meet the new requirements that have emerged in the field of software engineering such as portability, reusability and interoperability. Also, this analysis shows that there are two main categories for such platforms viz; desktop-based and web-based platforms and nine main groups of ZCP development approaches. Like any new platform, the adoption of these platforms by Muslims is not free from issues. To fulfil the needs of current ZCP issues and to overcome the aforementioned drawbacks a new modernized ZCP must be designed to fulfil the Zakat calculation requirements.

3 Framework and Implementation

3.1 Functional and technical characteristics

The framework ZCMGenerator is the first and innovative web framework dedicated to the Zakat domain for the modeling and generation of ZCM models, it is developed using the most advanced languages such

as JEE and XML in web-based environment. It is designed to be used in several technological platforms and to allow the reuse, portability and scalability of ZCM models. This framework makes the reuse of these models possible by describing with a metamodel the elements and their relations constituting a ZCM 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. As the first web framework dedicated to the Zakat domain, ZCMGenerator makes it very simple to generate simple ZCM models, while providing a high-level framework for more complex ZCM models. The technical features and benefits of this framework are many and varied viz;

A user-friendly interface that makes it easy for users to design their ZCM models, this interface is designed using Tiles technology, which is a framework designed to easily allow the creation of web application pages with a consistent look and feel;

Compliance with standards, models generated in XML format conform to W3C (World Wide Web Consortium) standards, these files are verified and tested by W3C accredited validators;

A point of communication with other platforms, the technology of ZCMGenerator is based on the use of XML files and especially in the generation of ZCM models, to guarantee a channel of communication and cooperation with other platforms;

It is a web application developed using the JEE and the XML technology in order to be available to a maximum number of users and to be consultable with any Web-browser. It can be accessed from computer, Tablet, Smartphone or any other device; and so on.

3.2 ZCMGenerator business processes

A business process is the combination of a set of activities with a structure describing their logical order. Its objective is to produce a desired result [16]. It can be visualized as a flowchart or as a process matrix of a sequence of activities. The business process of ZCMGenerator framework involves three important processes.

3.2.1

Process of modelling and generating of ZCM Models: The modelling of the ZCM models is carried out by respecting our ZC2M metamodel. The new models generated are 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. In this process, 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 (categories, Sub-categories and wealth). At the technical level, due to the importance of the XML technology in the field of software engineering, ZCMGenerator instantiates the new models in this format.

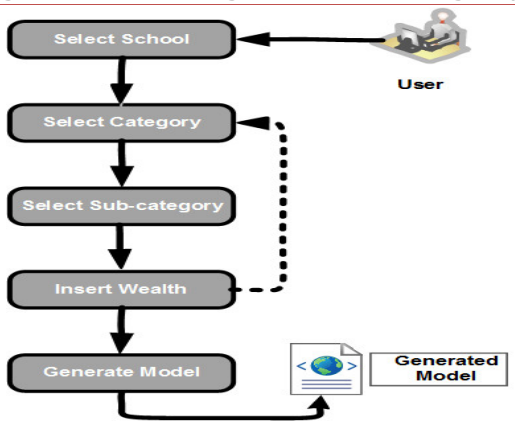

Figure 4. Process of modelling and generating of ZCM models

3.2.2

Process of calculation and generating ZCR reports: Zakat calculation is done through the generated models. To this end, the ZCF functions affect the values to Zakatable wealth after their implementation in the environment of the development. Thus, ZCMGenerator permits a numerical representation of the characteristics composing these models. Indeed, the reports generation process allows extracting directly from the generated models all the desired information and to export them in open formats.

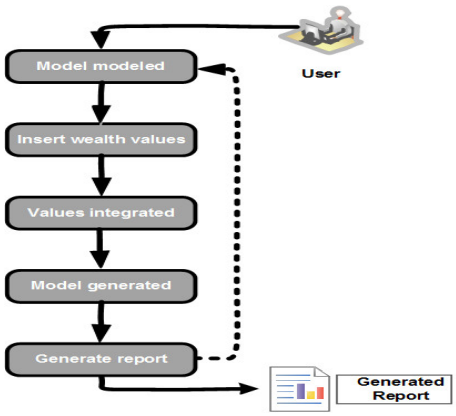

Figure 5. Process of calculation and generating ZCR reports

The generated reports are developed based on the XSLT (eXtensible Stylesheet Language Transformations) technology. Indeed, W3C proposed the XSLT standard to develop transformations of XML documents. The approach defined in this standard allows specifying transformations of XML documents using Template. This later consists of an applicability clause and a set of instructions. The operating principle of XSLT is to browse a given XML document at the input of a transformation in order to detect which templates can be applied to it. When a Template is detected as being applicable, the set of instructions that it defines is executed.

3.2.3

Process of sending and receiving e-mail notifications: The process of sending notifications allows users to send the reports and models generated by e-mail with the objective of allowing the reuse of these models

in future projects. To this end, we used the e-mail management standard of JEE technology, JavaMail API. This later allows sending and receiving e-mail and handling messages. Its purpose is to be easy to use and to provide flexibility that allows it to evolve and to remain as independent as possible of the protocols used. To send or receive messages, JavaMail uses different protocols such as SMTP, IMAP or POP. Simple Mail Transfer Protocol (SMTP) is the standard messaging protocol. The Post Office Protocol (POP) allows retrieving mail from a remote server. Internet Message Access Protocol (IMAP) is an alternative to POP offering more possibilities such as concurrent access management, multiple mailboxes and more. In our case, we have used the SMTP protocol due to its simplicity and efficiency.

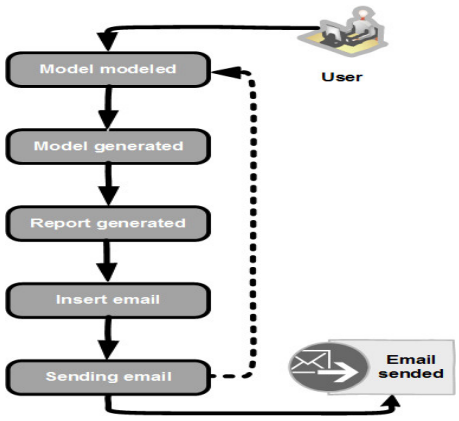

Figure 6. Process of sending email notifications

3.3 ZCMGenerator development process

The development process of our ZCMGenerator framework consists of four main phases viz; domain analysis, architectural design, implementation and validation phase. We describe these phases in detail below.

3.3.1

Domain analysis: In this first phase, we have developed a solid knowledge of Zakat domain. To this end, we first tried to identify the commonalities points of view and differences among Islamic schools exist on this subject. Then, we studied the different ZCP platforms to get an overview of what is proposed. Through a comparative analysis, we have been able to describe in detail the important characteristics, advantages and limitations of them. Subsequently, after analyzing these platforms, we have been able to define and design a specific model (PSM) corresponding to each platform. After analyzing the structures of these PSM models and from the comparison between their structures, we have proposed a general model (PIM). This later is obtained by integrating the common features of all PSM models in a common model. The general model obtained describes all the business classes, their attributes and operations and their relationships in a UML class diagram. It allows us to generate all existing ZCP models according to the requirements of the designer. In order to propose a new architecture and new features for ZCP platforms, we have enriched this general model. The result of this work is a new improved PIM model that we call ZC2M metamodel. This later is a high level abstraction metamodel which we used to create a database in the XML files format allowing the representation of the knowledge of this domain. The ZC2M metamodel is a sufficiently generic metamodel to model the structure of the ZCM models and their relationships.

3.3.2

Architectural design: The architecture of software describes its major components, their relationships and how they interact with each other. It serves as a blueprint for software and provides an abstraction to manage the software complexity and to establish a communication and coordination mechanism among components. The primary goal of the architecture is to identify requirements that affect the structure of the software [17]. In such architecture, we have implemented the MVC 2 (Model-View-Controller) architecture using the Struts 2 framework, which is a standard for developing well-architected web applications based on the MVC design paradigm [18]. In MVC2 architecture there is only one controller that receives all requests for the application and is responsible for taking appropriate action in response to each request. Then, we have identified the components and their collaboration that we need to implement our framework. Subsequently, we have determined the internal and external components necessary to ensure the proper functioning of this framework, among these important components we mention the following main APIs.

JDOM (Java Document Object Model), API for reading, writing, and manipulating XML from within Java code;

JavaMail, API for sending and receiving email via SMTP, POP3 and IMAP protocols;

XSLT: API for transforming XML documents into other documents;

JUnit: API for writing and running unit tests; and so on.

The ZCMGenerator is a set of software components structured and cooperated with each other to define the basic functions of this framework. Our development approach ensures reuse, portability, scalability, extensibility and more.

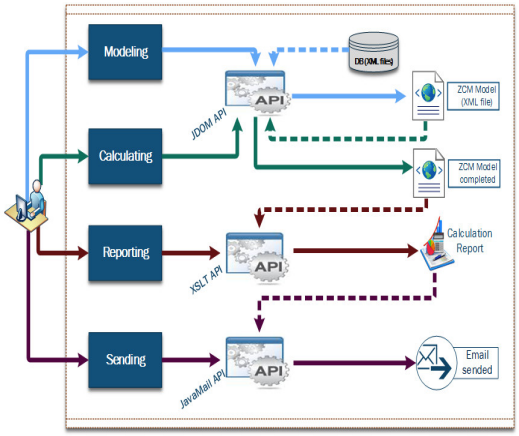

Figure 7. Technical architecture of ZCMGenerator

3.3.3

Implementation: In this phase, we have developed each element of the software architecture and implemented the basic functionalities of our framework. We present in the following sub-sections the graphical interfaces representing these functionalities.

Creation of a new ZCM model:. The ZCMGenerator allows users, using its user-friendly interfaces, to create their own ZCM models, through these interfaces, users can choose their school of belonging,

categories, subcategories and wealth, according to the SCW (School-Category-Wealth) structure which is inevitably required by the respect of our ZC2M metamodel. Figure 8 below allows users to create a specific model corresponding to the Maliki School and to Livestock Zakat. To do this, the user should choose the Maliki School, the category "Livestock", the sub-categories "Camels, Cattle and Goat-Sheep" and, thereafter, the wealth corresponding to each selected sub-category appears.

Figure 8 (1) corresponds to the interface of the Islamic schools;

Figure 8 (2) corresponds to the interface of categories of wealth;

Figure 8 (3) corresponds to the interface of the sub-categories;

Figure 8 (4) corresponds to the interface of the wealth;

Figure 8 (5) allows to user to add other wealth.

After the user has defined the ZCM model corresponding to the Maliki School and to Livestock Zakat, he has also the possibility to extend this model by adding as much wealth as he wants to calculate in a clear and comprehensible way. For instance, the user can extend the generated model by adding Zakat on Income. For this end, he should therefore choose the category Income, the sub-category (Salaries, professionals, etc.) and in the stage of the wealth corresponding to each sub-category, he can enter the corresponding values in each property field if he wants to calculate the Zakat after the generation of his model. By clicking on the "Download your Zakat Calculation Model" button (Fig. 9), the ZCMGenerator allows the automatic generation of this model in XML format (Fig. 10).

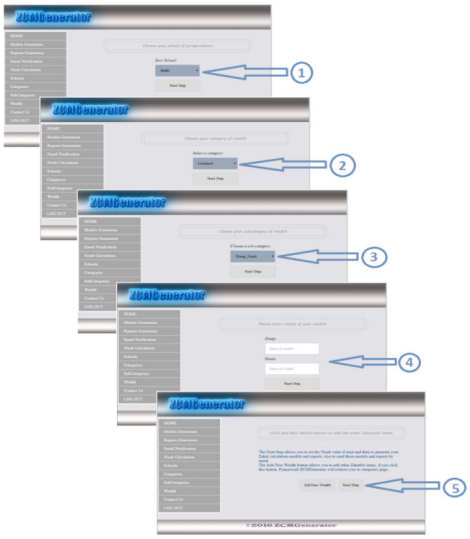

Figure 8. ZCM model modeling

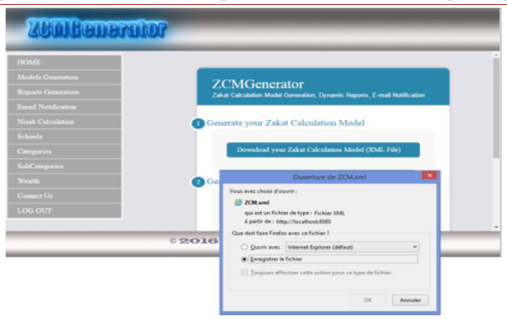

Figure 9 Instantiation of ZCM model in XML format

```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>
</School>
```

Figure 10. The generated ZCM Model

Calculation and generation of ZCR reports: In the previous sub-section, we have shown that our framework allows users, using its user-friendly interfaces, to create their own ZCM models. In this second sub-section, we see how to calculate the Zakat from these models. To do this, in the wealth step, the users have to enter the values corresponding to the fields in the form above (Figure 8 (4)) and as explained in the previous sub-section, they have the possibility to add as much wealth as they want to calculate, for example, in addition to the Zakat on Livestock, they can add Zakat on Income. To this end, in the wealth step corresponding to the Income category, users have to enter the values corresponding to each field. After the user has entered the data requested by our framework, by clicking the "Download your Zakat Calculation Report" button (Fig. 11), the ZCMGenerator allows the automatic generation of ZCR report in XHTML format (Fig. 12).

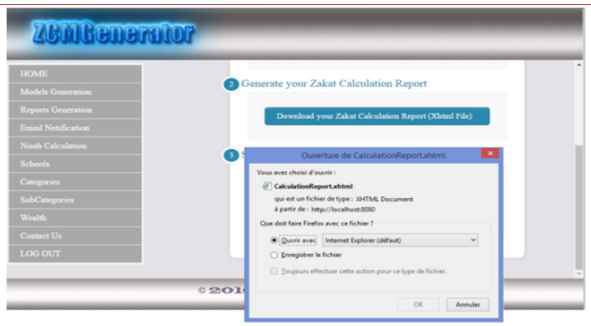

Figure 11 Generating ZCR report

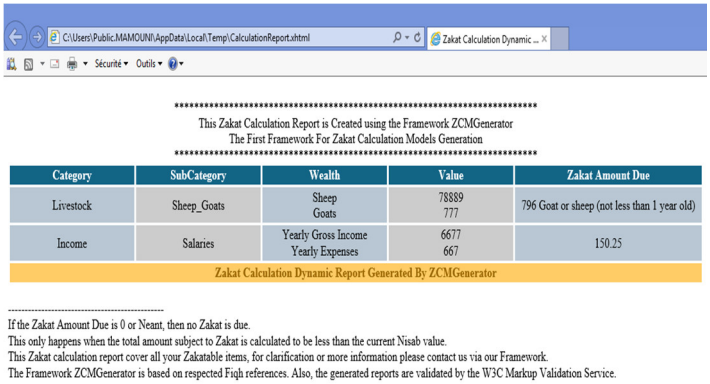

Figure 12. The generated ZCR report

Sending ZCR reports by email: In the two previous sub-sections, we have seen how to create ZCM models, how to calculate the Zakat from these models and how to generate the ZCR reports. In this sub-section we will see how to send these reports by email. To do this, simply enter a recipient email, for example, “mamouni.abdelaziz@gmail.com” and then click on the "Send E-mail" button (Fig. 13). After clicking this later button, the user receives an email containing his ZCR report. Below, we present the result of sending this report by email (Fig. 14).

Figure 13. Sending ZCR reports by email

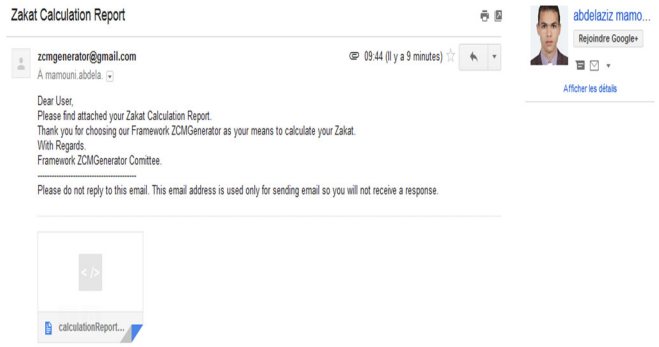

Figure 14 Message received by the user

3.4 Validation of results

In this step we have verified all the business functions for which ZCMGenerator has been designed. To do this, we have created a database in the form of XML files based on our ZC2M metamodel and on the actual data. Subsequently, we have created an XSD schema corresponding to these XML documents using a tool proposed by XMLGrid.net. Then, for the validation of the results of the generation of ZCM models in XML format, as well as the generation of the ZCR reports in XHTM format, we have used external tools that positively tested the conformity of these generated files to the standards of W3C, we present below the tools that we have used to validate our results.

The XmlGrid.net Generator service provides an XSD schema generation service from an XML document. Our XML documents check result was well formed by this service (Fig. 15) and the generated XML schema is presented as an XML file format (Fig. 16).

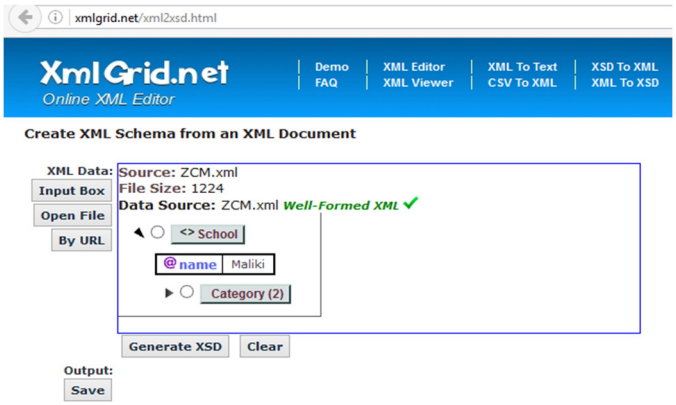

Figure 15. Result of XMLGrid Generator

Figure 16. The generated XSD schema

The Freeformatter Validator provides a validation service to validate an XML file against a specified XSD schema. The result of the generation of ZCM model in XML format was validated by this validator (Fig. 17).

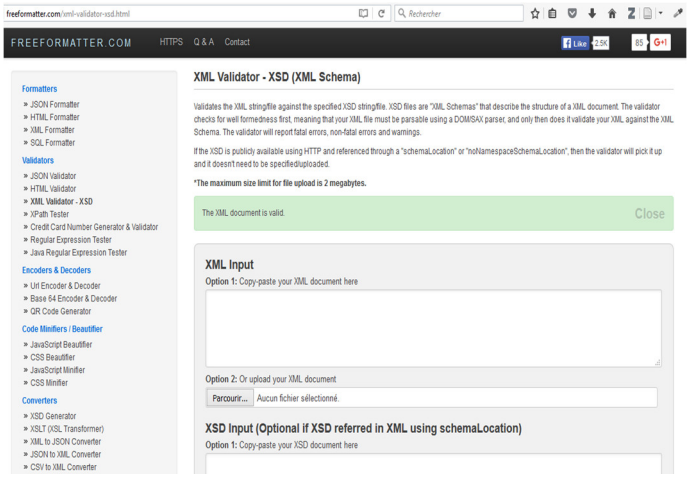

Figure 17. Result of Freeformatter Validator

The W3C validator presents an XHTM document validation service. The result of the generation of ZCR report in XHTML format was well validated by this validator (Fig. 18).

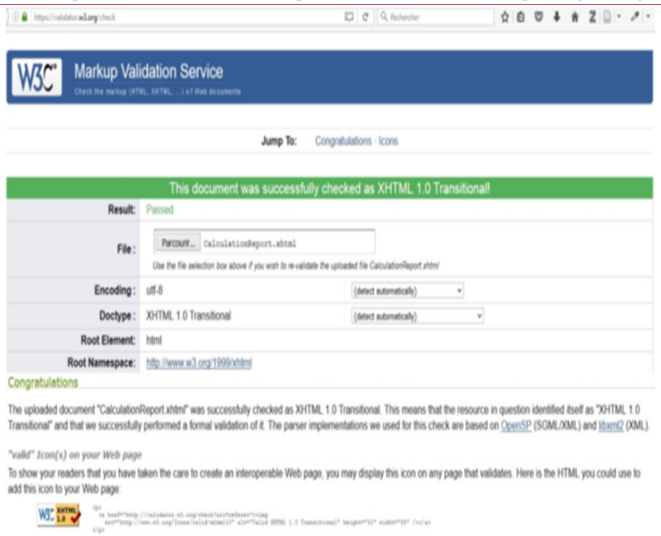

Figure 18. Result of W3C XHTML Validator

4 Discussion and Comparative Study

In the second section, we have analyzed the different existing ZCP platforms and covered the main characteristics, limitations and advantages of these platforms. This analysis shows that although multiple ZCP using different approaches have already been proposed, they do not provide complete requirements of Zakat calculation, for instance, the most of them are limited to one school of jurisprudence and are poorly implemented with limited calculation categories. Moreover, the vast majority of them do not meet the new requirements that have emerged in the field of software engineering. Furthermore, they do not meet the needs of almost all Muslims living in modern societies. In this forth section we compare the characteristics of our framework with those of the existing ZCP and we discuss the reasons for which the framework proposed presents the best results. This comparison is presented in the form of a table (Table I) and a graph (Fig. 19), the table lists the characteristics of each ZCP while the graph shows the frequency of the characteristics in each ZCP. These frequencies are ranked in descending order.

Table 1 Comparative study: ZCMGenerator Vs Existing ZCP Platforms

Criteria	ZCP Platforms											
	ZZMWA	*IZAMD*	*ZES*	*ZCE*	*ZMS*	*ESs*	*PERISA*	*SICZ*	*MAAZS*	*ABACZ*	*M2A*	*ZCMGenerator*
ZCM Modeling												X
ZCM Generating												X
ZCR Generating												X

ZCR Sending	X											X
Cover all wealth												X
Cover all schools		X										X
Full & partial modes	X	X	X		X							X
Nissab calculation	X								X	X		X
Ease of use	X	X	X	X	X	X	X	X	X	X	X	X
Multi-platforms	X						X					X
Zakat information	X	X	X	X	X	X	X	X	X	X	X	X
Meet the new requirements of software engineering												X

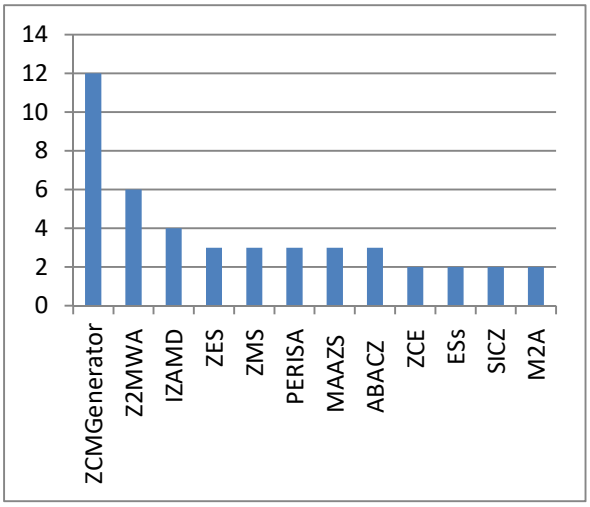

Figure 19. A comparison of the characteristics of ZCP platforms

As shown in the Table I above, there is no platform to allow modelling and generation of ZCM models while enabling the users to calculate their Zakat on any Zakatable wealth according to the principles of their jurisprudence school, whereas our framework is able to overcome these limitations by providing features to support the Muslims in calculating their Zakat according to the rules recognized by their school of belonging, whatever the desired Zakatable wealth. Thus, it allows generating models adaptable to the recent technologies and specific to each of these schools. More specifically, the Table I and Figure 19 above show the qualities of the proposed framework compared to existing platforms. From these results,

we deduce that: ZCMGenerator is the first framework allowing the modelling and the generation of ZCM models adaptable to the new technologies and specific to each school of jurisprudence. These models are extensible, reusable, portable and can be communicated to all technological platforms regardless of technical specifications; It allows the automatic generation of ZCR reports in the form of XHTML files, while the other solutions only allow to display the calculation result on their user interfaces; It not only has the ability to generate the ZCR reports in the XHTM format and the ZCM models in the XML format but also the ability to send these reports and models by email in order to allow their reusability in other projects; It is designed to work on multiple platforms, ZCMGenerator is a web application developed to be accessed by any hardware containing a web browser; It enables Zakat to be calculated on any Zakatable wealth regardless of the school of jurisprudence, whereas most of the existing platforms are designed for certain types of wealth and for a specific school; It offers two modes of calculation, the full mode calculates everything about Zakat, whereas the partial mode allows calculating the Zakat on a particular type of wealth, also, the user has the freedom of choice, either one type or as much as he desires to calculate in an understandable and clear manner; It meets the new requirements that have emerged in the field of software engineering such as portability, reusability and interoperability. Indeed, the proposed framework not only takes advantage of the MDA approach but also integrates other important aspect of portability, interoperability and reusability that is XML technology. The XML brings a number of powerful capabilities to information modelling such as heterogeneity, extensibility, flexibility, internationalization and many others. Then, by automating of the generation of ZCM models in this format, our tool addresses many issues such as reusability, portability, interoperability, flexibility and so on; 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. Actually, for implementing the ZCF functions, we have used the strategy pattern. To this end, we have created a strategy interface defining an action, context class which uses a strategy and concrete strategy classes implementing the strategy interface, whereas, for implementing these functions, the existing platforms use multiple if-else statements, which allow a choice to be made between two possible alternatives. This approach is not recommended since it does not make it possible to handle more data while reducing the time required for handling more user requests, while the strategy is a better design pattern to handle such a case, which replaces if-else statements with a polymorphic call. It enables an algorithm's behaviour to be selected at runtime. The experimental results indicate that this pattern has significantly reduced the amount of time required to calculate the amount owed to pay than the calculation by using multiple if-else statements. Thus, as we have already seen above, the ZC2M metamodel was used to create a database in the XML files format allowing the representation of the knowledge of the Zakat domain. Each XML file conforms to the same XSD schema and depends on the specific school. We have created several XML files in order to allow handling the small amounts of data, when files are small it is much quicker and easier to parse, whereas, the existing platforms store all of their data in a database which needs to be read in completely to update any one field. Our framework has then the capability to handle a growing amount of data and the potential to be enlarged in order to accommodate that growth.

5 Conclusion

This paper provides the first and innovative web framework dedicated to the Zakat domain for the modelling and generation of ZCM models, ZCMGenerator was developed using the most advanced languages such as JEE and XML in web-based environment. It 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 a ZCM 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. As the first web framework dedicated to the Zakat domain, ZCMGenerator makes it very simple to generate simple ZCM models, while providing a high-level framework for more complex ZCM models. For the validation of the results of the generation of ZCM models in XML format, as well as the generation of the ZCR reports in XHTM format, we have used external tools that positively tested the conformity of these models to the standards of W3C. Experimental results show that, compared with traditional ZCP platforms, ZCMGenerator is more scalable and high-performance.

REFERENCES

[1] A. Mamouni, A. Marzak, and Z. A. Haddad, 'ZCMGProcess: ZCM-based process for the Zakat calculation models generation (ZCMGenerator a framework supporting this approach)', in 2016 11th International Conference on Intelligent Systems: Theories and Applications (SITA), 2016, pp. 1–6.

[2] K. A. Ali, 'XML Technology', 1983.

[3] A. Mamouni, A. Marzak, Z. Al Haddad, and Y. Boukouchi, 'Meta-Model for Zakat Calculation Platforms (ZCP) Based on the ADM Approach', Jun-2016.

[4] A. Mamouni, A. Marzak, Z. Al Haddad, and A. Belangour, 'A meta-model for automation of the deduction of judgments relating to Zakat', Dec-2015.

[5] A. Mamouni, A. Marzak, Z. Al Haddad, and Y. Boukouchi, 'ZCMGenerator: Framework for Generating Zakat Calculation Models based on the MDA Approach', Sep-2016.

[6] A. Huraimel, M. J. Zemerly, and A. Al-Hammadi, 'Islamic Zakah Application for Mobile Devices', presented at the The 3rd International Conference on Information Technology ICIT 2007, AL-Zaytoonah University, Amman, Jordan, 2007.

[7] A. Al-Riyami, A. A.-H. K. Al-Amri, and K. A. Al-Busaidi, 'Zakat Expert System', Vol. One, p. 31, 2014.

[8] A. Mohammad, A. Ahmad, H. Hassan, S. Rashid, and Z. Maamar, 'Design and Development of a Zakat Expert System', presented at the The 3rd Annual Undergraduate Research Conference on Applied Computing (URC 2011), DUBAI, UNITED ARAB EMIRATES, 2011, p. 35.

[9] H. Harun, N. Nordin, and A. Hussain, 'Ontology of Zakat Management System', in transfering, Managing and Maintaining knowledge for nation capcity development, Langkawi, Malaysia, 2008.

[10] N. F. I. Abdul Hamid and Z. M. Kasirun, 'PERISA: A PERSONAL ISLAMIC ASSET MANAGEMENT SYSTEM USING OBJECT-ORIENTED APPROACH', presented at the Proceeding of the 2nd International Conference on Informatics, Hilton Petaling Jaya Hotel, Kuala Lumpur, Malaysia, 2007, pp. 37–42.

[11] E. Fenty, K. Hulliyah, and M. Ekafitri, 'Applying mobile application development life cycle in the development of Zakat maal mobile web application using JQuery mobile framework', in Cyber and IT Service Management (CITSM), 2014 International Conference on, 2014, pp. 89–92.

[12] Noorul, Izzatthol, and Akhbarie, 'Muslim Android Application for Zakat Selangor (MAAZS)', Kuala Lumpur, Malysia, 2012.

[13] N. A. N. Ahmad, N. I. Akhbariee, and M. Hafizuddeen, 'Requirements analysis of android application using activity theory: A case study', in Information and Communication Technology (ICoICT), 2013 International Conference of, 2013, pp. 145–149.

[14] H. F. IMAM and S. Usman, 'SISTEM INFORMASI PERHITUNGAN ZAKAT BERBASIS ANDROID', Skripsi Fak. Ilmu Komput., 2015.

[15] R. Atunnisa, E. Satria, and R. Cahyana, 'PENGEMBANGAN APLIKASI ZAKAT BERBASIS ANDROID MENGGUNAKAN METODE PROTOTYPE', J. Algoritma, vol. 11, no. 1, 2015.

[16] R. S. Aguilar-Saven, 'Business process modelling: Review and framework', Int. J. Prod. Econ., vol. 90, no. 2, pp. 129–149, 2004.

[17] Tutorialspoint, 'SOFTWARE ARCHITECTURE & DESIGN INTRODUCTION'. 2013.

[18] M. Jibitesh, Software Engineering. India: Pearson Education India, 2011.