MDA and Object-Oriented System Analysis and Design Integration for TanSSe-L System Development

Ellen A. Kalinga
Department of Computer Science and Engineering
College of Information and Communication Technologies
University of Dar es Salaam
Tanzania

and

Burchard B. Bagile
Department of Computer Science and Engineering
College of Information and Communication Technologies
University of Dar es Salaam
Tanzania

ABSTRACT

In software development, a successful information system is subject to frequent evaluation and revision within the framework known as System Development Life Cycle (SDLC). SDLC is the basis for most software development methodologies, whether structured approach or object oriented approach. Tanzania Secondary Schools e-Learning (TanSSe-L) system is a customized learning management system (LMS) developed to enable ICT support in teaching and learning functions by allowing for the creation and storage of learning materials, making them available, easily accessed and sharable by students from different secondary schools in Tanzania. The development of the TanSSe-L system made use of software engineering discipline using Object-Oriented System Analysis and Design (OOSA&D) with Unified Modelling Language (UML) and Model Driven Architecture (MDA). Implementation made use of open source LMS through customization to help generate a timely solution to TanSSe-L system development.

This paper presents how TanSSe-L system was developed through the integration of Object-Oriented System Analysis and Design approach and Model Driven Architecture to address the SLDC in a systemic way, the approach which led to the stage of customizing an open source LMS. It shows the importance of creating a base guide for customization.

Keywords—Software Engineering, Object Oriented System Analysis and Design, UML, MDA, Customization, ICT, e-Learning, Tanzania Secondary School

1. INTRODUCTION

As defined by Pollice [13] and the Institute of Electrical and Electronics Engineers (IEEE), [6], software engineering is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software, and the study of these approaches. Software engineering addresses the SDLC in a systematic way.

LMSs that are in use today are either commercial products (e.g. WebCT, Blackboard), or free open source products (e.g. moodle, claroline), or customized software systems that serve the instructional purposes of particular organizations. LMS that belong to the third category are exponentially increasing, as most education and training institutions are building or planning to build their own LMS. This is due to the fact that a customized LMS will fit better their specific learning purposes, and proves to give a good return of investment over the years [2].

Tanzania being a developing country could not afford the cost of commercial software. It was wise to effectively use the advantaged opportunities brought by open source community. There is a possibility of creating something concrete using open source software. In this view, TanSSe-L system was fully developed using open source software (OSS). OSS included: Open Source LMS for customization, Apache Web server, MySQL database management system (DBMS), PHP scripting language and Linux operating system.

Many OS LMS are designed to suite higher institutions while TanSSe-L system was intended to be used by lower level of secondary schools, as per Tanzanian specific context. Customization of OS LMS for TanSSe-L system development needed a detailed customization guide. Apart from several strategies presented by Kalinga et al [7] for the development of TanSSe-L system, Object oriented system analysis and design together with MDA were used to get a well defined software development process. Object oriented system analysis and design (OOSA&D) with Unified Modeling Language (UML) in software engineering discipline was used to create a TanSSe-L system Platform Independent Model as per MDA as indicated by Kalinga [8] and Kalinga et al [7]. Transformation techniques between models as provided by MDA accelerated a means of reaching to select OS LMS for customization.

The rest of this paper is organized as follows. Section two briefly gives the short theoretical background of OOSA&D, followed by an overview of MDA in section three. Section four shows how the two approaches were integrated to give a well defined process to follow in developing TanSSe-L system. Section five briefly provides a related work. Finally, this paper provides concluding remarks in section six.

2. OBJECT ORIENTED SYSTEM ANALYSIS AND DESIGN

Software Engineering discipline differs from software development. In software development the focus is on the software which is being developed and the outcome as desired. The term software development is also referred to computer programming, the process of writing and maintaining the source code [16]. As defined by Wikipedia [17], Pollice [13] and IEEE [6], software engineering is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software, and the study of these approaches. Software engineering addresses the life cycle in a systematic way. Schach [14] have highlighted the key criteria for software engineering to be: A well-defined methodology, Predictable milestones, traceability among steps, documentation and control, that is maintainability. To insist on that Liu [10] says that a software engineering is not to produce a working software system only, but also documents such as system design, user manual, and so on. To build good systems, it needs a well defined development process with clear phases of activities, each of which has an end product, methods and techniques for conducting the phases of activities and for modeling their products.

Several software development methodologies were employed. The two most commonly applied approaches are the structured approach and object-oriented (OO) approach. The main difference of the two approaches is that structured system analysis and design (SSA&D) approach examines an information system in terms of the functions it performs and the data it uses and maintains [18], [5] and [9]. The structured approach identifies the major functions or processes of a system, then breaks or decomposes each function down into its smaller composite steps [5] and [9]. On the other side OO decomposes the system into objects, that is, it examines the system in terms of the components in the system and how these components act and interrelate. The analyst first identifies the objects that comprise the system, then creates an object model which groups the objects into classes, and describes each class in terms of its attributes (or data), methods (or functions) and relationships with other classes [15]. The key idea of the OO approach is that the real world can be accurately described as a collection of objects that interact. OOSA&D is model driven approach where a number of diagrams can be created using the Unified Modeling Language (UML) to create different views in the process of development.

SDLC is the basis for most software development methodologies [4]. Pressman [12] adds to this by pointing out that a system's life cycle consists of four principal phases: requirement phase, Analysis phase, design phase and implementation/testing phase. UML is now widely used for modeling software at different stages during its development: requirement, analysis, design and implementation. Development of TanSSe-L system adopted the OO software development approach with UML. TanSSe-L was a system to be developed, and so it abided by the SDLC, which includes:

- System Requirement Specification where use-case diagrams was used
- Analysis phase where system requirements were analyzed using a conceptual model, system sequence diagrams and system operation contracts;
- System Design phase where collaboration and design class diagrams were used; and
- System Implementation, testing and deployment phase where an open source LMS was customized, tested and deployed for usage at Tanzania secondary schools

As stated, a well defined software development process as per software engineering was supported by the use of MDA, hence there was an intermediate step between the TanSSe-L system design phase and the TanSSe-L system implementation phase which involved transformations as applied in the MDA approach.

3. MODEL DRIVEN ARCHITECTURE

Design through modelling, which is a norm in the engineering domain, enforces a careful investigation of the structure, behaviour and architecture of a system in the early stages of development and promotes documentation and reuse. Czarnecki et al. [3] mention that the Object Management Group (OMG) initiative concerning MDA attempts to separate the application functionality specification from the implementation of that functionality on specific technology platforms. This approach is designed to play a key role in the fields of information systems and software engineering. MDA is supposed to provide a basic technical framework for information integration and tool interoperability based on the separation of the platform specific models (PSM) from the platform independent models (PIM).

The MDA guide indicates that the PIM represents the functionality and behaviour of a system and captures only the application logic. The PIM is then converted into the PSM, which captures the technology-specific details of system implementation. The separation of concerns between the application and technical aspects of a system promotes separate, yet controlled, evolution of both aspects of the system based on different needs. The primary goals of MDA are portability, interoperability and reusability through architecture separation of concerns [11]. Apart from PIM and PSM, there is also computation Independent Model (CIM) which comes before PIM aiming to capture the requirement in a problem domain. A key standard in the MDA is that it is based on UML as recommended by OMG. The set of layers defined includes the CIM, the PIM and the PSM. Along with the conceptual framework as shown in Figure 1, OMG through MDA has also provided a set of standards to express models, model relationships and model-to-model transformations [11].

The MDA guide explains each of these layers as follows:

- A ***computation independent model*** is the view of the system from the computation independent viewpoint, which focuses on the environment of the system, and the requirements of the system, while the details of the structure and processing of the system are hidden or are as yet undetermined.
- A ***platform independent model*** is the view of the system from the platform independent viewpoint, which focuses on the operation of a system while hiding the details necessary for a particular platform. A platform independent view shows that part of the complete specification that does not change from one platform to another.
- A ***platform specific model*** is the view of the system from the platform specific viewpoint. A PSM combines the specifications in the PIM with the details that specify how that system uses a particular type of platform. The platform specific viewpoint combines the platform independent viewpoint with an additional focus on the detail of the use of a specific platform by the system.

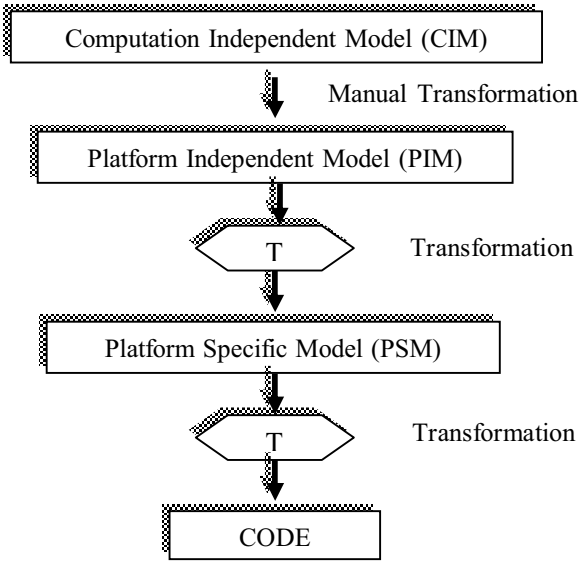

Fig. 1: MDA Conceptual Framework

4. INTEGRATION OF OOSA&D AND MDA

TanSSe-L system development integrates the OOSA&D approach with MDA as an added approach. When relating OOSA&D to the MDA conceptual framework, as shown is Figure 2, the CIM for TanSSe-L system is a modelling system showing specifications requirements. UML's use case diagrams are used to model the users' requirements and their boundaries. The CIM also shows the TanSSe-L system's architecture and standard specifications.

MDA guide [11] explains clearly that a common technique for achieving platform independence is to target a system model for a technology-neutral virtual machine. A virtual machine is defined as a set of parts and services (communications, scheduling, naming, etc.), which are defined independently of any specific platform and which are realized in platform-specific ways on different platforms. A virtual machine is a platform, and such a model is specific to that platform. But that model is platform independent with respect to the class of different platforms on which that virtual machine has been implemented. This is because such models are unaffected by the underlying platform and, hence, fully conform to the criterion of platform independence.

In view of the TanSSe-L system, the development of PIM targeted and focused on a global way of developing LMSs, specifically viewing the presence of open source LMS for customization, as one methodology of this research. Three main functional areas realized include:-

- Management of TanSSe-L system users
- Management of learning materials with learning activities
- How to track users' interaction with learning materials and learning activities, and generation of tracking reports

PIM as indicated in Figure 2 is related to the design class diagram (DCD) of the TanSSe-L system design phase as per the OOSA&D approach. System analysis and part of the system design phases is the transformation approach from CIM to the TanSSe-L system PIM (hereby a DCD). UML is used to model the TanSSe-L system at different levels of abstraction from the platform independent viewpoint. These models give what was expected to be provided by the TanSSe-L system for Tanzanian secondary schools. At this stage, the structure of the TanSSe-L system database through DCD was used to select the open source LMS platform. The selected open source LMS is a virtual machine specific to the TanSSe-L system PIM. Again to include the TanSSe-L system context, merging the model of the TanSSe-L system PIM with the TanSSe-L system-specific PIM was employed to get the actual TanSSe-L system-specific PIM.

When PIMs are based on virtual machines, transformation is not necessary [11]. Instead, it is the PIM of the virtual machine itself that needs to be transformed into a PSM for a particular platform. The selection of the TanSSe-L system-specific PIM also considered the PSM which the research used, that is a Linux, Apache, MySQL and PHP (LAMP). A summary of the PIM transformation into PSM is as follows:-

- DCD was used to study the functionalities, similarities and differences in the number of possible open source LMS to be customized;
- One was selected which would suit the TanSSe-L system's requirements, including the technology available and already prepared for the TanSSe-L system. The open source LMS selected was the TanSSe-L system-specific PIM;
- Needed major modifications to the specific PIM when related to DCD (TanSSe-L system PIM) were planned and documented; and
- The technological PSM for the specific PIM selected was understood in relation to the PSM for the TanSSe-L system.

During TanSSe-L system implementation, the following were done with reference to the transformation of the PIM into the PSM:

- The TanSSe-L system PSM was in place,
- The TanSSe-L system-specific PIM was installed,
- Models were merged to incorporate all missing information in the TanSSe-L system-specific PIM (selected open source LMS), and
- Continued customizing of the TanSSe-L system-specific PIM with the help of the documented plan.

An overview is that, the TanSSe-L system PSM depended on the platform implementation specification for the TanSSe-L system-specific PIM (selected and merged open source LMS). The implementation stage involved modifying the database, customizing codes, page interfaces and Structured Query Language (SQL) of the TanSSe-L system-specific PIM. Figure 3 gives a summary of the transformation from TanSSe-L system CIM - TanSSe-L system PIM – TanSSe-L system specific PIM – TanSSe-L system PSM.

5. RELATED WORK

"Model Driven E-learning Platform Integration" is the paper authored by Bizonova et al, [1]. This paper provides a generalized e-learning architectural framework enabling an integrated specification of platform architectures. This platform-independent framework can then be used to specify and classify existing or future LMS and to simplify migration of data between different kinds of e-learning systems.

Bizonova et al, [1] have compared PIMs of different existing open source LMSs to create a PIM that covers common functionalities of all LMSs. Their goal was to define a generalized model of the LMS system consisting of features of all other LMS systems that can be incorporated into it.

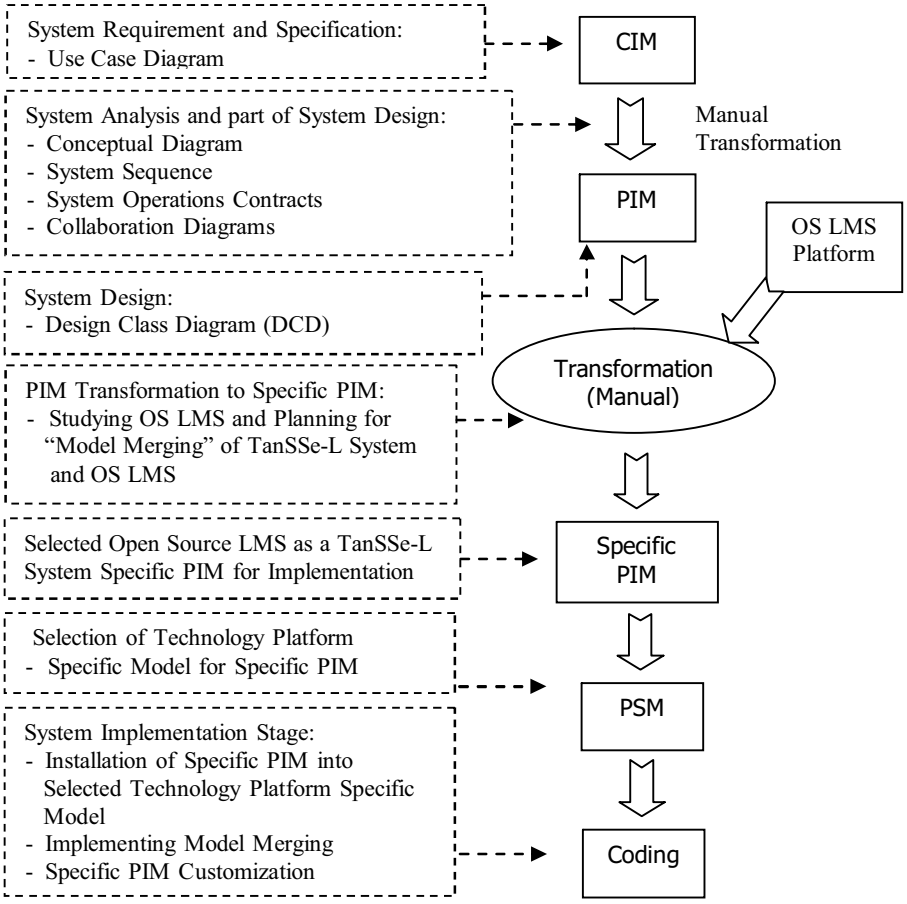

Fig. 2: TanSSe-L System OOSA&D Integration with MDA

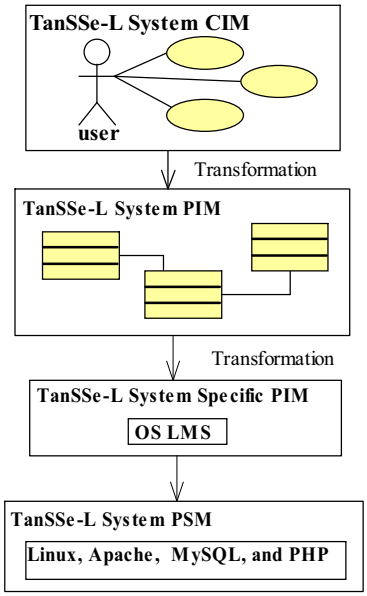

Fig. 3: TanSSe-L System PIM Transformation to PSM

The paper introduced a new strategy called the reversed MDA paradigm, by first using real LMS systems to define and abstract a model of a generalized LMS.

In the course of TanSSe-L system development, the approach used was first to create the DCD as a TanSSe-L system PIM using the OO system development approach. This TanSSe-L system PIM was then used to reach to the specific PIM and PSM in forward order.

6. CONCLUSION

There is a number of open source learning management systems developed which can be customized for a certain particular context. Selection of which one to customize can be very difficult since many of the OS LMS have similar functionalities and use the same technologies. When thinking on open source customization idea, it can be overlooked and taken as the issue of not needing software development concept, with no well defined process as is being insisted in software engineering discipline. This paper shows how customization cannot be effective enough if there will be no base to guide customization. A proper selection of OS LMS and the base for customization can be reached if there is a base system structure of the organization as a PIM and use that structure's functioning to compare with those of the OS LMS.

SDLC has well defined phases to follow for producing a software application. This work has included the involvement of open source customization in developing the TanSSe-L system. The work developed a TanSSe-L database structure to represent the Tanzanian secondary school context. It then, did a thorough evaluation of an open source LMS in accordance with the TanSSe-L system DCD. The areas needing to be customized were recorded and used as a guide during the implementation phase, when customization of the Moodle LMS was done.

Many of the existing open source LMS platforms can support the basic functions needed for the e-learning process, but a detailed evaluation of the specific areas needed to be customized has to take place. The idea is not to start customizing an open source LMS without knowing the specific context of the organization in detail.

Model transformations can be from CIM to PIM, PIM to PIM, PIM to PSM and PSM to code. A CIM can be refined and mapped to another CIM, in the same way PIM can be refined to another PIM. This paper explains the transformation of the TanSSe-L system DCD as its PIM into another PIM (OS LMS PIM) to get the TanSSe-L system-specific PIM.

7. ACKNOWLEDGEMENT

It is my pleasure to that all reviewers of this paper. Specifically non-blind reviewers Prof. Nerrey Mvungi and Dr. Alois Mvuma of the University of Dodoma in Tanzania

8. REFERENCES

[1] Bizonova Z., Ranc D. and Drozdova M., (2007), "**Model Driven E-learning Platform Integration**" , In 2nd European Conference on Technology Enhanced Learning EC-TEL PROLEARN 2007 Doctoral Consortium, Vol. 288, September, pp. 8-14.

[2] Collier, G. (2002), "**e-Learning Application Infrastructure**", Sun Microsystems White Paper, http://www.sun.com/products-n-solutions/edu/whitepapers/index.html, January.

[3] Czarnecki, K. K., Favre, J., Gogolla, M. and Mens, T. T. (2005), "**Essentials of the 4th UML/MoDELS Workshop in Software Model Engineering (WiSME'2005)**".

[4] Davis, A., Bersoff, E. H. and Comer, E. R. (1988), "**A Strategy for Comparing Alternative Software Development Life Cycle Models**", IEEE Transactions on Software Engineering, October, Vol. 14, No. 10, pp. 1453-1461.

[5] Davis, W. S. (1983), "**Systems Analysis and Design: A Structured Approach**", Addison-Wesley, ISBN: 0201102714.

[6] IEEE (1993), "**IEEE Standards Collection: Software Engineering**", IEEE Standard 610.12-1990, IEEE 1993.

[7] Kalinga E. A., Bagile B. B. R, and Lena Trojer L., (2007), "**Strategies for Developing e-LMS for Tanzanian Secondary Schools**", World Academy of Science, Engineering and Technology (WASET), International Journal of Social Sciences (IJSS), Volume 2, Number 3, Pages 145-150.

[8] Kalinga, E. A., (2008), "**Development of an Interactive e-Learning Management System (e-LMS) for Tanzanian Secondary Schools**", Blekinge Institute of Technology, School of Technoculture, Humanities and Planning, Licentiate Dissertation Series No. 2008:03, ISSN 1650-2140, ISBN 978-91-7295-134-1.

[9] Kendall, P. A. (1996), "**Introduction to Systems Analysis & Design: A Structured Approach**", ISBN: 0697124142, Irwin.

[10] Liu, Z. (2001), "**Object-Oriented Software Development Using UML**", March, The United University – International Institute for Software Technology (UNU/IIST), Report No. 229.

[11] OMG (2003), "**MDA Guide Version 1.0.1**", Document Number: omg/2003-06-01, 12th June.

[12] Pressman, R. S. (1997), "**Software Engineering: A Practitioner's Approach, Fourth Edition**", ISBN: 0070521824, Publisher: McGraw-Hill Companies.

[13] Pollice G., (2005), "**Teaching software development vs software engineering**", Worcester Polytechnic Institute, IBM, 15th December.

[14] Schach S. R. (2002), Object-Oriented and Classical Software Engineering, Stephen R., McGraw Hill, 5th edition, 2002, ISBN 0-07-239.

[15] Schach, S. R. (2004), "**An Introduction to Object-Oriented Systems Analysis and Design with UML and the Unified Process**", McGraw-Hill/Irwin, ISBN 0071215107.

[16] Wikipedia, (2009a), "**Software development**", Retrieved on June, 2009 from: http://en.wikipedia.org/wiki/Software_development.

[17] Wikipedia, (2009b), "**Software engineering**", Retrieved on June, 2009 from: http://en.wikipedia.org/wiki/Software_engineering.

[18] Yourdon, E. (1991), "**Sayonara, Once Again, Structured Stuff**", American Programmer, Vol. 4 No. 8, pp. 31-8, Publisher: American Programmer.