MODEL DRIVEN DEVELOPMENT OF ONLINE BANKING SYSTEMS

Avornicului Mihai-Constantin
„Babeş–Bolyai" University Cluj-Napoca, Faculty of Economic Science and Business Administration

Bresfelean Vasile Paul
„Babeş–Bolyai" University Cluj-Napoca, Faculty of Economic Science and Business Administration

In case of online applications the cycle of software development varies from the routine. The online environment, the variety of users, the treatability of the mass of information created by them, the reusability and the accessibility from different devices are all factors of these systems complexity. The use of model drive approach brings several advantages that ease up the development process. Working prototypes that simplify client relationship and serve as the base of model tests can be easily made from models describing the system. These systems make possible for the banks clients to make their desired actions from anywhere. The user has the possibility of accessing information or making transactions.

Keywords: MDA, UML, Online Banking, Class diagram, Platform Independent Model

Cod JEL: C88, M15

1. Introduction

Data intensive web based systems, that in the most of their uptime do data processes have become a part of our everyday life. Several systems of this type appeared on the internet: webshops, online ticket shops, banks, etc. These are used everyday by users around the worls.
It case of these systems design the target platform is choosen in the beginning as well as the technology used for the development. These are usually platform dependant application. After the development process, during the implementation process it oftenoccures that the system exceeds its bounadries and a new platform is needed. Often in this case the whole system needs to be recoded.
The importance of the present research consists of the fact that we can plan a platform independent banking system which can be implemented on any kind of platform. Furthermore offers the opportunity of generating a navigation model from the class diagram for the developers, giving real support to both developers and users. The final model can be applied to mobile applications too.

2. Model Driven Architecture

Model Driven Architecture (MDA) is a newtype of view on application development. According to MDA you have to focus on the functionality and behaviour of the studied system, putting aside the technological environment. This way an application has to be modelled only once (Anand et al., 2010).
The starting point is given by the domain model (CIM – Computational Independent Model) which can be expressed by use cases like business objects, activities and tasks. This model is refined with thorough examination; this is how the PIM (Platform Independent Model) is formed. This layer already contains the objects specifications, description of their cooperation and expresses the object states and changes. Taking the environment into consideration the PIM undergoes some more abstraction, this time giving precise specification on the objects, their

behavior, interoperability and reusability (Kennedy, 2006). The model layer that is to be implemented is stated. The Platform Specific Model (PSM) is the base of the components creation and functioning plan. MDA separates application architacture from system architecture. Application architecture contains components and information that specificate the applicatipons functional targets. System architecture contains the low level components and links, that make possible the application architectures axecution. This separation is the base element of MDA.
Recoding of the application is not necessary, when a new technology appears, the application is simply regenerated to the new environment.
For a lot of popular platforms, like CORBA or .NET; PIM→PSM can be generated. OMG-s standard, MOF (Meta Object Facility), XML (Extensible Markup Language) and CWM (Common Warehouse Metamodel) ensure that MDA is a complete software development method (Raffai, 2005).

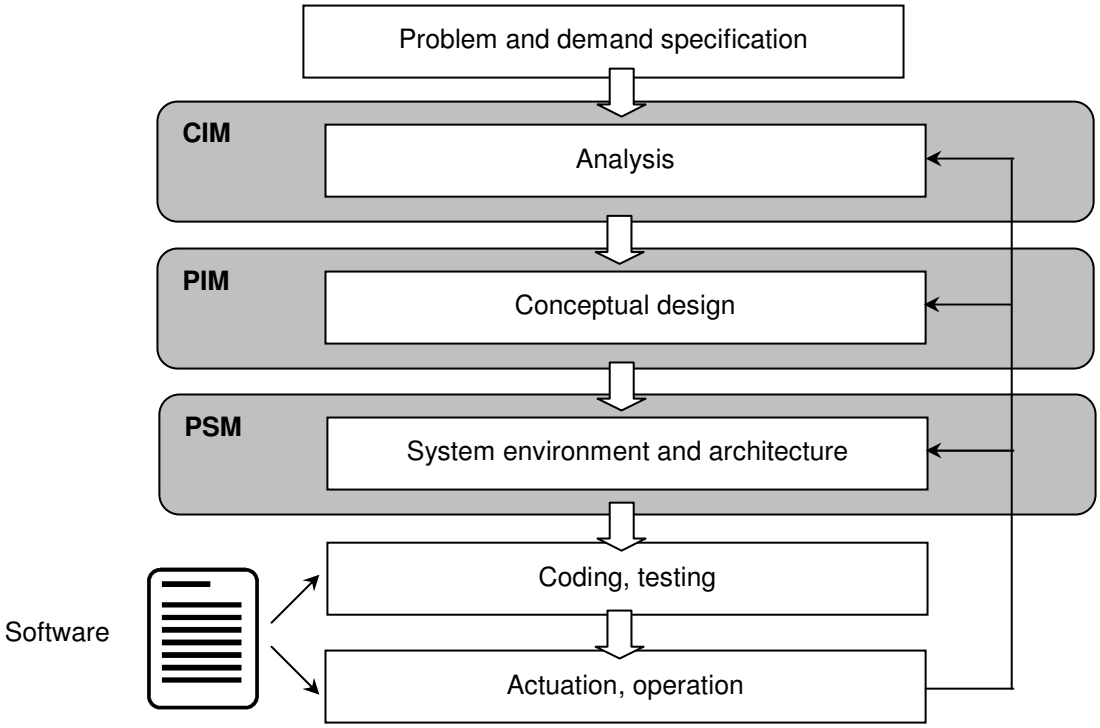

Figure 1: Model driven development

The lifecycle of online system developments base does not vary from the common method. Tasks have to be completed through the same steps, but theorys define certain principles that define the products of different steps and spesificate the models to be made.
MDA can be separated into four parts (Raffai, 2005; Anand et al., 2010):
1. As a first step the system has to be understood and the CIM model devfeloped according to it (Kennedy, 2004). This models specification is usually a document signed by the user, containing diagrams and tables. So the produst is a document that sepcifies the users demends from the application.
2. The product of the second step is an abstractization, that is still technology independent. The PIM model contains specifications on the system from the point of view of the most effective solution, it contains metadata from elements of the real system, these connections, structures,

elements behaviour, tasks, interfaces and dependancy between elements. The making of the PIM model is an iterative process, elements and their specs are refined continously, they are tested for their funtionality and compatibility with the system.
3. The next step is the definition of the environment and the transformation of the PIM model into PSM. This means that in terms of the environment we assosiate technology to every PIM element. More PSM models can be made from a PIM model, in condition of the platform that it is realised on.
4. The final step is the realization of the PSM model, the execution of implementation and testing processes, the creation of the whole model of the systems and giving information on its functionality.
I case of classic development, transformation of different models is made manually, while in model driven view these can be automated, speeding up the compilation process. Definition of transformation rules and their coherent use serve the solution of several problems. This way the modification of different applications can be made quick and documented, cooperating software can be attached. Implementation of concept models working on different platforms is more simple.

3. Methodology Proposed

We intend to describe the advanced design of online banking system by using the MDA paradigm. The standard development process consists of 4 steps, as previously presented. The research has been made through a case study, in which we design an online banking system, taking in consideration the next steps:
1. Demand analysis: by which the system's CIM modell is designed;
2. Analysis and planning: at this point PIM model is designed conform CIM;
3. Realizing the system: finally PIM model is changed into a platform depended model which reffers to the system's implementation;
These steps will be presented in the followings.

4. Case study: Online banking system

The system gives possibility to the banks clients to do their banking from anywhere. When a client opens a bank account he receives a username and a password, so that he can log into the system. This is a web based system; therefore the client can access the system from his home, workplace or anywhere. He has the possibility of interrogating he's account or making transfers.
With the password given by the bank the user signs in on the banks website. Other personal information is given, like name, birth date, e-mail, etc. The server checks for credentials and logs the user in. The user can access different services.
It is possible to interrogate information on accounts, traffic for a period or to make transactions and pay bills. The system has to approve every step for security. When the client is done he signs out.

4.1. Demand analysis

Demand analysis is the process throughout which the type of the system to be designed is specified. The demand is a supposed ability that the system has. Demand analysis is not a simple task, especially in case of web applications, because these raise other difficulties:
- An application has to have more entry points;
- Only one detailed information can be offered to the buyer;
- Environment changes quickly;
- There is no predefined development target;

The development process always starts with the demand analysis, which contain functional demands, actors and their tasks (Avornicului et al., 2006). This is the part that is called CIM (Computational Independent Model).
Usage cases help to maintain our demands from the system and to ensure their integrity. Usage case diagrams describe a certain part of a systems behavior from the point of view of external actors. Actors are those who use the system or are responsible for their maintenance.
Actors of online banking systems:
- User: the person that uses the banks online services.
- A bank server: the user accesses his account and registers tasks through it. The server sends the information the user asks for.
- Bank account: the online version of the client's bank account. It is a separate actor because every information and transaction depends on it.
- Recipient: in case of transfers the amount of money is transferred to the account of the recipient.
The use case diagram is as follows:

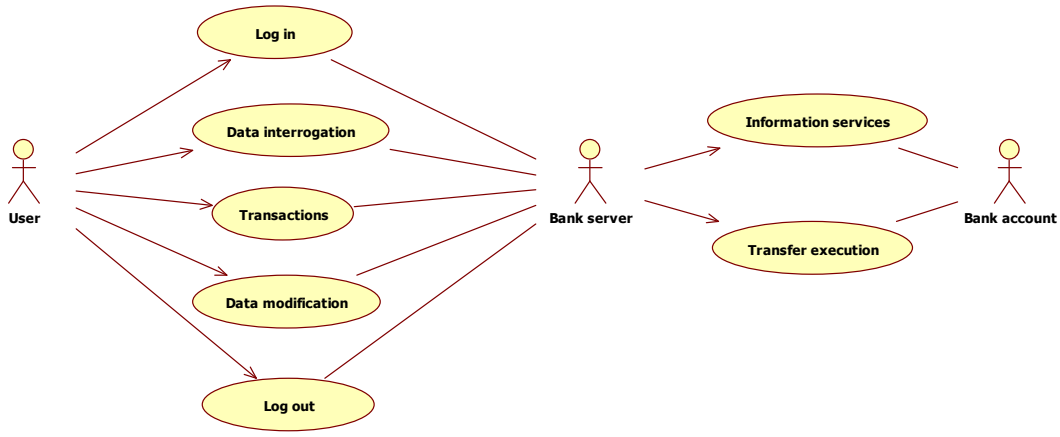

Figure 2: The use case diagram

Usage cases represent a chain of events, every mode that the client can use the system. The main parts of the system are the user and the bank server, executable tasks gather around them. Usage cases show how actors can use the system, what actions they have.
Interrogation and transaction use cases integrate more concepts. To avoid complex diagrams we will use the following keywords instead of drawing every detail separately:
- Sign in / sign out: the user signs in with his username and password into the internet banking system and signs out after he has done his work.
- Interrogations: the system enables the enquiry of information. This information can be: balance, traffic details, account information (date of opening, type, account number). Information is given by the server.
- Change of personal data: the user can change his personal data.
- Transactions: transaction can be made with the help of the server; these transactions can be transfers, payments of public services and other bills.
- Transfers: in case of transaction, the bank puts money from the users account to the recipients account.

4.2. Analysis and planning

The target of the analysis and planning processes is to prepare the plans based on the demand analysis that serves as the base of the web application. Throughout the analysis tha non-

functional demands have to be taken into consideration, during the planning functional demands can be dealt with as well.
While planning the wb application, principles of MDA will be followed. The most important is the conceptual planning, while we bulid a model that contains concepts defining the application. The concept model is made from the CIM model, that contains specifications concerning the professional field. This is considered a PIM (Platform Independent Model) for it does not take into consideration any specific technology (Avornicului et al., 2006).

4.2.1. Class Diagram

The class diagram serves for the static illustration of the system, presenting the classes and their bondaages. Es a first step the objects are defined and then they are grouped into classes by their behaviour.
In case of an internet banking application the following classes are used:
- User: The users class contains information on the banks clients. The clients can be identified and logged in.
- Account: Contains information on the users accounts.
- BankServer: The Bank Server class contains the detail necessary for the banks operations.
- Transaction: Transactions that the system can handle.
- Verification: Control class validates the users ID and password.
- LogInWindow: The user logs in through it.
- MenuSystem: The menu is an interface between the systems options and the users screen.

After the definition of the classes, their extensions, connections and multiplicity the class diagram is built.

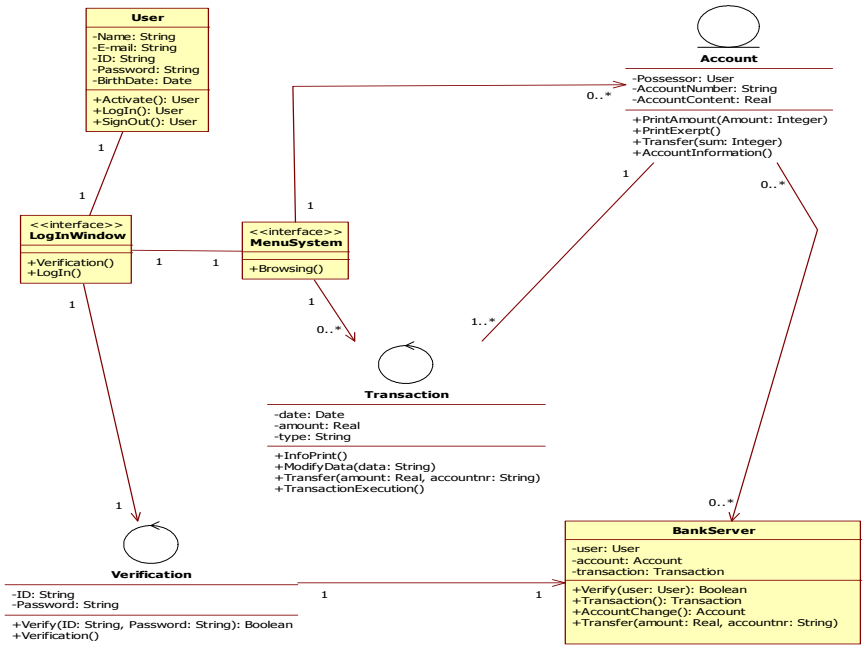

Figure 3: The class diagram

The class diagram shows the building blocks of any object-orientated system. Class diagrams depict a static view of the model, or part of the model, describing what attributes and behavior it has rather than detailing the methods for achieving operations (Anand et al., 2006).
In the conceptual design of a system , a number of classes are identified and grouped together in a class diagram which helps to determine the statical relations between those objects (Frankel, 2003).

4.3. Realizing the system

Following the views of MDA this step is the transformation of PIM models into PSM, platform specific models. The transformation of platform independent models has more phases in terms of the point of view from which we analyze the web application. The goal of every transformation is to pair every action of the system with a technology capable of dealing with it. On certain platforms parts can be exchanged without affecting the systems functionality.

5. Conclusion

Designated time for design has been drastically reduced, so new solutions are needed to be capable of keeping up with demands in the world of developing technologies. Model driven architecture delivers tools that simplify the process of application development.
MDA follows and embeds the most technical standards, ensuring the developed systems flexibility and renewability. The most important fact though is that using MDA, software engineers and business professionals can communicate in a common language and reach goals using high end technologies.
The designed system can be implemented on any platform. Moreover every bank can personalize it, if necessary it can be moved on mobile devices.
The model-based development's only problem is that configuration data for example: platform definition data does not differ from modeltransformation because during.

Acknowledgement

This work was partly supported by the CNCSIS **TE_316** Grant "Intelligent methods for decision fundamentation on stock market transactions based on public information".

References

1. Anand Rao A., Madhavi K., *Framework for Visualizing Model-Driven Software Evolution and its Application*, International Journal of Computer Science Issues, Vol. 7, Issue 1, No. 3, January, 2010
2. Avornicului C., Avornicului M., *Aspects of using MDA paradigm in sistem development*, Proceeding of International Conference on Business Information Systems, Iaşi, 2006
3. Avornicului C., Avornicului M., *Managementul şi proiectarea sistemelor informatice de gestiune*, Editura Risoprint, Cluj-Napoca, 2010
4. Frankel D. S., *Applying MDA to Enterprise Computing*, Wiley Publishing Inc., Indianapolis, 2003
5. Hazra T. K., *MDA brings standards-based developing Modeling to EAI Teams*, Application Development Trends, 2002
6. Kennedy A., *Model Driven Architecture and Executable UML – The Next Evalutiopnaryx Steps in System Development?*, Szoftvertechnológiai Fórum, Budapest, 2004
7. Raffai M., *Az UML 2 modellező nyelv* Szakkönyv Kiadás alatt, Palatia Kiadó, Győr, 2005