Journal of Theoretical and Applied Information Technology
15th July 2016. Vol.89. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

MODEL-BASED ARCHITECTING AND MODEL DRIVEN APPROACHES IN E-LEARNING SYSTEM DEVELOPMENT PROCESS: REVIEW OF LMSGENERATOR

RACHID DEHBI

Department of mathematics and computer sciences,

Faculty of Science Aïn Chock , Hassan II University, MOROCCO

E-mail: dehbirac@yahoo.com

ABSTRACT

For many years, various approaches in system design and implementation differentiate between the specification of the system and its implementation on a particular platform. People in software industry have been using models for a precise description of systems at the appropriate abstraction level without unnecessary details. Model-Based Architecting and Model-Driven (MD) approaches in system development process increase the importance and power of models by shifting the focus from programming to modeling activities. Models may be used as primary artifacts in constructing software, which means that software components are generated from models. Software development tools need to automate as many as possible tasks of model construction and transformation requiring the smallest amount of human interaction and why not the total automation of development processes in software engineering.

This article focuses on the model-based architecture and software engineering approaches needed for the automation in the software engineering development process. And as an illustrative example it represents the architecture and the functional core of LMSGENERATOR, a software factory for business components and platforms for distance learning.

Keywords: *Model-Based Architecting, model driven engineering (MDE), model driven Architecture (MDA), component-based approach, Learning Management System (LMS).*

1. INTRODUCTION

To promote growth and maximize the scalability of a growing company, it is necessary to set up an automation strategy for these key business processes that passes through the design of a business application suite developed specifically for these internal processes. Indeed, by standardizing, systematizing and optimizing the processes and the current operations, you give your business all the tools needed to reach full growth.

Therefore, the systematization of software engineering development process requires a global or partial care of the software packages guidelines implementation.

By implementing automation, optimization strategies and the systematization directives, the software engineering specialized companies release their internal workforce, which gains in efficiency and effectiveness, and they reduce risk and errors related problems. The company thus gains in value, increasing its production and delivery capacity significantly.

However, software engineering experts and especially the developers have acquired personal methods in the development of software applications for specific platforms. Nonetheless, these methods that focus on the code become obsolete when we produce solutions in a multi-development environment in which technological and functional needs are closely changing.

Certainly, the productivity of software development has always been one of the most important interests in the field of software engineering. After the Object Oriented Programming (OOP) [1] the Designs Patterns [2, 3], component-oriented programming or Service (SOA) [4], a new vision proposes to improve interoperability, portability and productivity of computer systems through abstract models. This is the promise of Model Driven Engineering [5], and especially model-driven architecture [6] (Model Driven Architecture -MDA), its implementation

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

initiated in 2003 by the Object Management Group (OMG).

To cope with the complexity of information systems and software development projects, this new software engineering trend has given rise to several tools approach and adopting the model-driven engineering. Their use is however responsible for processing tasks and repetitive manual generation and it is a source of error and is time consuming [7, 8, 9, 10, 11], which calls in question these approaches and makes their use heavy and uninteresting. Automating these transformation tasks can increase productivity. But the difficulty is to determine when automating repetitive manual task to be reused and at what level in the software development cycle. Research work [12, 13, 14, 15, 16, 17, 18] was done but we certainly do not cover the full cycle of software engineering and only support some repetitive tasks in the development process.

In this work, we focus on a specific approach [19, 20] of generic vacation which can be generalized in SEVERAL study area and based on model driven engineering coupled with methods and the most efficient software engineering approach. This approach can facilitate the integration and the automation of the all complex systems development stages and covers the full cycle of their implementation from analysis to deployment in the execution environments.

This approach introduces a new methodology to programming virtual learning environments. From a set of domain specifications [21], we will generate reusable and adaptable business components to different actor's needs, forming a reusable components patrimony. Then we'll assemble these components to generate complete solutions. This idea is used at different levels in the design and implementation of our system LMSGENERATOR [20], a software factory for business components and platforms for distance learning. This work shows how the model driven engineering (MDE) and the engineering of reusable components can be combined to develop such systems (LMS).

This article is organized as follows. The second section is reserved to the presentation of our approach. In Section 3, we present the features, actors, architecture and components of LMSGENERATOR, the implementation of our approach. The section 4 describes the two generation phases of this generator with a preview case study. At the end, we conclude with some issues related to this approach.

2. OUR APPROACH

Our approach accords with the research ideas on Domain Driven Development [22, 23] and on those about models and language for the design and handling of domain reusable components [24]. It is mainly based on the ability to define business models that meet the needs of the area specified in the previous section, and will be processed to generate business components and refined to a specific execution platform. These business components, in turn, will be assembled to generate virtual learning environments.

The methodological concept followed in our research is an approach with three basic steps:

2.1. Analysis and Design Step:

It sets up two extensible repositories [21], one for business models and the other for the execution platform models. The business models will be reusable, adaptable, flexible, integrating the various phases of the overall process of ODL and representing criteria of adaptability to different user groups, including an educational model that includes different teaching approaches. Similarly, models of execution platform will also be adaptable, extensible and open to technology developments.

In this step, shown in figure (Fig 1), the e-learning experts conduct a detailed analysis of the area to determine the functional requirements identified for reuse as an autonomous part of the domain model. Once these needs are identified, they will be designed as a conceptual component, according to the art of engineering components for reuse, and stored in our repository. It is based on the principle of decomposition of the model recently introduced in the Model Driven to reduce system complexity and to improve the reuse of their conceptual component engineering, by proposing "simply" that the parties to build and assemble should be not programs like it was always considered in Software Engineering but models.

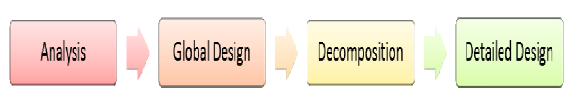


Figure1. The Analysis And Design Process Of Our Approach

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The process of this step is an iterative process of instructional design which partially resembles the ADDIE Model [25] process; see Figure 1, except that ours supports only the analysis part coupled with the process of decomposition to identify the autonomous design parts representing criteria for reusability to facilitate the design process.

This means that when analyzing the designer develops a clear understanding of "differences" between the desired outcomes or behaviors and existing knowledge of the auditorium and skills. The documents phase of the overall design specifying learning objectives will be broken down into conceptual entities representing unit needs in autonomous and reusable format specification that will be supported by the detailed design stage to deliver us conceptual models of the business models repository.

These models, once identified and designed, will undergo a generation process following the art of model-driven engineering to give us the executable software components in target environments. This generation process detailed in the second step is based on a technology repository containing definitions, meta-models, structures and portions of program code, that dependent on dynamic websites development technologies according to web 2.0.

The implementation of this technological framework is supported by the platform execution experts who will identify the technological needs necessary and sufficient to implement any dynamic web application. These technological needs will be expressed in several technologies to ensure, at the time of generation of the software components, the choice between several technologies, thereby ensuring adaptability to execution environments.

2.2. Generation step:

The generation process of the second step adopts Model Driven Engineering coupled with engineering for reuse, to generate reusable software components dependent on the target technology.

This step aims to contribute to setup a software component repository used as brick in the final generation process of virtual learning environments at the third step.

Based on the conceptual model of our business models framework [21], we can apply a set of transformation rules, derivations and successive enrichments to produce refined LMS software components, ready to be used as basic building blocks for the construction of platforms e-learning. This generation phase consists of two basic steps. The first one generate from a business model a component model regardless of the technology, while the second uses this model to generate a dependent component of a target component technology.

The figure (Fig 2) describes the generation process based on MDA approach. This generation step is based on three successive transformations. The first takes as input the business models of our business models repository and converts them into component models regardless of the technology. These are supported by the second transformation in order to transform them into a specific technological components based on platforms execution meta-models of our technological repository.

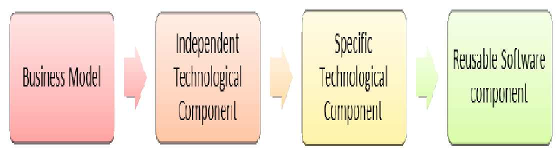


Figure 2. The Generation Process Based On MDA Approach

Once the specific technology components are generated, they are transformed into software components using a technological Template of our technological model repository. After the generated software components are engendered, they will supply a repository of software components that will be used as the basis for the integration and assembly process of the third step in our approach to generate finished solutions.

2.3. Integration and assembly Step:

The Business Components depending on the platform, generated in step 2, are used to generate LMS respecting the rules of the component engineering.

In this step, we define the structure of the platform that you want to generate, by identifying software components to be integrated, as well as the choice of technology used for their implementation. The use of these components to build our learning management systems is the "Component-based Software Engineering" [26], which focus more on reuse of COTS (software component) on the development process.

This is reflected in a series conducted in the generation process of a learning management system (PGLMS) and the assessment process

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

components (PEC) in our repository component, mutually impacting on each to define specifications system. This strong relationship between the two processes is shown in figure (Fig 3).

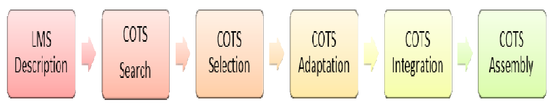


Figure 3. The Generation Process Based On Reuse Engineering

During the generation process, the research and selection stages consist in identifying one or more components to meet the functional needs identified and expressed in LMS description step.

The adaptation and integration stages consist respectively in adapting the component to the identified needs and integrate them into the final solution architecture. Once the selected components are integrated, they will be assembled to produce a coherent solution that will be integrated and deployed in a target runtime environment.

This allows an original idea that becomes our primary focus [20, 19]. This idea consists of modelling and implementation of a generator, named LMSGENERATOR. This generator is based on two approaches: The Model Driven Engineering to generate business components that run on target execution platforms and the component approach based on the assembly of business components, in order to build learning management system and deploy them in multiple environments.

3. LMSGENERATOR

3.1. Features and actors:

LMSGenerator [20] is an environment used for integration of business components and generation of other components from model. It aims the generation of platforms for distance learning, executable on multiple execution environments and representing criteria of adaptability to different user categories. Based on multi-layer and interconnected software components (Figure 5), this generator provides to its users (Figure 4) the ability to:

- Maintain a repository of conceptual components (business model repository) (PIM) durable, customizable, flexible and reusable as well as an extensible repository of platform execution model (PDM).
- Apply transformation, generation and refining rules into models (element of business model repository) in order to generate a business components runtime on the target platforms (using PDM).
- Integrate existing components in its business components repository.
- Describe and maintain models (structure) of learning platform and store them in a descriptive repository for use as needed.
- Generate distance learning environments based on predefined models in the platform descriptive repository (repository of LMS description) or by generating directly a new platform with the ability to save its structure in the dedicated repository.
- Deploy the environments in target execution platforms (on specific application servers and database servers).

This generator follows a methodology with two automatic generation process. The first one aims the generation of software components from the conceptual model by adopting a series of Matching, Mapping and Transformation operation, according to models driven engineering, in particular the MDA. While the latter adopts an automatic generation process based on reusable components engineering.

These two processes of automatic generation require the intervention of several categories of actors organized in three areas of expertise:

- E-learning Area: contains the e-learning designers involved in the conceptual models design process that feed our LMS business repository. They work in collaboration with experts of the engineering for reuse in order to identify the elements from the domain model may present an independent conceptual component. They also work with the modeling experts in order to formulate these models with a modeling language such as UML. These actors play a key role in our approach and a secondary one in our "LMSGENERATOR" system.
- Software Engineering Area: the generation process of our approach is based on model-driven engineering, and in particular MDA, and engineering of reusable components. The model-driven engineering involves two types of users, according to [27]: the MDA Experts and the MDA Designers. The MDA experts specialized in the management of

Journal of Theoretical and Applied Information Technology
15th July 2016. Vol.89. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

meta-models and platforms, and responsible for the generation of a transformation model, often expressed as rules. The MDA designers specialized in managing business models and their translations into a target platform using the transformation model implemented by the MDA experts. While the engineering of reusable components involves the engineering of reuse expertise. According to this type of engineering the task is the responsibility of specialists in research, selection, adaptation and integration of software components.

- Systems Engineering Area: contains the system expert who supports the configuration deployment of the solutions generated in the target application and database servers and feed the deployment description repository.

Since the generation process of our approach is automated through LMSGENERATOR, the roles of identified users are reduced to five:

- The e-learning experts who feeding the LMS business repository with the identified models, designed at analysis and design step, and expressed with the UML notation.
- The platform execution experts who supports the feeding of the technological model repository with the platform execution meta-models and their technological templates.
- The MDA experts who define the Matching, Mapping and Transformation rules between the source meta-model and the target meta-model.
- The CBSE (Component Based Software Engineering [26]) Experts who set the search, selection, adaptation and integration rules of the software components.
- The system experts that define the configuration deployment rules of complete solutions into the target execution environments.

To these five users expert involved in the generation process of our approach and whose functions are partially automated in the cores of our generator system, we add LMSGENERATOR administrator. This user is involved in the final generation process, once the user expert tasks are completed, and that, in order to generate the learning management systems and deploy them on the target runtime environments.

The figure (Figure 4) shows the use case diagram of LMSGENEARTOR which summarizes the functional relationships between its actors and their use cases.

3.2. Architecture and component:

LMSGenerator based on an infrastructure using a business model repository and a repository of platform execution models [32], whose alimentation is currently the subject of our research. This infrastructure is composed of a generator (Business Component Generator), an integrator (Integrator Component) of business components for a specific execution platform and a learning platform builder. The generator is based on model driven architecture approach and permitting, through mechanisms of transformation, to define and generate specific business components (brick), adapted to the construction of a learning management system.

Once the repository for business components is supplied by the generated or integrated bricks, these bricks will be assembled in the art of reusable components engineering by the learning platform builder to produce a virtual learning environment. The figure (Figure 5) identifies the software components that support the two generation phases of LMSGENERATOR detailed in the following Section.

Each phase involves one or more components of our generator that we have classified into three categories of software components according to the nature and domain of their intervention:

3.2.1 MDA support component:

This component category involved in the step based on the MDA approach, it contains two components the "Manager Prerequisites" components (PM) and the "Business Component Manager" (BCM).

The PM component is used by e-learning experts and by technological platforms experts in order to feed the two repositories that are the prerequisites of our system: the "LMS Business Model Repository" and the "Technological Model repository".

The BCM component supports three functions ensuring the production of reusable software components in the assembly step. These functions are: the transformation of the conceptual model into a model representing a software component regardless of the technology used for its implementation, the generation of executable software component depends on a target technology from the previous model, and the integration of existing software component in the repository of software components. These three functions are

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

guaranteed respectively by the following components: the "Business Model Transformer" (BMT), the "Business Componant Generator" (BCG) and the "Component Integrator» (BI).

The BMT component is responsible for the conceptual model transformation into a component model independent of the platform (PIM), thereby feeding the "Model Component Repository". The elements of that framework follow a process of transformation to generate software components items of the "Refined Business Components Repository".

This generation process is supported by the BCG component. Once the software component is generated, its description is stored in the "Description Component Repository» and it is ready to be used in the assembly process by CBSE experts.

3.2.2 CBSE support component:

The component category involved in the generation phase based on the component-based systems engineering (CBSE) contains two components the " LMS Descriptor Manager" (LDM) and the " LMS Integrator" (LI) . The LDM component is responsible for the management of LMS structures. The definition and configuration of these structures are supported by CBSE experts. These predefined templates are used to supply a "LMS Template Repository» used by the LMSGENERATOR administrator at the time of LMS generation. The LI component is responsible for the identification, selection, adaptation and assembly of software components elements of the" Refined Business Component Repository» and that by using the predefined structures stored in the LMS structure repository. The configuration of this process based on CBSE is the responsibility of the CBSE experts, while its triggering is the responsibility of the LMSGENERATOR administrator.

3.2.2 CBSE support component:

The "LMS Deployment" component supports the process of deploying the generated LMS. It has a management and description deployment module with the descriptive deployment repository (DDR).

This management module is used by the system expert to define deployment configurations associated with a set of available application and database servers.

These descriptive deployment files that feed the DDR repository will be used during the deployment of a generated LMS by the LMSGENERATOR administrator.

4. THE GENERATION PROCESS OF LMSGENERATOR:

4.1. Process based on MDA approach

This phase aims to build LMS business components representing criteria of adaptability and dependent on a target technology (target runtime). The starting point of this phase is the elements of our business repository. This repository is composed of business model designed to be used in the web. For this reason we opted that these models should respect the MVC paradigm (Model-View-Controller). This programming schema proposes the separation of the application into three parts:

- The model, which contains the logic and the state of the application;
- The view, which represents the user interface;
- The controller, which handles the synchronization between view and model.

The essential point of this paradigm is to separate the graphical objects of business objects, so they can evolve independently and reused. It is in this sense that we proposed in our approach a necessary and sufficient modelling process to effectively build our business components. For this, we used a subset of UML models to describe our business. Therefore our business model consists of three sub models:

- The business logic model to describe the business logic component.
- The application logic model to structure and synchronize between the business and the presentation of the component.
- The presentation model to define the user interface, regardless of the display means.

Each of these three models is structured under package format covering the various UML diagrams of the analysis and design stage and using a good presentation of the business model needs. Once the model is presented, it undergoes a generation process based on MDA approach supported by two software components of LMSGENERATOR: Business Model Transformer and Business Component Generator. Starting from a conceptual business model(element of Business Model Repository), we can apply a set of

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

transformation, derivations rules, and successive enrichments to produce a refined LMS business component(element of Refined Business Component Repository) , ready to be used as a fundamental building block in the construction of our learning platform. This generation phase consists of two basic steps. The first step aims to generate from a business model a component model independently of the technology expressed in XML file format, while the second uses the XML file to generate a business component depending on a target technology.

The first step is based on a generation process based on model transformations adopting the transformation by programming approach using APIs manipulating UML models provided in development environments. Thus, with minimal effort, a component model independent from technology can be quickly obtained by a series of transformations, mergers or refinements of the three sub models mentioned previously (Figure 6).

This transformation by programming approach is the charge of Business Model transformer (Figure 5). This component model, as close as possible to the needs of the application, specifying the services provided and required, enter in its turn in a generation process supported by Business Component Generator (Figure 5), adopting the transformation by template approach. This second step is based on a set of conceptual models for technology development defined in our technological model repository (Figure 5). Starting from the description under XML file format of the component independently of technology, Business Component Generator identifies the XML element and replaces its abstract description with a concretized description and dependent technology, until obtaining finished artefacts of the business component for a specific platform execution and the XML file description of the component [28].

This XML file contains a detailed description of the internal components of the business component (web pages, business classes, control classes and the SQL schema). It is also used to feed the component description repository used at the time of platform construction. Once business components are generated, they are refined and ready for use by the process of LMS generation of the second phase. The figure below, present a partial view of the generation process based on MDA approach.

Experimentally [29], each model of our business repository is represented with two rich information diagrams required for a build process. A class diagram that allows presenting the component according to these three perspectives respecting the MVC paradigm: Perspective of presentation, of service and training content perspective and the transition state diagram for representing the navigation model that will ensure navigation between different web pages forming our business component.

4.2. Process based on component approach

The purpose of this phase is the production and deployment of virtual learning environments in a target runtime. It is based on a generation process adopting the approach of reusable component engineering. Starting from a repository of LMS business components, generated in the first phase, we can proceed by assembling, adapting and refining to produce each time a new device for learning adapted to the needs of training organizations. The construction process of LMS begins once the process of generating business components completed. This process is triggered by the LMSGENERATOR administrator which begins the process of construction by the description of the desired platform. Under the format of an XML file, this description (structure) is a set of functional specifications meeting the descriptions generated bricks of business components repository (phase1). Once completed the step of description, the process proceeds by building identification of the specified components in the structure for generating a LMS learning device. This process of LMS generation brings these bricks, and then refines and integrates them to build a device that will be deployed by the deployment manager on the target environmental execution.

This process of generation and deployment is supported by three LMSGENERATOR software components that are: LMS descriptor manager, LMS integrator and LMS deployment. The figure (Figure 6), represent a complete view of the generation process supported by LMSGENERATOR.

Experimentally, the component approach process consists of three main phases: the Assembly process, the description process and the integration process. These phrases are sequenced as shown in figure (Figure 7).

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Figure 7: The Component Approach Process

4.2.1 Assembly process:

The assembly process adopted in this part is based on two models: a communication model ensuring communication between different software components of the e-learning platform and a navigation model that shows the links between the different pages of the assembled components. The first model identifies the communication ports and the objects exchanged between the different components, so that the second describes the navigation properties between their pages. In the following we will introduce the assembly process associated with our case study that aims generating an e-learning platform composed of three components: authentication component, discussion forum component and learning Management component.

4.2.1.1 Elaboration of the communication model

Once the three components associated with our case study are generated with the same technology, based on their corresponding conceptual models, they will be assembled in the art of the reusable components engineering.

According to this type of engineering, generated software components can communicate and collaborate together, via an exchange of object ensures via a communication port for each component. In our case the Learning Management component and the discussion forum component depend on the session object generated at the time of authentication process by the authentication component. This object is used throughout the navigation platform by these two components in order to manage the confidentiality and the adaptation of content to users. Figure (Figure 8) corresponds to the communication model associated with our case study in which the two components are connected by an arm's length that shows the direction of dependency and the object "session" exchanged between these components.

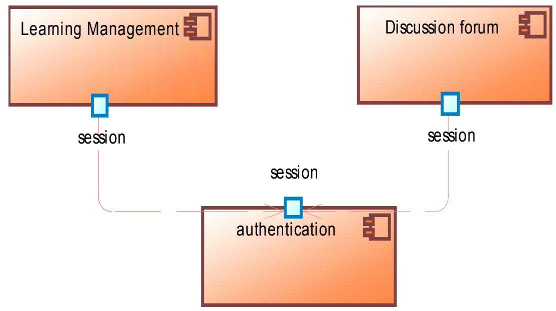


Figure8. Communication Model Of The Prototype Platform Components

4.2.1.2 Elaboration of the navigation model

The navigation model describes the navigation properties between pages constituting the e-learning platform presentation. It offers users the ability to view these features in a coherent, clear and honest context. So the principle is simple, because each full component has its own navigation model. Therefore, the operation is to unify these navigation models in a single one. For this we have the choice between two navigation strategies: Serial navigation and parallel navigation.

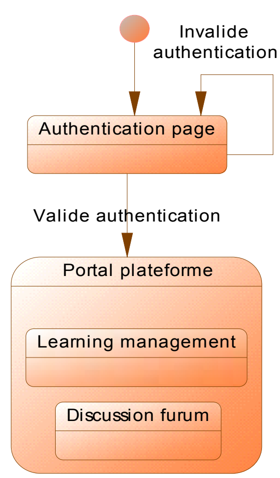


Figure 9. Parallel Navigation Model

The figure (Figure 9) corresponds to the navigation model associated with our case study according to the parallel optical navigation. According to this perspective, the generated platform uses the default page of the authentication component as Start Page. This page prompts users to enter their login and password. Then check their validity, if it is an invalid user, it will remain on the same page, otherwise it is redirected to a welcome page of the platform.

Journal of Theoretical and Applied Information Technology
15th July 2016. Vol.89. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

This page has content area reserved to place the content of each embedding object component. In our case this page will contain two zones, one for the learning management component and the other for the discussion forum component. The order of appearance of these two components is determined by a priority set by the administrator or CBSE expert at the time of the LMS structure generation.

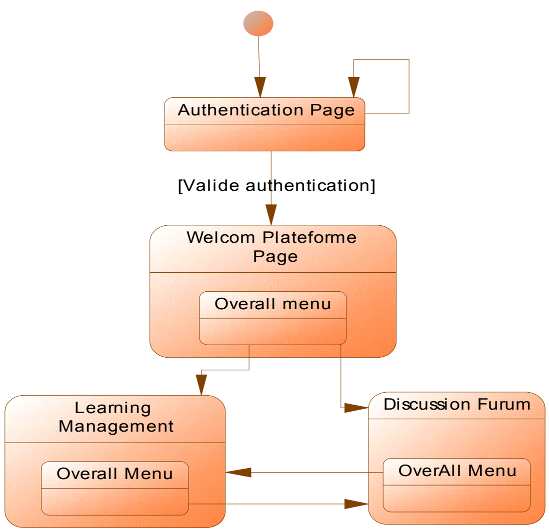


Figure10. Serial Navigation Model

The model in Figure (Fig 10) shows the standard navigation model associated with our case study. Such as parallel browsing, this model uses the authentication page as the start page of our platform. Once the user is authenticated, they are redirected to a welcome page containing a comprehensive menu that includes navigation menus associated with the two components object of study. These two components, in turn, must have the same menu to ensure consistent navigation. The only transformational character change completed during the assembly process is the replacement of the navigation menus of the components with the overall menu.

4.2.2 Description process:

The LMS Templates creation operation which includes the definition of the communication and the navigation models is supported by the "LMS Descriptor Manager" component. This component via an interface provides to the LMSGENERATOR administrator the ability to define the structure of the desired platform. Once the structure is created, it is stored in the LMS structures repository, pending their use in the component integration process of "LMS Integrator". The figure (Figure11) shows the appearance of this structure in XML format with an optical parallel navigation.

```
<?xml version="1.0" encoding="utf-8" ?>
<LMStemplate name="LMSPrototype">
 <CommunicationModel>
  <Component name="Authentication"
path="//BusinessComponentRepository/AuthenticationComp
onent/*">
      <Output type="Objet" value="session"
targetComponent="ForumComponent|ElearningComponent"
></Output>
  </Component>
  <Component name="Forum de discussion"
path="//BusinessComponentRepository/ForumComponent/*"
>
   <Input type="Objet" value="session"
sourceComponent="Authentication"></Input>
  </Component>
  <Component name="gestion d'apprentissage"
path="//BusinessComponentRepository/ElearningComponent
/*">
   <Input type="Objet" value="session"
sourceComponent="Authentication"></Input>
   </Component>
 </CommunicationModel>
 <NavigationModel type="parallèle">
  <WelcomePage Component="Authentication"
name="Login.jsp" />
  <PortalPage name="LMSPortal.jsp">
   <PortalZone order="1"
content="ForumComponent"></PortalZone>
   <PortalZone order="2"
content="ElearningComponent"></PortalZone>
  </PortalPage>
 </NavigationModel>
</LMStemplate>
```

Figure 11. Structure Of The Platform

4.2.2. Integration process:

Once the assembly strategy is developed, through the implementation of the desired e-learning platform structure, the integration and assembly process begins. This process is supported by the "LMS Integrator" component.

This component receives as input the structure of the platform and begins with the identification of components to be integrated with the elements "Component" of the communication model and in particular the value of "path" attribute. During treatment of the LMS structure, "LMS Integrator" component identifies the component from software component repository. Then he selects and place them in a folder with the name as the value of the "name" attribute of the "LMSTemplate" element.

Then it begins to integrate the other one based on communication and the navigation models

Journal of Theoretical and Applied Information Technology
15th July 2016. Vol.89. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

according to the LMS structure of Figure (Figure11).

The figure (Figure12) shows the result of the assembly process object of our case study which the generation process is based on the J2EE environment. The generation process is based on the model-driven engineering has already been explained in the related work.

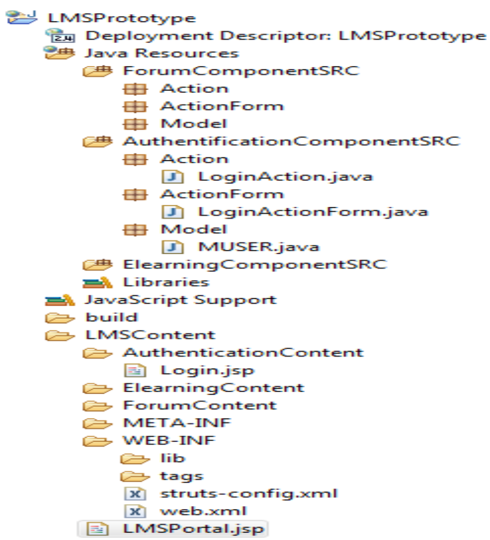


Figure 12. The Applicative Component Of The Lmsprototype Plateforme

According to this figure which represents the result of the integration process, the classes Action, ActionForm and the business classes (model) are grouped by component respectively in the following folders packages: "ForumComponentSRC", "AuthentificationComponentSRC" and "ElearningComponentSRC". The presentation part is also organized by component in the "LMSContent" folder via the sub-folders "ElearningContent", "ForumContent", "AuthentificationContent". The platform configuration file (struts-conFigurexml) is the assembly result of the sections: "form-bean", "action-mapping" files and "struts-conFigurexml" files of each component. While the overall deployment file (web.xml) is the same of each component except what the default page of the platform which must correspond to the "login.jsp" page. While the database creation file corresponds to a global file in which the files associated with the various components are called explicitly.

5. CONCLUSIONS

Our approach facilitates the design of Virtual Learning Environments, through their construction by assembling components generated from model or from components directly integrated into the Business Component Repository. It also accelerates their development and deployment by the principle of software reuse, as it facilitates their development by providing a clear separation between specification and implementation components. The idea of adopting such an approach comes from the will to implement the reuse of properties, cooperation, adaptability and interoperability. Reuse reduces cost and time for design, implementation and maintenance, adaptability makes the environment flexible while cooperation and interoperability makes it open. The main advantage of our approach is the consideration of future developments in technology or domain. These developments are supported by feeding the technological model repository with meta-models, the new transformation rule and the technology template of target execution environment. The generators associated with the new input will automatically propagate changes across all environments and components products.

We have presented in this work, our approach and an overview of LMSGENERATOR, its software architecture with its two generation phases. The assembly process was detailed with an overview of an experiment. Future work will include the generation of this approach into other fields by introducing ontology and by extending our technological model repository by using mobile technologies.

REFRENCES:

[1] T Rentsch. (1982). Object oriented programming, *ACM Sigplan Notices.*

[2] JO Borchers. (2001). A pattern approach to interaction design. *Ai & Society, Springer.*

[3] E Gamma, R Helm, R Johnson, J Vlissides. (1993); Design patterns: Abstraction and reuse of object-oriented design, *Springer.*

[4] J. Rouillard, Thomas Vantroys, V. Chevrin.. (2007) Les Architectures orientées service. Une approche pragmatique des SOA. *Editions Vuibert.*

[5] J. Bézivin, "In Search of a Basic Principle for Model-Driven Engineering", *Novatica Journal, Special Issue,* March 2004.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[6] X.BLANC. (2005). MDA en action: Ingénierie logicielle guidée par les modèles *(Eyrolles ed.).*

[7] Rhanoui, M., & El Asr, B. (2012). An analysis study of contract-based approach for quality-driven service component architecture. *Journal of Information & Systems Management,* 3(3).

[8] Moalla, S., & Yangui, S. (2011). Discovery of capacity-driven web services. *Journal of Intelligent Computing,* 2(2), 81–96.

[9] Meligy, A., Diab, H., & Tork, M. (2012). Security requirements engineering using outsourcing. *Journal of Information Security Research,* 3(3), 103–115.

[10] Lei, O. (2012). Applications of numerical calculation in engineering design. *Journal of Digital Information Management,* 10(5), 308–311.

[11] Evan, E. (2003). Domain-driven design: Tackling complexity in the heart of software. *Addison-Wesley.*

[12] W Alouini, O Guedhami, S Hammoudi, M Gammoudi, D Lopes. (2013). Semi-Automatic Generation of Transformation Rules in Model Driven Engineering: The Challenge and First Steps, *International Journal of Software Engineering and Its Applications (IJSEA).*

[13] DV Panfilenko, K Hrom, B Elvesæter, E Landre, S Hammoudi . (2012). Model Transformation Recommendations for Service-Oriented Architectures. *ICEIS (2),* 248-256.

[14] E. Rouillé, B. Combemale, O. Barais, D. Touzet, and J.-M. Jézéquel. Leveraging CVL to Manage Variability in Software Process Lines. *In APSEC,* pages 148–157, 2012.

[15] E. Rouillé, B. Combemale, O. Barais, D. Touzet, and J.-M. Jézéquel.Improving Reusability in Software Process Lines. *In SEAA,* pages 90–94, 2013.

[16] E. Rouillé, B. Combemale, O. Barais, D. Touzet, and J.-M. Jézéquel. (2013). Integrating Software Process Reuse and Automation. In *APSEC*, pages 380–387.

[17] Sun, W., Guo, C., Zhang, P., Zhang, N., & Wang, H. (2011). A dynamic role-based authorization model in grid environment. *Journal of Digital Information Management,* 9(6), 242–248.

[18] Cong, X., Zhang, H., Zhou, D., Lu, P., & Qin, L.(2010). Lecture Notes in Computer Science: Vol. 6249. A model-driven architecture approach for developing e-learning platform. Entertainment for Education, *Digital Techniques and Systems* (pp. 111–122). doi:10.1007/978-3-642-14533-9_12

[19] Rachid DEHBI, Mohamed TALEA, Abderrahim TRAGHA. (2013).The generation approach of Multi-target learning management system. *ADVANCED SCIENCE LETTERS.* volume 19, number 8, pp 2326-2330 ;

[20] Rachid DEHBI, Mohamed TALEA, Abderrahim TRAGHA. (2012). LMSGENERATOR: multi-target learning management system generator based on Generative Programming and Component Engineering. *2012 International Conference on Education and e-Learning Innovations, ICEELI 2012.*

[21] Rachid DEHBI, Mohamed TALEA, Abderrahim TRAGHA. (2012). The modeling elements of LMSGENERATOR business repository. CiSt 2012 - *Proceedings: 2012 Colloquium in Information Science and Technology.*

[22] Z. I. BATOUTA, Rachid DEHBI, MOHAMMED TALEA, OMAR HAJOUI, "Multi-criteria analysis and advanced comparative study between automatic generation approaches in software engineering". *Journal of Theoretical and Applied Information Technology (2015). Volume 81, No 3. pp 609-620.*

[23] C. Atkinson, T. Kühne, "Model-Driven Development: A Metamodeling Foundation", *IEEE Software,* September 2003.

[24] P. Ramadour. Modèles et langage pour la conception et la manipulation de composants réutilisables de domaine. *thèse de doctorat à l'Université d'Aix-Marseille III,* Le 17 décembre (2001).

[25] ADDIE MODEL, ISU College of Education. [Online].Available: http://ed.isu.edu/addie/index.html

[26] L. Crnkovic and M. Larsson. (2002). Building Reliable Component-Based Software Systems. *Artech House Publisher.*

[27] S. Hammoudi, S. Vale and D. Lopes, (2008). Vers un processus semi-automatique de la transformation dans l'approche MDA. *IDM'2008.*

[28] Rachid DEHBI, Mohamed TALEA, Abderrahim TRAGHA. (2012). LMSGENERATOR: The contribution of XML technology and model transformations. *2nd International Conference on Innovative Computing Technology, INTECH 2012.*

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[29] R. DEHBI, M. TALEA and A. TRAGHA. (2013). MDA-based transformation of LMS Business Components: The contribution of XML technologies and model transformations. *International Journal of Enterprise Information Systems.* Vol. 9, No. 4, pages 67-88.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Figure 4 Use Case Of LMSGENERATOR

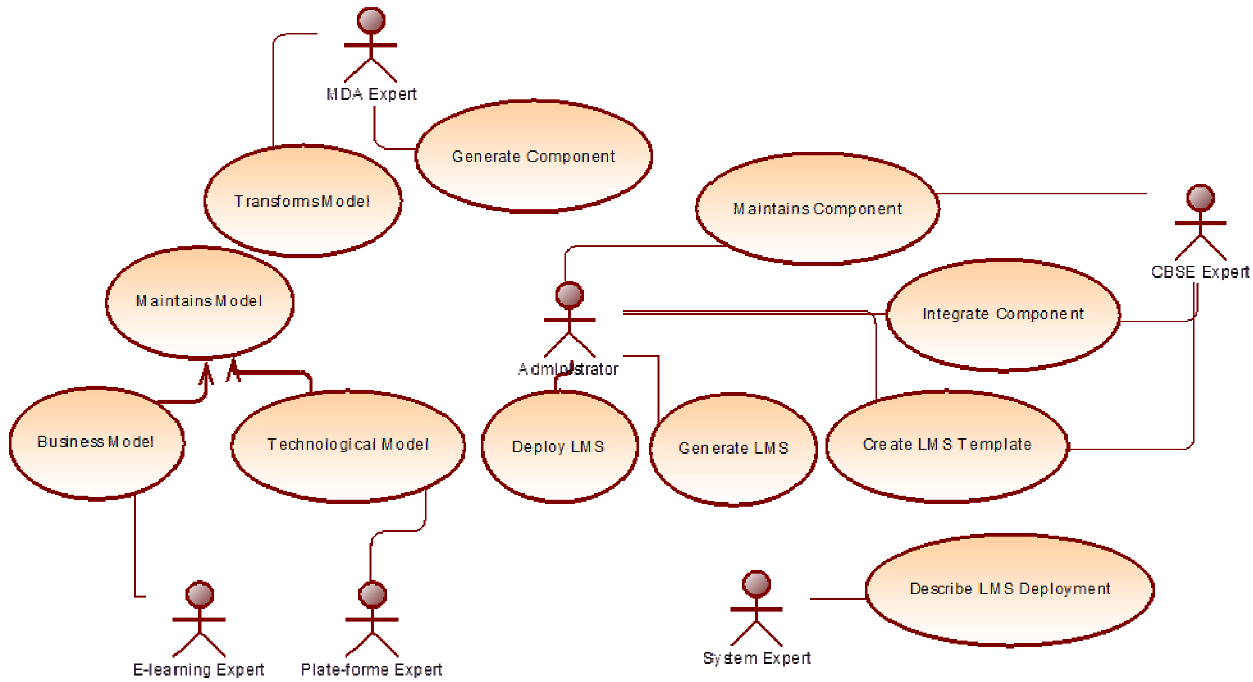
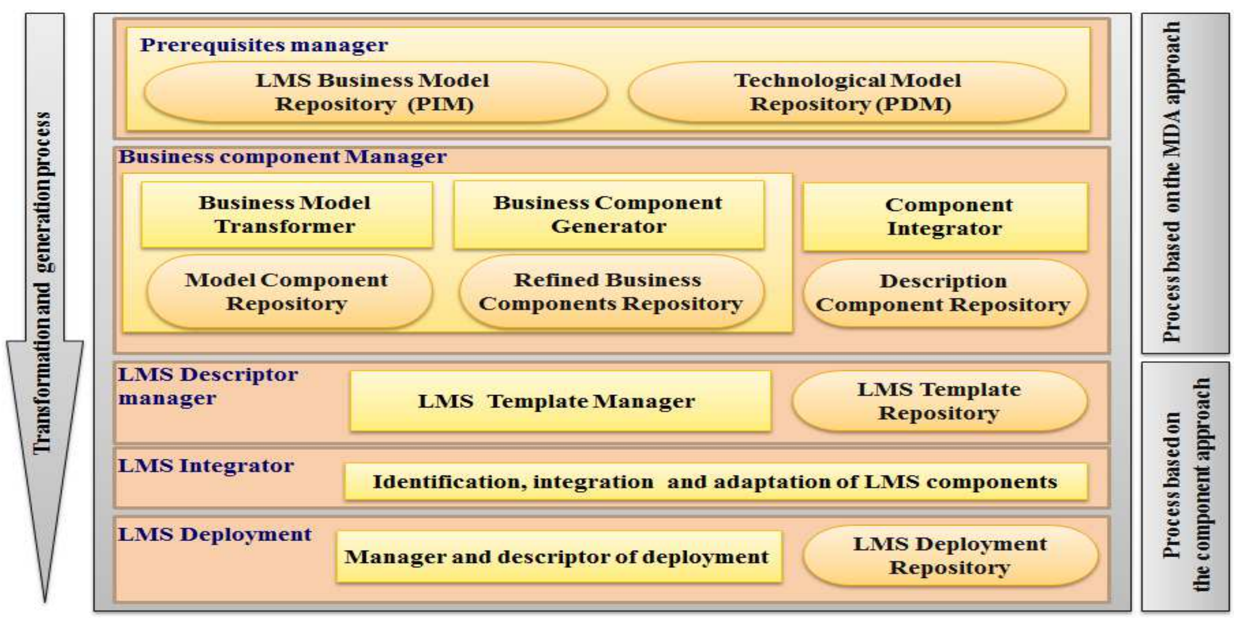


Figure. 5: LMSGENERATOR Architecture.



ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Figure. 6: Generation Process Of LMSGENERATOR