ZCMGenerator: Framework for Generating Zakat Calculation Models based on the MDA Approach

Mamouni Abdelaziz, Marzak Abdelaziz, Al Haddad Zayed, Boukouchi youness

Abstract— In these days, the ZCP (Zakat calculation platforms) form a necessary tool and play a crucial role in Muslims life. They have made the entire process of calculating Zakat extremely simple and straightforward. They support the Muslims in calculating their Zakat easily, correctly, quickly and accurately. However, many challenges exist in the area of Zakat calculation which can be an obstacle for its adoption by Muslims to calculate their Zakat. Indeed, the most of them are developed in traditional manner and have lack of portability and reusability. Moreover, the vast majority of them are designed for specific categories of wealth and limited to one school of jurisprudence. For this, the Framework ZCMGenerator is done to overcome these obstacles by facilitating Zakat calculation models modelling, ensuring reuse and portability and providing direction and guidance to ZCP developers.

Index Terms— ZCP, ZCMGenerator, Zakat, MDA, ADM, XML

1 Introduction

ZAKAT is one of the most fundamental obligations in Islamic religion. It is a fundamental component for the establishment of the economic community. Zakat is doubly important in the way of life that is Islam(Al-Qardawi, 1999). On the one hand it is a means of spiritual purification and on the other a way to regain balance and equilibrium in social and economic life. Nowadays, the number of Muslims paying their Zakat grows larger each year. But, most of them still do not comprehend the proper way of calculating their Zakat according to the Islamic law. This has led to a kind of guessing at the Zakat payment. Moreover, Muslims jurists differ on many details of Zakat. Each has his own opinion and arguments, and there are sometimes contradictions among their interpretations(Zakat Foundation of America, 2007).

To overcome these drawbacks, several Zakat calculation platforms have been developed. These platforms form a necessary tool and play a crucial role in Muslims life. Indeed, they have made the entire process of calculating Zakat extremely simple and straightforward. They support the Muslims in calculating their Zakat easily, correctly, quickly and accurately. However, this multitude of platforms, although it has advantages, it also presents a number of disadvantages viz; they are either incomplete, too complicated, or both; the most of them do not meet the new requirements that have emerged in the field of software engineering such as portability, reusability and interoperability; the majority of them are limited to one school of jurisprudence; they are poorly implemented with limited calculation categories; Certain of them are not recommended from any official authority so far; the vast majority of them are basically a Zakat calculator with no services; they do not meet the needs of almost all Muslims living in modern societies; and so on.

In order to overcome these limitations a new ZCP must be developed to fulfil the Zakat calculation requirements. For this, the ZCMGenerator Framework is done to overcome these obstacles by facilitating Zakat calculation models modelling and ensuring reuse and portability. To develop and implement this Framework, we have already proposed a new MDA-based process named ZCMGProcess for generating the Zakat calculation models.

The architecture of this MDA-based 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 General PIM, where the goal is to generate a new improved model. The third phase aims to generate the Zakat calculation models. These phases are in accordance with the three stages defined by the ADM approach that is, reverse engineering, restructuring, and forward engineering.

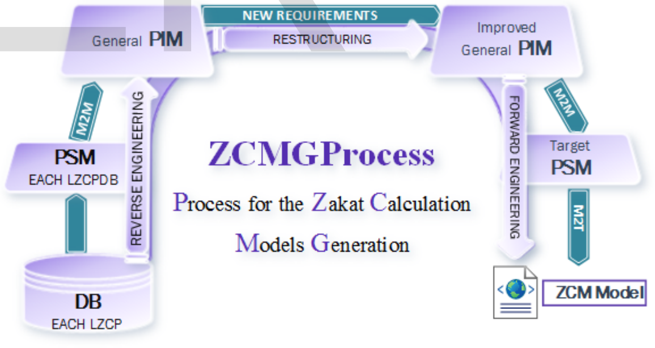


Fig. 1. ZCMGProcess

The aforementioned ZCMGProcess was inspired on idea of ADM (Architecture Driven Modernization) approach. It aims at guiding designers in the modelling of extensible, reusable, and portable Zakat calculation models which can be communicated to any platform without taking into account technical specifications using MDA 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(Ali, 1983).

This process has been defined by bearing in mind its use by developers who wish to design a new ZCP, but who wish 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 (Mamouni et al., 2016) a new Meta-model called ZC2M (Zakat Calculation Meta-Model) (figure 2) that is conforms to MOF meta-metamodel. It is a generic model that enables us, 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. This metamodel is given after an in-depth study of the Zakat domain (Mamouni et al., 2015) and a survey of the existing Zakat calculation platforms.

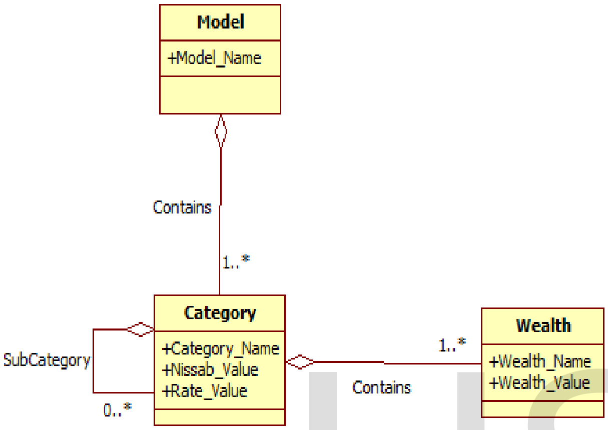


Fig. 2. ZC2M Metamodel

This Meta-model is divided into hierarchical elements. Its structure is divided into three levels viz; model, category and wealth, whose categories can be divided into several subcategories. The ZC2M metamodel describes the main concepts used in the Zakat domain. These concepts are described below.

1. Model: It represents a school of jurisprudence. Indeed, there are several schools of Islamic jurisprudence viz; Hanafi, Maliki, Shafii, Hanbali, etc. These schools share many of their rulings, but differ on how to calculate the amount owed to pay and other aspects of Zakat. Muslim scholars of each school interpret Zakat issues from a different viewpoint. For this and for taking into account all schools of jurisprudence, we choose to present each school by a model.
2. Categories: For the purpose of Zakat calculation, each of these schools has defined different categories of wealth, the amount that constitutes Nissab and the rate of Zakat that applies to each of these categories. Indeed, Zakat is divided into several categories, which is zakat on income, zakat on business, zakat on share, zakat on saving, zakat on gold and silver, zakat on EPF, etc. These categories are broken up into several sub-categories.
3. Wealth: Each category and sub-category has a collection of wealth. For instance, Zakat on income has the following wealth, salary, allowance, bonus and other income.
4. Nissab: It means the minimum amount of property, wealth or assets owned by an individual upon which Zakat is payable. This minimum amount of wealth is separately defined for each category of wealth. The schools of jurisprudence have prescribed a specified Nissab for each category of wealth or possession. This minimum amount of wealth is called the nissab threshold (the limit of Nissab). Zakat becomes due when the wealth or possessions of an individual exceed this nissab threshold(Mazhar Hussaini et al., 2005).
5. Zakat Rate: Wealth is a product of labor and capital input. The rate of Zakat varies with the amount of labor and capital expended. The greater the amount of labour and capital requirements in the generation of wealth, the lower will be the rate of Zakat and vice versa. This principle in general enhances the productivity of labour and capital. Islamic jurisprudence has determined the percentage rate of Zakat applied to each Zakatable item (Mazhar Hussaini et al., 2005). The rate of Zakat on each of the above categories is different. Once the categories of wealth reach Nissab, they will be subjected to four types of rates viz; 2.5% from the capital income; 10% (for harvest from non-cultivated land) or 5% (for harvest from cultivated land) from the land crops; according to detailed code of rates for livestock; and 20% of the found treasures (Rikaz) (Mahmud & Shah Haneef, 2008).

This ZC2M metamodel will be the basis used to properly instantiate the ZCM models in XML file, whereas in this paper, we present the Framework ZCMGenerator that is a web-based application. It is a Meta-model based tool for designing ZCM models and permitting, through mechanisms of transformation, to define and generate the ZCM models in XML file.

The remainder of this paper is organized as follows. Section II summarizes related work. Section III introduces the main approaches and standards on which the proposal is based. Section IV presents the most relevant details of the ZCMGenerator. Finally, Section V discusses our conclusions and future work.

2 Related Work

The purpose of this literature review is to get more understanding on Zakat calculation platforms using different approaches. Nowadays, several Zakat calculation platforms have been developed due to the high demands on such Islamic applications. There are two main categories for such a platform namely, desktop-based and web-based platforms.

In (Huraimel, Zemerly, & Al-Hammadi, 2007), an application that calculates the Zakat for mobile phone users with a GPRS connection has been developed using J2ME technology. This application is based on client-server approach. The server side is just various web pages to get the prices of gold, silver and stocks. On the other hand the Zakat application in the mobile side is very important module since it is what the clients will use. The main functionality in the mobile device side is to allow the client to set the Zakat configuration and to add Zakat items or to list the saved Zakat items. These items will be stored in the record management system

(RMS). This platform has been built using the J2ME package which makes it a highly portable application. It is user friendly and the Zakat can easily be calculated at the end of the year on the day the user specifies. Also, it has the ability to obtain online prices for various Zakat items such as gold, silver, and stocks. Moreover, it allows the user to change the school of jurisprudence and year type and can be used in offline mode in case a GPRS connection is not available but in this case the user should provide the required prices. However, it is limited to Mobile phone users and to some Zakat items.

(Al-Riyami, Al-Amri, & Al-Busaidi, 2014) 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 idea of an Expert System is to convert knowledge from human expertise to a computerized system. The Zakat expert system was developed using a freeware rule-based shell called eXpertise2Go that is a web-based expert system. Knowledge that is acquired from a human expert is represented in the form of rules, such as IF-THEN statements. These rules are used by the inference engine component to perform the reasoning process. The developed expert system offers the calculation of different types of Zakat, the user has the freedom of choice, either one type or as much as he desires to calculate in an understandable and clear manner. Also, the system is very useful for calculating Zakat quickly and accurately, very beneficial, relevant, convenient, and easy to use. However, it had some technical and presentation limitations viz; the system does not support the Arabic language, the need for a computer to operate it and it does not state the source of the information. Moreover, it is not user friendly and also is not able to answer to questions of all Muslims belonging to different Islamic schools.

(Abdul Hamid & Kasirun, 2007) have been developed Islamic Asset Management System (PERISA) using object-oriented approach by implementing Rational Unified Process (RUP) model. Using the RUP, software product life cycles are broken into four main phases that are termed as inception, elaboration, construction and transition phase. PERISA is developed using PHP language and MySQL database. This platform consists of nine modules, which are General information, Admin, Profile, Archive, Registration, Zakat, Fidiah, Heritage and Waqaf. Zakat module is a module to manage zakat, which consists of zakat details information, zakat calculator, personal zakat organizer and list of zakat counters. The main function of zakat module is to calculate zakat. Zakat is divided into six categories, which is zakat on income, zakat on business, zakat on share, zakat on saving, zakat on gold and silver and zakat on EPF. This platform is easy to use for any level of users and flexible for future integration and enhancement. It uses English language to enable it to be used worldwide and compatible with various platform. However, it is limited to one school of jurisprudence and to Windows and Linux users.

(Fenty, Hulliyah, & Ekafitri, 2014) have proposed to apply mobile application development life cycle approach (MADLC-Approach) in the development of Zakat Al-Mal mobile web application using JQuery Mobile Framework. JQuery Mobile is a development platform of JQuery, which provides a variety of user-interface elements and features for use in mobile applications. The MADLC includes seven phases viz; identification, design, development, prototyping, testing, deployment and maintenance phase. This platform can help 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. Users can also receive an e-mail notification containing the amount of zakat that should be issued along with a list of any assets that have been entered such as Nissab and Haul. It can be accessed by any platform and delivers speed, stability, and an excellent cross-browser experience for web mobile visitors. However, it is limited to one school of jurisprudence and does not cover all categories of wealth.

(Noorul, Izzatthol, & Akhbarie, 2012) have developed Muslim Android Application for Zakat Selangor (MAAZS) using agile methodology. Agile allows for a more iterative for the development, which includes the client throughout the development of their system, using agile stakeholder can be join all process in development. The Mobile application development process comprises five phases viz; analyze, design, development, certify, deploy and maintenance phase. The MAAZS approach is developed using Rational Unified Process (RUP) model and is being implemented using PHP and JQueary language and MySQL database for the Android mobile platform. It provides a calculator to ease the users in calculating various type of zakat easily and effectively without hassle. It not only has the ability to calculate Zakat but also the ability to produce user's status either the user is eligible to pay zakat, not eligible to pay or qualified to receive zakat. However, it is developed especially for Selangor Residents and does not cover all types of wealth.

(Ahmad, Akhbariee, & Hafizuddeen, 2013) have proposed the requirements analysis approach using the activity theory for android application development. The requirements are then used to develop a zakat application that is based on android platform. Activity theory is a conceptual framework that focuses on the activity, where the interaction between actors and their surroundings are counted. The benefit of using activity theory for android application development is, we can see the interaction between the user (subject), community and the environment in order to accomplish certain activity and they are mediated by mobile technologies. The applied activity theory includes nine steps viz; identify problems, identify actions and operations of the activity and Model the activity using Engeström's diagram. The latter consists of six essential elements, namely, subject, object, tools, rules, community and division of labour. Basically, an activity theory helps to structure the analysis. However, it does not prescribe what to look for. Moreover, this platform does not cover all wealth and all schools of jurisprudence.

(IMAM & Usman, 2015) have developed Zakat Calculation Information System using Waterfall model. This model 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.

TABLE 1
SUMMARY OF EXISTING ZAKAT CALCULATION PLATFORMS

Reference	Approach	Technology	Environment	Zakat items	Type	Schools	Characteristics / Advantages	Limitations / Challenges
Huraimel et al.(2007)	Client-Server	J2ME GPRS Toolkit's emulator	Mobile devices	Money Gold Silver Stocks Fruits Crops Cattle	Desktop-based	Shafii Hanbali Maliki Hanbali	Provides online prices of Gold, Silver and stocks. Covers all schools and years types. Can be used offline and online. Provides full and partial zakat calculation. Calculates the personal allowance (Nissab).	Limited to Mobile phone users and to specific zakat items.
Al-Riyami et al.(2014)	Expert System	EXpertise2Go	Windows	Cash Silver Gold Securities Debt Cattle Crops Merchandise	Web-based	Maliki	Calculates zakat quickly. There is no need for an internet connection. Provides full and partial zakat calculation. Calculates a total Zakat for money type.	Poor user interface design Limited to one schoo and to specific wealth.
Abdul Hamid & Kasirun(2007)	Object Oriented	Dreamweaver PHP MySQL Photoshop Rational Rose Microsoft Project UML	Windows Linux	Income Business Share Saving Gold Silver EPF	Web-based	Shafii	It is flexible for future integration and enhancement. It uses English language to enable it to be used worldwide. Provide full and partial zakat calculation. The system is compatible with various platforms.	Limited to one school of jurisprudence and to Windows and Linux users.
Fenty et al.(2014)	MADC-approach	HTML5 UML JQuery Mobile Dreamweaver CS5 XMAPP	Mobile devices	Gold Silver Income Mines Business Livestock Agricultural products Rikaz	Web-based	Shafii	Users can receive e-mail notification containing the amount of zakat that should be issued along with a list of any assets that have been entered. It is compatible with various platforms.	Limited to one school of jurisprudence and does not cover all types of wealth.
Noorul et al.(2012)	Agile Methodology	PHP JQuery Java Script UML MySQL Eclipse Photoshop Rational Rose	Android	Earning Business Savings Stock EPF Gold Sliver	Desktop-based	Shafii	It not only has the ability to calculate zakat but also the ability to produce user's status either the user is eligible to pay zakat, not eligible to pay or qualified to receive zakat.	Limited to Selangor Residents, to specific wealth, and to mobile users.
Ahmad et al.(2013)	Activity theory	Engeström Endroid	Android	Gold Silver Savings Stocks Business EPF Earning	Web-based	Shafii	Allow to find the nearest zakat counter.	Limited to Malay language and to specific items.
Imam & Usman(2015)	Waterfall model	JAVA UML Eclipse	Android	Gold Silver Agricultural products Savings Stocks Business Profession Gifts Livestock	Web-based	Shafii	It has a complete calculation which includes the zakat Al-Fitr and zakat Al-Mal and it can help users in calculating zakat independently.	Limited to one school of jurisprudence and does not cover all types of wealth.
Atunnisa et al.(2014)(Atunnisa, Satria, & Cahyana, 2015)	Prototype method	Java Eclipse UML	Android	Professions Zakat Al-Mal Zakat Al-Fitr	Desktop-based	Shafii	It can be used in the mobile devices and tablets based on Android. The Nissab is predetermined.	Limited to one school and does not cover all types of wealth.

This application has been developed using the Java programming language. This platform is developed to help the community in general and in particular the users of Android-based Smartphone in terms of calculating zakat including zakat Al-Fitr and zakat Al-Mal. However, it is limited to one school of jurisprudence and does not cover all types of wealth.

(Atunnisa et al., 2015) have developed a zakat calculation application using prototype method. A prototype display main functional capabilities of the proposed system. The process of prototyping involves six steps viz; Requirements gathering, Quick design, Building prototype, Customer evaluation, Refining prototype and Engineer product. Prototyping 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, the prototype model is usually not complete and many of the details are not built in this prototype. Moreover, the developed system does not cover all types of wealth and limited to one school of jurisprudence.

In Table I, we provide an analytical study of above platforms in tabular format with their strengths and limitations are given.

In this section, we analyzed the different existing Zakat calculation platforms. This analysis shows that although multiple ZCP using different approaches have already been proposed which help in calculating the amount owed to pay. Each platform has some limitations and there is no universal efficient solution found yet. Indeed, this multitude of platforms, although it has advantages, it also presents a number of disadvantages viz; they do not provide complete requirements of Zakat calculation; they are either incomplete, too complicated, or both; 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; the most of them are limited to one school of jurisprudence and are poorly implemented with limited calculation categories; the vast majority of them are not recommended from any official authority and basically a Zakat calculator with no services; they do not meet the needs of almost all Muslims living in modern societies; and so on. Also, our research shows that there are eight main groups of ZCP development approaches viz; approach-based on expert system technology, approach-based on client-server, approach-based on object-oriented, approach-based on agile methodology, approach-based on prototype model, approach-based on activity theory, approach-based on mobile application development life cycle and approach-based on Waterfall Model. To overcome the limitations of these existing ZCP, in this paper, a new Framework named ZCMGenerator has been proposed to fulfil the Zakat calculation requirements, which aims to generate the extensible, reusable and portable ZCM models and to ensure scalability of these platforms.

3 BackGround

The following subsections introduce the two main concepts to provide a better understanding of the proposal viz; the Model-Driven Architecture standard and XML technology.

2.1 Model-Driven Architecture

The Object Management Group (OMG) introduced in 2001 the Model-Driven Architecture (MDA) initiative as an approach to system-specification and interoperability based on the use of models in software development (Poole, 2001). This initiative is aimed at increasing productivity and re-use through separation of concern and abstraction. Its three primary goals are portability, interoperability and reusability through architectural separation of concerns (Truyen, 2006). As shown on figure 1, in MDA, a Platform Independent Model (PIM) is an abstract model which contains enough information to drive one or more Platform Specific Models (PSMs) (Sparx Systems, 2007).

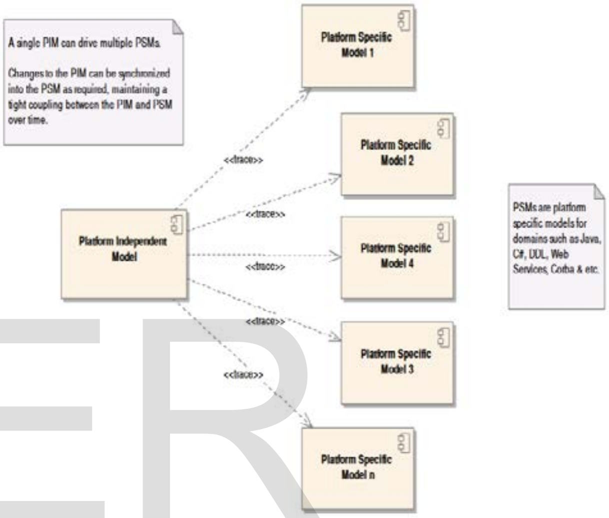

A single PIM can drive multiple PSMs.
Changes to the PIM can be synchronized into the PSM as required, maintaining a tight coupling between the PIM and PSM over time.
Platform Independent Model
Platform Specific Model 1
Platform Specific Model 2
Platform Specific Model 4
Platform Specific Model 3
Platform Specific Model n
<<trace>>
<<trace>>
<<trace>>
<<trace>>
<<trace>>
PSMs are platform specific models for domains such as Java, C#, DDL, Web Services, Corba & etc.

Fig. 3. Driven one or more PSMSs from PIM

The MDA is a model-driven framework for software development that proposes to specify the whole system with platform independent models (PIM) to be able to develop the software system for different platforms (Caceres, Marcos, & Vela, 2003). It specifies three default viewpoints on a system viz; computation independent, platform independent and a platform specific, and specifies three default models corresponding to these viewpoints (Truyen, 2006).

1. Computation Independent Model (CIM): A CIM is a business or domain model. A CIM presents exactly what the system is expected to do, but hides all information technology related specifications to remain independent of how that system will be implemented.
2. Platform Independent Model (PIM): A PIM is a model with a high level of abstraction, independent of any implementation technology.
3. Platform Specific Model (PSM): A PSM is a model for a specific platform. A PIM can be transformed into one or more PSMs, which are for a specific implementation technology.

MDA aims to separate business or application logic from underlying platform technology(Force, 2012) and to enhance portability by separating system architecture (abstract) from platform architecture (concrete). MDA has the capability to define transformations that map from PIMs to PSMs. This facilitates the development of a system in abstraction, and simplifies implementation of that system across a variety of target platforms(Sparx Systems, 2007). For this, it provides mappings between models. A mapping is a set of rules and techniques used to modify one model in order to get another model(Caceres et al., 2003). Mappings are used to transform:

1. PIM to PIM: This transformation is used to map the analysis to design models.
2. PIM to PSM: This transformation is used when a PIM is refined to be deployed in a platform specific model.
3. PSM to PSM: This transformation usually is related to platform specific model refinement.
4. PSM to PIM: This transformation is required to obtain a platform independent model from models existing in a particular technology.

According to (Lopes et al., 2006), the creation of mapping specification and transformation definition is not an easy task, for this the others proposed a new process that involves several entities (figure 4) namely, metametamodel, a source and target metamodel, a source and target model, a mapping model, a transformation language model, and a transformation engine.

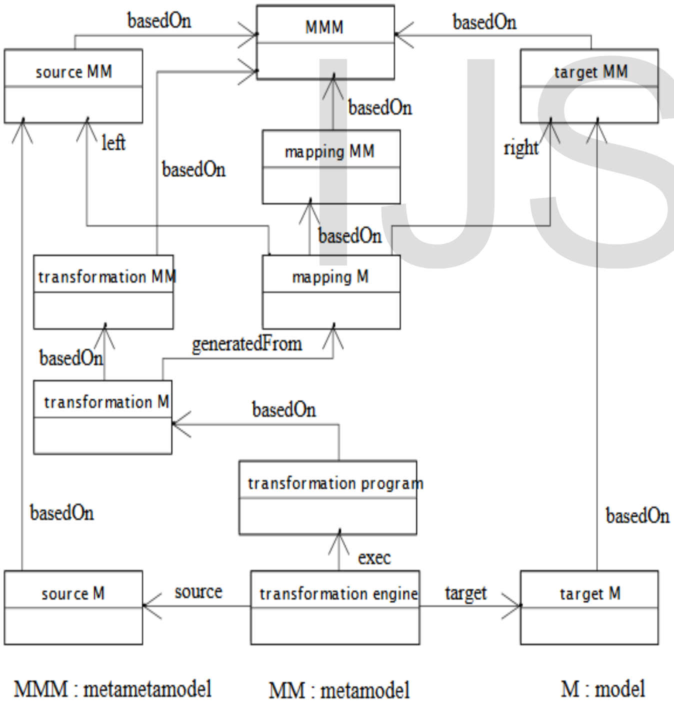


Fig. 4. Transformation process in MDA

The MDA approach gives the facility to understand complex and real-world systems while providing an abstraction of the physical system as shown on Figure 5. This abstract view of the system is represented through the OMG's modelling standards namely the Unified Modelling Language (UML), the Meta-Object Facility (MOF) and the XML Metadata Interchange (XMI)(DE ALMEIDA, 2008).

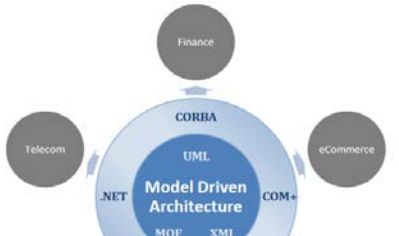


Fig. 5. OMG's modeling standards

2.2 XML technology

XML is an open and flexible standard created by the World Wide Web Consortium (W3C) in early 1998 for storing, publishing and exchanging any kind of information (Ali, 1983). It is an abbreviation for extensible mark-up language, in many ways it is similar to HTML(Lindsay, 2008). XML brings a number of powerful capabilities to information modelling viz; heterogeneity, extensibility, flexibility, and many others. According to (XML4Pharma, 2004), XML has taken a high flight in a short time viz;

1. XML is human readable: You do not need a special program to actually read the contents of the file.
2. XML is self-describing: The metadata are included in the file.
3. XML is in Unicode: The Unicode character set includes all the characters of almost all the written languages in the world. So you can have the same document in French or any other written language.
4. XML is hierarchical data: This facilitates exchange of the information.
5. XML is machine-independent: XML allows exchanging data between Windows, UNIX and McIntosh.
6. XML separates content from presentation: XML itself is purely content, whereas the content can be presented in several ways, e.g. in HTML, in WAP, in Word, in Excel, and even in PDF etc. The presentation is usually performed by using a style sheet. One such a style sheet can then be used to present different XML-files which have the same data structure.
7. Results from SQL queries into databases can easily be converted to self-describing XML: XML files can also easily be stored into databases. The major database manufacturers even base their database technology on XML and Java.

There are a lot of technologies related to the XML viz; SOAP (the Simple Object Access Protocol), XLink (the XML Linking), XPointer (the XML Pointer Language), XSL (the eXtensible Stylesheet Language), in addition to APIs (the Application Programming Interfaces). Application Programming Interfaces APIs are used to process an XML document by accessing its internal structure (Ali, 1983).

1. Document Object Model (DOM) is an API for tree-based XML parsing. DOM provides a standard set of programming interfaces for accessing, reading, writing, and modifying the nodes in the document tree and represents a document tree fully in memory, where it

can be accessed and manipulated by the user. Its major advantage is that it provides a complete and editable view of the document's contents. But, it requires lots of processing power and memory.

2. The Simple API for XML (SAX) is a simple API for event-based XML parsing. SAX does not create a data structure. SAX scans an input document and generates events, such as the start of an element, the end of an element, and so on. Its major advantage is that SAX can result in reduced memory overhead. But, SAX does not allow random access manipulation of the document.
3. The Java Document Object Model (JDOM) is a simple productive API for XML parsing. JDOM incorporates the best features of DOM and SAX and provides a full document view with random access, but it does not require the entire document to be in memory

Figure 6 below shows the life cycle of XML document

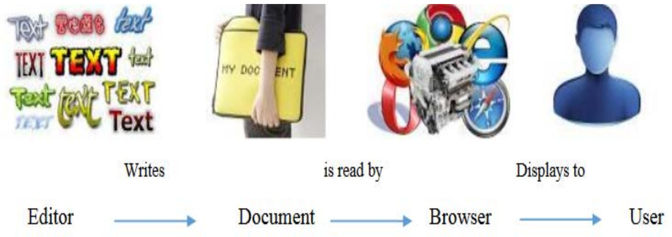


Fig. 6. Life cycle of XML document

We have now covered the key concepts needed to use MDA concepts and XML technology. In the next section, we will describe how we can use these powerful technologies to develop and implement our Framework.

4 ZCMGenerator Framework

4.1 ZCMGenerator's features

The ZCMGenerator (Generator of Zakat Calculation Models) is a Zakat calculation models generation framework that follows the Model Driven Architecture (MDA) paradigm. It takes a design model from their input and generates the ZCM (Zakat Calculation Model) models in XML file (Figure 7). The generated model contains schools, categories, sub-categories and wealth. This model can be adapted to the construction of a new Zakat calculation platform. This Framework was built to support automatic generation of ZCM models. In fact, ZCMGenerator is more than one tool of generation of ZCM Models. It offers a lot of functionality to calculate Zakat, to generate the Zakat calculation reports, to send and receive the e-mail notification, and so on. Another key concept of ZCMGenerator is the strong association between calculation methods and the generated XML file. Our framework uses XML technology, which could be easily, used to generate, and communicate models between all the stakeholders that is responsible for the construction of Zakat calculation platforms without considering the diversity of platforms. XML brings a number of powerful capabilities to information modelling viz; heterogeneity, extensibility, flexibility, internationalization and many others. For this, our framework instantiates these ZCM Models in XML file format using JDOM (Java Document Object Model) API.

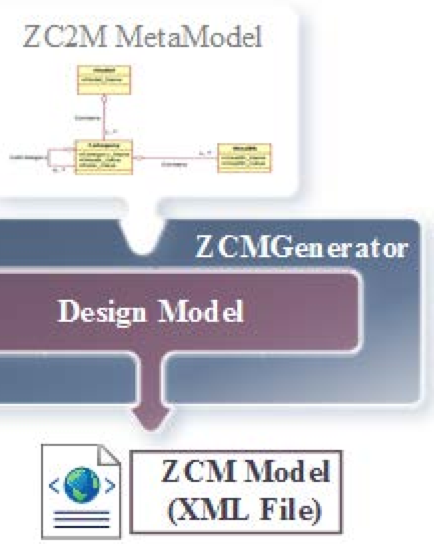


Fig. 7. ZCM Models generation

4.2 Business process of ZCMGenerator

A business process is the combination of a set of activities with a structure describing their logical order within an enterprise. Its objective is to produce a desired result(Aguilar-Saven, 2004). It can be visualized as a flowchart or as a process matrix of a sequence of activities. Four our case, we have used cross functional flowcharts for modelling the business process of ZCMGenerator. Indeed, a basic flowchart shows how micro processes are sequentially linked together in an operation, whereas, cross functional flowcharts allows to expand the limitations of the conventional flowcharts to a multidimensional range.

Figure 8 depicts our business process by using cross functional flowchart. In this business process, we describe the four steps involved in our Framework viz;

1. Analyze: In this step, we define and describe the requirements and the needs of the users.
2. Modeling: In this step, we design the model of requirements, which is defined in the previous step. These requirements are transformed into a ZCM Model under the following structure: schools, categories, sub-categories and wealth. The generating ZCM Model is in accordance with our ZC2M meta-model.
3. Development: This step consists of developing the Zakat calculation functions and implementing it in the platform. These functions are developed by specialists of development and implementation according to the specificity of platform Java. It allows associating the values of wealth in XML file in order to Zakat calculation.
4. Reporting: After the calculation step, the final step is to prepare proper reports for the users and to send those reports to specific users.

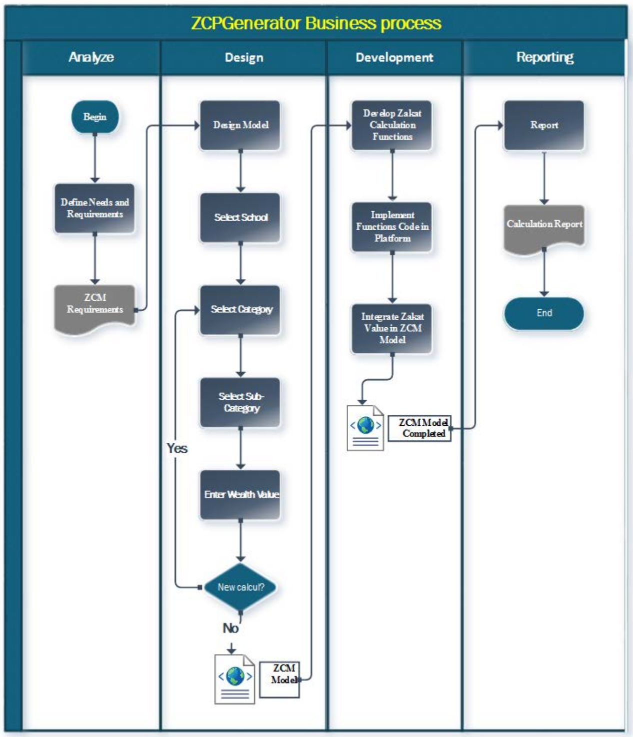


Fig. 8. Business process of ZCMGenerator

These are the four steps involved in our Framework. These steps should only be carried out in this order viz; analyze, modelling, development and finally, reporting. The next section describes in detail its technical architecture.

4.3 Technical Architecture

Figure 9 shows technical architecture of our framework. This architecture is implemented as a three-tiered application with user interface, functional logic and data storage layers.

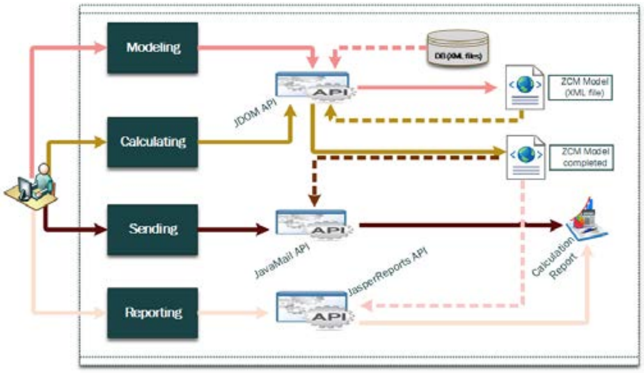


Fig. 9. Technical Architecture of ZCMGenerator

We used Struts2 Framework to build web based presentation layer, Spring Framework for business layer and Hibernate for the data access layer (Figure 10).

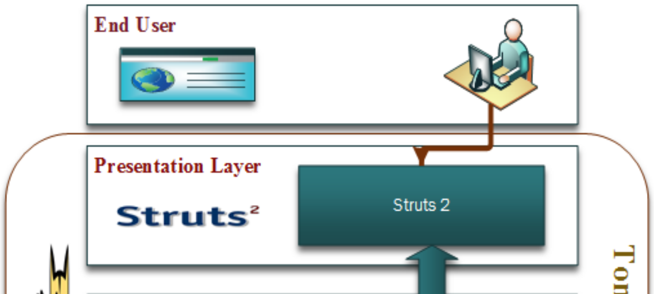
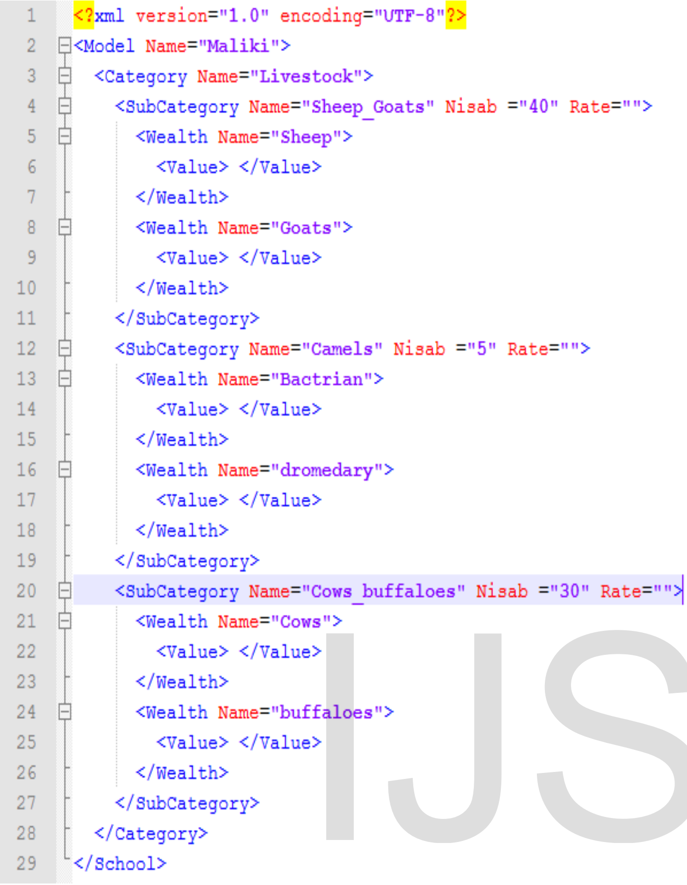


Fig. 10. Three-tiered application

This Framework is developed by using the JEE programming language in Eclipse environment. We used the JDOM API (Java Document Object Model) for reading, writing, and manipulating XML from within Java code, that is an open source Java-based document object model for XML that was designed specifically for the Java platform. We used JasperReports for generating the Zakat calculation reports that is a popular reporting tool used by the java programmer. It provides the necessary features to generate dynamic reports. Finally, we used the JavaMail API for sending the e-mail notifications containing the amount of Zakat that should be issued along with a list of information that have been selected or entered such as school, categories, subcategories and wealth. In fact, ZCMGenerator is not just a tool for generating ZCM models. It is a fairly comprehensive yet practical platform that straightforwardly addresses the contemporary uncertainties and concerns of Muslims. It is a Framework that provides a few special features for users such as a flexible Zakat calculator to calculate the amount owed to pay including all Zakat categories and all schools of jurisprudence, an e-mail service enabling users to submit detailed questions to our platform and to receive the e-mail notifications containing the amount of Zakat that should be issued along with a list of information that have been selected or entered. Another feature of this Framework is able to generate dynamic calculation reports and has been equipped with detailed information on Zakat.

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 Zakat categories 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.

Figure 11 shows an example of the XML file generated by our framework ZCMGenerator. The ZCM model generated correspond to the case of Zakat on Livestock according to Maliki law.



```
<?xml version="1.0" encoding="UTF-8"?>
<Model Name="Maliki">
  <Category Name="Livestock">
    <SubCategory Name="Sheep_Goats" Nisab ="40" Rate="">
      <Wealth Name="Sheep">
        <Value> </Value>
      </Wealth>
      <Wealth Name="Goats">
        <Value> </Value>
      </Wealth>
    </SubCategory>
    <SubCategory Name="Camels" Nisab ="5" Rate="">
      <Wealth Name="Bactrian">
        <Value> </Value>
      </Wealth>
      <Wealth Name="dromedary">
        <Value> </Value>
      </Wealth>
    </SubCategory>
    <SubCategory Name="Cows_buffaloes" Nisab ="30" Rate="">
      <Wealth Name="Cows">
        <Value> </Value>
      </Wealth>
      <Wealth Name="buffaloes">
        <Value> </Value>
      </Wealth>
    </SubCategory>
  </Category>
</School>
```

Fig. 11.The XML file generated by Framework ZCMGenerator

The ZCMGenerator generates this Livestock model using the Zakat calculation rules basing on the Maliki law. Maliki law does not stipulate details in respect of Zakat on livestock. It simply provides that the calculations for Zakat applicable to other forms of property similarly apply to livestock. It does assert in particular that the only condition upon which Zakat on offspring is paid is that their mothers are kept with the owner throughout the year. Moreover, according to (Abu Muhammad, 2001), Sheep and goats are considered as one category for Zakat purposes. This is by consensus. That is what he said in at-Tahqiq because the name of the species includes both of them in the words of the Prophet. Cattle and buffaloes are one category. This is by agreement because the generic name includes them all. Bactrian and dromedary camels are one category. These are the camels of the people of Khorasan since camel is used for both of them in the words of the Prophet.

4 Conclusion

In recent years, several ZCP (Zakat Calculation Platforms) have been developed due to the high demands on such Islamic applications. These platforms form a necessary tool and play a crucial role in Muslims life. Until now, there is still no implicit solution to ensure ZCP portability and reusability, for this, we are working on a framework to achieve this purpose. In order to devise the best approaches for implementation of our framework, different approaches were studied. Our research shows that recently, Model Driven Approaches are increasingly exploited to develop different frameworks to solve several issues such as portability, reusability and interoperability. For this, this paper provides state-of-the-art of existing ZCP. Moreover, in order to have better understanding our Framework ZCMGenerator, this paper describes in detail the architecture technique, the features and the capability of the ZCMGenerator to enhance the portability and reusability of existing ZCP.

The generic architecture of the framework not only takes into account of the general MDA patters but also integrates other important aspect of portability, reusability and interoperability that is XML technology. Our Framework uses this technology, which could be easily, used to generate, and communicate models between all the stakeholders that is responsible for the construction of ZCP without considering the diversity of platforms.

References

[1] Abdul Hamid, N. F. I., & Kasirun, Z. M. (2007). PERISA: A PERSONAL ISLAMIC ASSET MANAGEMENT SYSTEM USING OBJECT-ORIENTED APPROACH (pp. 37–42). Presented at the Proceeding of the 2nd International Conference on Informatics, Hilton Petaling Jaya Hotel, Kuala Lumpur, Malaysia. Retrieved from http://repository.um.edu.my/413/1/SDD.pdf

[2] Abu Muhammad, A. (2001, January 12). The Zakat of Livestock. Retrieved 4 April 2016, from https://bewley.virtualave.net/Rislive.html

[3] Aguilar-Saven, R. S. (2004). Business process modelling: Review and framework. International Journal of Production Economics, 90(2), 129–149.

[4] Ahmad, N. A. N., Akhbariee, N. I., & Hafizuddeen, M. (2013). Requirements analysis of android application using activity theory: A case study. In Information and Communication Technology (ICoICT), 2013 International Conference of (pp. 145–149). IEEE. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6574563

[5] Ali, K. A. (1983). XML Technology. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.656.4317&rep=rep1&type=pdf

[6] Al-Qardawi, Y. (1999). Fiqh az-Zakat: A comparative study. Dar Al-Tawwa, Ltd, London, 407.

[7] Al-Riyami, A., Al-Amri, A. A.-H. K., & Al-Busaidi, K. A. (2014). Zakat Expert System. Volume One, 31.

[8] Atunnisa, R., Satria, E., & Cahyana, R. (2015). PENGEMBANGAN APLIKASI ZAKAT BERBASIS ANDROID MENGGUNAKAN METODE PROTOTYPE. Jurnal Algoritma, 11(1). Retrieved from http://sttgarut.ac.id/jurnal/index.php/algoritma/article/view/142

[9] Caceres, P., Marcos, E., & Vela, B. (2003). A MDA-based approach for web information system development. In Workshop in software model engineering. Citeseer. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.94.8774&rep=rep1&type=pdf

[10] DE ALMEIDA, P. (2008). MDA–Model Driven Architecture. Retrieved from http://diuf.unifr.ch/drupal/sites/diuf.unifr.ch.drupal.softeng/files/teaching/studentprojects/mda/AmeidaP_Ma_Dec_08.pdf

[11] 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 Cyber and IT Service Management (CITSM), 2014 International Conference on (pp. 89–92). IEEE. Retrieved from http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=7042182

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

[13] Huraimel, A., Zemerly, M. J., & Al-Hammadi, A. (2007). Islamic Zakah Application for Mobile Devices. Presented at the The 3rd International Conference on Information Technology ICIT 2007, AL-Zaytoonah University, Amman, Jordan. Retrieved from http://icit.zuj.edu.jo/icit07/PaperList/Papers/314Ahmed%20.pdf

[14] IMAM, H. F., & Usman, S. (2015). SISTEM INFORMASI PERHITUNGAN ZAKAT BERBASIS ANDROID. Skripsi, Fakultas Ilmu Komputer. Retrieved from http://eprints.dinus.ac.id/16813/

[15] Lindsay, J. (2008). XML Databases and Biomedical Informatics. Retrieved from http://www.engr.uconn.edu/~steve/Cse300/lindsay.pdf

[16] Lopes, D., Hammoudi, S., Bézivin, J., & Jouault, F. (2006). Mapping specification in MDA: From theory to practice. In Interoperability of enterprise software and applications (pp. 253–264). Springer. Retrieved from http://link.springer.com/10.1007/1-84628-152-0_23

[17] Mahmud, M. W., & Shah Haneef, S. S. (2008). Debatable issues in fiqh al-Zakat: a jurisprudential appraisal. Jurnal Fiqh, 5. Retrieved from http://repository.um.edu.my/699/

[18] Mamouni, A., Marzak, A., Al Haddad, Z., & Belangour, A. (2015, December). A meta-model for automation of the deduction of judgments relating to Zakat. Retrieved from http://www.ijcttjournal.org/2015/Volume30/number-1/IJCTT-V30P101.pdf

[19] Mamouni, A., Marzak, A., Al Haddad, Z., & Boukouchi, Y. (2016, June). Meta-Model for Zakat Calculation Platforms (ZCP) Based on the ADM Approach. Retrieved from http://www.ijeecse.com/V3N2-007.pdf

[20] Mazhar Hussaini, M., Siddiqui, M., Noor Abdullah, M., & Omar Farooq, M. (2005). THE INSTITUTION OF ZAKAT (First Edition). Chicago: The Council of Islamic Organizations of Greater Chicago (CIOGC). Retrieved from https://zakatchicago.com/wp-content/uploads/2015/04/ZakatBook.pdf

[21] Noorul, Izzatthol, & Akhbarie. (2012). Muslim Android Application for Zakat Selangor (MAAZS). Kuala Lumpur, Malysia. Retrieved from https://fyprepository.files.wordpress.com/2014/11/muslim-android-application-for-zakat-selangor.pdf

[22] Poole, J. D. (2001). Model-driven architecture: Vision, standards and emerging technologies. In Workshop on Metamodeling and Adaptive Object Models, ECOOP (Vol. 50). Retrieved from http://www.adaptiveobjectmodel.com/ECOOP2001/submissions/Model-Driven_Architecture.pdf

[23] Sparx Systems. (2007). MDA Overview. Retrieved from http://www.omg.org/mda/mda_files/MDA_Tool-Sparx-Systmes.pdf

[24] Truyen, F. (2006). The Fast Guide to Model Driven Architecture The Basics of Model Driven Architecture. Cephas Consulting Corp. Retrieved from http://secure.omg.org/mda/mda_files/Cephas_MDA_Fast_Guide.pdf

[25] XML4Pharma. (2004). XML Technology - what is it□? Retrieved 18 April 2016, from http://www.xml4pharma.com/XML/

[26] Zakat Foundation of America. (2007). Th Zakat hand book a practical guide for Muslims in the West. Prosody LLC—A Writing Company, United States of America. Retrieved from www.thezakat.org