

A methodology for CIM modelling and its transformation to PIM

BOUSETTA Brahim*, EL BEGGAR Omar, GADI Taoufiq
LAVETE Laboratory, FSTS, Hassan 1st University, Settat, Morocco

* E-mail of the corresponding author: ibbousetta@gmail.com

Abstract

Developing with Model Driven Architecture is nowadays widely used starting with a CIM that can be transformed to models of low abstraction (PIM, PSM) that can be used to generate the code. The CIM represents the highest level of abstraction of the approach which allowing modeling system's requirement. However, there is no standard method to build this type of model or how to transform it to lower level of abstraction (PIM) which is considered the final objective of building such model.

This paper provides an approach to build the CIM that can be transformed (semi-) automatically later to lower levels of abstraction in PIMs. Thereby, the proposed architecture represents both the static and dynamic view of the system based on the business process model. Meanwhile, the PIM level is represented by the Domain Diagram class and Sequence Diagram of Systems External behavior. Thus, the proposal helps bridging the gap between those that are experts about the domain and its requirements, and those that are experts of the system design and development.

Keywords: CIM to PIM transformation; MDA; software process;

1. Introduction

The model driven architecture (MDA) [1] is an approach for software development that was initiated by the Object Management Group (OMG) in 2001. MDA proposes a Y cycle development and promotes the use of models different levels of abstraction [1]:

- Computation Independent Model (CIM): A CIM does not show details of the structure of systems. A CIM is sometimes called a domain model and a vocabulary that is familiar to the practitioners of the domain in question.
- Platform Independent Model (PIM): A PIM exhibits a specified degree of platform independence so as to be suitable for use with a number of different platforms of similar type.
- The Platform-Specific Model (PSM) combines the platform independent model with an additional focus on the detail of the use of a specific platform by a system.
- The code that represent the final implementation of the solution in the target platform.

Another important issue in MDA approaches is transformation among those models. The concept of OMG's MDA process to transform higher levels (CIM, PIM) into lower levels (PIM, PSM) that are used to create implementation code. Transformation of a model is a process when one model is a source, converted into another model – destination with the use of certain transformation rules.

Since till now, almost designs approaches are based on the PIM level of MDA, the biggest emphasize is put on PIM to PSM transformation in both ways. There are various CASE tools presented as MDA tools that automate this transformation in a great manner. CIM to PIM transformation is not mentioned often by OMG and as this can be of a great help for business analysts and domain experts we have decided to focus on this higher level of MDA. CIM is the level that does not display details of the information system but it specifies activities that are being processed in this system. In other words this level represents business processes of the organization for which the system will be developed.

In this paper we present a methodology for modeling the CIM level based on the Business process model (BPM) using the Business Process Model Notation (BPMN). We propose also a model transformation to map this CIM to a PIM with a low level of abstraction. The approach proposes architecture for both the CIM and PIM levels. The CIM is represented by the BPM and use case model allowing thus to represent the functional, behavioral and static views

of the system. This architecture allows an easy semi-automatic transformation to PIM models. This last one is represented by two models: the Sequence diagram of system's external behavior (SDSEB) and the Domain Class Diagram (DCD) for the behavioral and static views respectively.

The reminder of the paper is structured as follow: The next section gives an overview on the CIM level and its transformation to PIM and introduces the relevant related works to this topic. Section 3, presents the proposed architecture for the CIM level and how it can be modeled using the BPM expressed with the BPMN notation. After that, section 4 introduces the architecture of the PIM level and the Business rules that is used to complete the Domain class diagram. Then section 5 presents how each element of the CIM level will be mapped to the PIM level. An example is given to illustrate the proposal. Before to conclude this work section 6 presents an evaluation of the proposal based on a case study of e-commerce web site and criterion evaluation.

2. Background and related works

The OMG describes different levels and their relations but it does not specify how to create these abstract levels and which exact models and notations to use for their representation and how to transform them from one to another. Transformation of CIM to PIM is not considered as a simple mapping from one model to another through a model transformation language rules. Indeed, the main reason of the fails of IT projects bellows to requirement understanding and specification. Therefore it is necessary to remind the reader that before the CIM to PIM transformation we have to actually transform the organization processes of the enterprise based on the business analyses or the domain expert of the institution into the CIM model. According to MDA principles, the main characteristic of such CIM is its capabilities to be transformed later to a PIM. However, his capability is discriminated by the chosen architecture of CIM.

Furthermore, many software architects understand that CIM level and its following transformation to PIM is the first step to quality design of complex information system. Nevertheless, this importance is underestimated and almost of the research concentrates on PIM to PSM transformation and PSM to code generation. When solving the CIM to PIM transformation it is necessary to understand that activities and processes on CIM level represent business reality and further levels are necessary for the system's development. CIM analytic model describes all the activities, manual and half automatic therefore it is necessary to do reengineering or redesign of existing processes.

Creation of CIM level is not unified now and does not use unified standard but it is assumed that this level is represented by the model of business processes [2], [3]. According to [2] the transformation CIM to PIM is presented like disciplined approach. It uses UML2 activity diagrams which model the business processes. It is modelled like the user's tasks. From detailed activity diagrams, system requirements are specified. From the model of requirement elements the system components are created. Finally, a set of business archetypes helps to transform the system components to the PIM layer in details. In [3] there is presented an approach in which CIM level is represented by business processes in BPMN notation. Various UML Use Cases which present some part of information system are obtained from business processes using Query/View/Transformation (QVT) [5] rules. In [4], an approach is represented where the features and components are adopted as the key elements of CIM and PIM building and responsibilities as the connectors between features and components to facilitate CIM to PIM transformation.

In [6] presented a possible solution for CIM modelling and then transform it to PIM using the analytic method of transformation. In that paper CIM level represented by Data Flow Diagram (DFD) that it is used for business process modelling.

While, in [2] proposed a disciplined approach for transformation of CIM into PIM. In this paper, CIM includes Business Process Model and requirement model. First Business Process modelled using an activity diagram then activity diagram details for specifying a system requirement. In [7] presented an approach for transforming CIM into PIM where the CIM level is represented by a secure business process in BPMN [8].

In this paper we present an approach for CIM to PIM transformation where the CIM level is represented by the BPM and use case model whereas the PIM level is represented using the SDSEB and DCD. The main advantage of the proposed architecture of both CIM and PIM levels is that it allows representing a complete view of the system including the static, functional and behavioural. Thus, the transformation and the understanding of the problem are assumed easily.

The proposal completes our previous works [33, 34, 35, 36] that subscribe in global approach that aims automating the whole development process. Indeed, the generated SDSEB and DCD are used later in [34] to generate the sequence diagram of system's internal behaviour that can be considered as design PIM. This SDSIB is used to generate later in [35] a PSM for the java platform and generate the executable code.

3. CIM modelling:

CIM does not have any information about models or artifacts that are used for the implementation of the system. It describes the environment in which the system operates and aid to recognize what it is been expected of this system. It is useful for the analysts on the top level (business analysts, domain experts or domain users of the system) to understand the problems that must be implemented. Thereby, CIM plays an important role in passing the gap among specialists for domain (business and domain experts) and specialists for design and development of the system (software analysts). In MDA specification the requirements on CIM should have relations to PIM and PSM construction and vice versa.

3.1 *CIM architecture*

A CIM shows the environment of the software and its requirements in a way that can be understood by domain experts. The CIM is often referred to as the domain model and is specified using the vocabulary of the domain's practitioners and the stakeholders [1].

The first question before thinking to build the CIM is what must be represented in this level? In [9] is mentioned that Regarding CIM, there are two topics: First one is business model [10] and the second is the system requirements [11]. Some researchers position both models representing business knowledge and system requirements at the CIM level [12]. Moreover, the main objective of building the CIM is to be transformed later to a lower level of abstraction in a PIM. Therefore, this CIM must represent both the behavior and static aspect of the system. The behavior aspect can be represented by Business process model- BPM that can be represented using three modeling techniques: Data Flow Definition-DFD, UML (activity diagram), and Business process Modeling Notation-BPMN.

Giving that, requirements should be modeled in CIM and that many software development processes are use case driven approach, software requirements are represented using "use cases". However, a use case diagram presents what the system is intended to do, without providing more details about the business process that remains necessary to pass the gap between the domain experts and software developers. Thus a business process model would be useful for this aim. Indeed, Jacobson [13] defines a UML use case as: '...a sequence of actions, including variants that the system can perform, and that yield an observable result of value to a particular actor'. According to this definition, a use case consists of activities (or actions), which are ordered in some way, are sequential and are aimed at delivering a particular result. Meanwhile, a simple UML use case diagram represents only the functionalities that will be performed by the future system. Therefore, we propose for representing the CIM to use a business use-case model which is a model of the business's intended functions that consists of business actors and business use cases, and the business process model that will represents the detailed behavior of each use case as well as the static aspect of the system that is considered necessary to build a transformable CIM. The BPM will be built using the BPMN that allows representing both dynamic aspect through processes, sub-processes, tasks... and the static aspect through data object used to perform each task or process. A BPM shows how the business use cases are "performed" in terms of interacting business workers and business entities.

Furthermore, According to Larman [14], this definition of the use cases deals with the definition of an EBP: 'a use case should specify the actor-system interactions of an EBP', that can be represented in the BPM using the sub-process element in the BPMN notation. Also, Larman [14] and Jacobson [15] have recommended writing use cases with activity diagrams. This technique has also been successfully experimented for several years by different practitioners [16].

Hence, the proposed architecture for CIM level will be represented by two models: the use case model that represents the functional aspect of the system that will represent the business actors and business functionalities that are intended to be realized; the second model is the Business process model that will represents both the behavior and static aspect that will represents the different activities necessary to model businesses and resource used by those activities enabling thus the capability of transforming this CIM to a low level of abstraction in the PIM level. Figure 1 below illustrates the proposed architecture for the CIM.

3.2 Business Process Modelling

A Business Process can be defined as a set of one or more linked procedures or activities which collectively realize a business objective or policy goal, normally within the context of an organizational structure defining functional roles and relationships.

Process modeling is widely used within organizations as a method to increase awareness and knowledge of business processes, and to deconstruct organizational complexity [17]. It is an approach for describing how businesses conduct their operations and typically includes graphical depictions of at least the activities, events/states, and control flow logic that constitute a business process [18], [19]. Additionally, process models may also include information regarding the involved data, organizational/IT resources, and potentially other artifacts such as external stakeholders and performance metrics, to name just a few [20].

Hence the modeling of business processes is becoming increasingly popular. Both experts in the field of Information Technology and Business Engineering have concluded that successful systems start with an understanding of the business processes of an organization. Furthermore, business processes are a key factor when integrating an enterprise [21]. Indeed, according to a survey of IT executives conducted by the Standish Group [25], only 29% of software projects succeeded, while 53% were challenged and 18% completely failed. As pointed out by these IT executives, the primary reason for software projects being challenged or failing is poor conceptual modeling (requirements' definition).

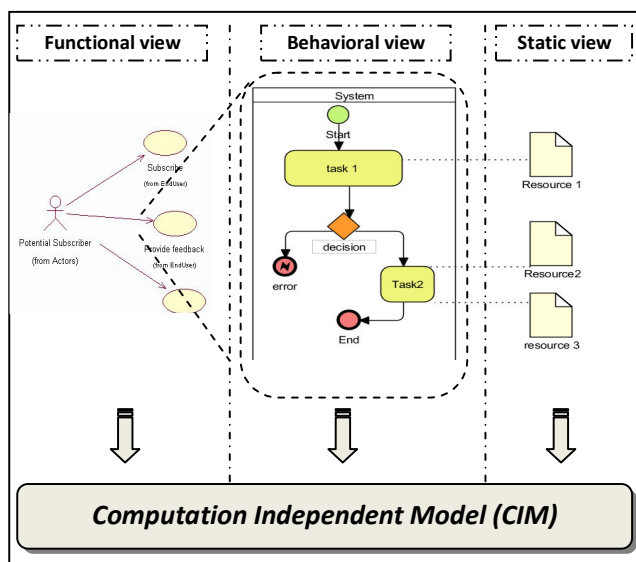


Figure 1: Proposed architecture for the CIM level.

BPM is the representation of a business process in a form that supports automated manipulation, such as modeling, or enactment by a workflow management system. The process definition consists of a network of activities and their relationships, criteria to indicate the start and termination of the process, and information about the individual activities, such as participants, associated IT applications and data, etc. Generally, a business process model includes concepts that combine three following basic descriptive views [22]:

- **Functional View:** The functional view is focused on activities as well as on entities that flow into and out of these activities. This view is often expressed by Data Flow Diagrams [23].
- **Behavioural View:** The behavioural view is focused on when and/or under what conditions activities are performed. This aspect of the process model is often based on various kinds of State Diagrams or Interaction Diagrams. More sophisticated approaches based on the theory of Petri Nets are convenient for systems that may exhibit asynchronous and concurrent activities [24]. The behavioral view captures the control aspect of the process model. It means that the direction of the process is defined on current state of the system and event that occurs.

- Structural View: The structural view is focused on the static aspect of the process. It captures objects that are manipulated and used by a process as well as the relationships that exist among them. These models are often based on the Entity-Relation Diagrams or any of the Object Diagrams that are used by the various kinds of Object Oriented Methods.

A BPM describes the activities (the dynamic aspect) and the objects (the actors, the resources required), the products or services resulting from the activity required for their realization (the static aspect). In this proposal we propose represent those three views in one diagram using the BPMN notation.

The modeling of business processes often starts with capturing high-level activities and then drilling down to lower levels of detail within separate diagrams. There may be multiple levels of diagrams, depending on the methodology used for model development. However, BPMN is independent of any specific process modeling methodology. In this high level process, we find basically a series of Sub-Processes with nodes decisions. The sub-process can be defined as an Elementary Business Processes (EBPs) that represent the down level that we are not interested to decompose any further. An EBP is defined as “A task performed by one person in one place at one time, in response to a business event, which adds measurable business value and leaves the data in a consistent state” [26]. It corresponds to a well-defined and well-delimited user’s task, based on the chronology of events and activities. It also identifies the business entities required by the task. Business entities encompass both resources used and objects created by the activity. Thus, the process is decomposed into a collection of EBPs which are related according to a workflow specifying the handing over of the tasks from one actor to another.

The BPM can also used to represent the lowest level abstraction of sub-process. In this diagram we find the tasks that are organized also in work flow with nodes decisions inducing objects/resource used or created by the task. While, the Figure 2 below shows an example of high level process for passing an online exam, the Figure 3 show expand sub-process that represent the lowest level process of the sub-process “Pass exam”. Figure 4 shows the complete business process model with both high and lowest level of process for the passing online exam.

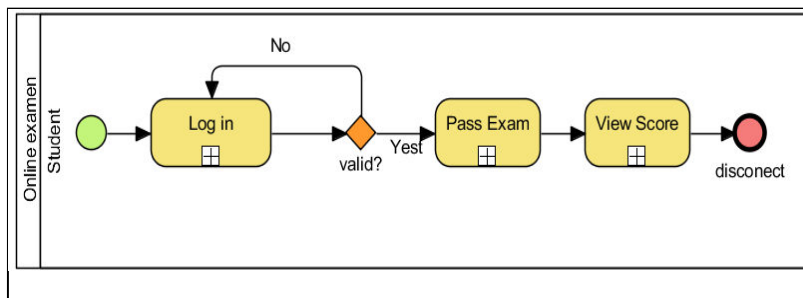


Figure 2: A high level process for an online exam case study.

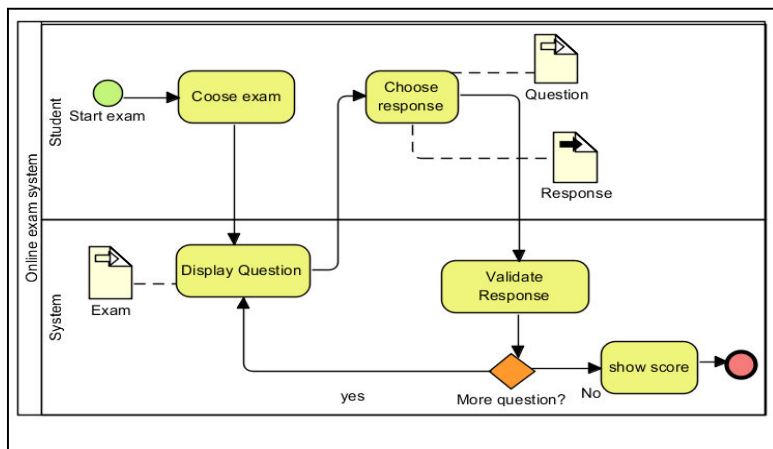


Figure 3: A lower level process for the Pass exam sub-process in the passing online exam case study.

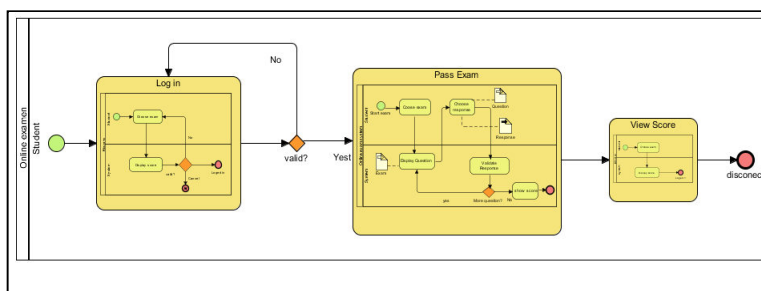


Figure 4: Detailed Business process model for passing online exam case study.

3.3 BPMN

The membership of the Business Process Management Initiative (BPMI) Notation Working Group represents a large segment of the business process modeling community, and they have come to a consensus and present The Business Process Modeling Notation (BPMN) as the standard business process modeling notation.

The BPMN is the new standard to model business process flows and web services that provides a notation that is readily understandable by all business users including the business analysts that create the initial drafts of the processes to the technical developers responsible for implementing the technology that will perform those processes.

BPMN specifies a single business process diagram, called the Business Process Diagram (BPD) that it is easy to use and understand. You can use it to quickly and easily model business processes, and it is easily understandable by non-technical users (usually management). It also, offers the expressiveness to model very complex business processes, and can be naturally mapped to business execution languages.

To model a business process flow, you simply model the events that occur to start a process, the processes that get performed, and the end results of the process flow. Business decisions and branching of flows is modeled using gateways. A gateway is similar to a decision symbol in a flowchart. Furthermore, a process in the flow can contain sub-processes, which can be in an expanded form that shows the process details of a lower-level set of activities. If a process is not decomposed by sub-processes, it is considered a task – the lowest-level process. A ‘+’ mark in the process symbol denotes that the process is decomposed (Figure 2); if it doesn’t have a ‘+’ mark, it is a task (Figure 3).

4. PIM modelling

PIM level shows certain level of independence in such a way that models that are represented there are suitably chosen for use in various platforms. PIM describes information system, but hides details in usage of concrete technology. PIM creates specification for required services of the information system without technical platform dependent details.

4.1 PIM architecture

A complete PIM should describe two aspects of a system: Structure and Behavior. The structure (or static) aspect emphasizes the static structure of the system using classes, objects, attributes, operations, relationships, etc., while the behavior (or dynamic) aspect emphasizes the dynamic behavior of the system by showing interactions among objects, etc.

In the proposal to represent the PIM level we propose to use the domain class diagram that shows the static aspect of the system and the sequence diagram of system's external behavior-SDSEB that is a UML sequence diagram that shows only interactions between actors and the whole system as unique entity which is represented by one lifeline without focusing on system objects interactions, it represents a high level of interaction. The domain class diagram is obtained based on the business rules. The SDSEB is transformed later into other PIM with low level of abstraction that will represent detailed interactions between system objects called the Sequence diagram of system's internal behavior (SDSIB). This last transformation is presented in our previous work [34]. The Figure 5 illustrates the proposed architecture for the PIM level.

4.2 Business rules

Business rules have been defined as 'declarations of policy or conditions that must be satisfied' [27], and their role is to determine how operational decisions within an organization must be made. In other words, business rules specify how businesses is conducted in an organization and what are the constraints to be respected in the different activities. Business rules are not "process" in any sense of the word. Roger Burlton recently expressed the business rule message this way: "Separate the flow from the know". Business rules represent the "know" part of that the stuff that guides the "flow." Guidance means following rules, of course, hence the name "business rules".

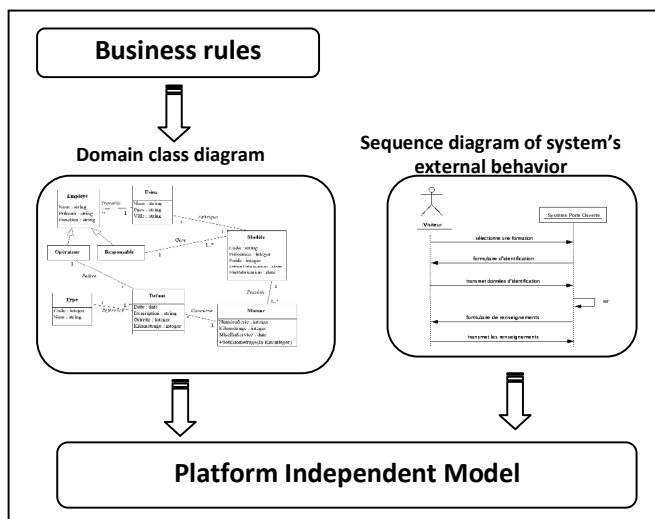


Figure 5: Proposed architecture for the PIM level.

The term 'business rule' has been used by different methodologists in different ways. In [28], business rules are 'statements of goals, policies, or constraints on an enterprise's way of doing business'. In [29], they are defined as 'statements about how the business is done, i.e. about guidelines and restrictions with respect to states and processes in an organization'. Mitchell Krammer [30] considers them as programmatic implementations of the policies and practices of a business organization. Whilst Halle states that 'Depending on whom you ask, business rules may

encompass some or all relationship verbs, mathematical calculations, inference rules, step-by-step instructions, database constraints, business goals and policies, and business definitions' [31].

The Business Rules Group (BRG) classifies business rules into three main types: structural assertions, action assertions, and derivations [32].

- Structural assertion is a statement about concept or relationship of something of importance to the business. There are two kinds of structural assertions i.e. terms and facts. A term is a word or phrase, which has a specific meaning to business. A fact asserts an association between two or more terms.
- Action assertion is concerned with the dynamic aspect of the business. It includes a conditional action, integrity constraints, and optional actions.
- Derivation is a derived fact that is created by an inference or a mathematical calculation from terms, facts, other derivations, or action assertions.

In this proposal we are interested to build the static aspect of the system by creating the Domain class diagram, therefore we will focus only on the Structural assertion type of the business rules that defines the structure, relationship and the integrity constraints on the data. This type of BR is based on two important concepts: Term and Fact. A term is a word, phrase, or sentence(s) which has a specific meaning for the business. Terms may be written as '<term> is defined as...'. Facts are used for asserting an association between two or more terms 'fact relating term'. Facts connect things in the business. How terms like 'vendor' and 'supplier' related to each other in the phrase "vendor supplies material" and "each supplier must have an address" are business rules. Facts cause a business term to take on roles in the business (the vendor takes on the role of supplier). Expressions like 'x <connecting verb> y'; 'k contains z', 'y is a type of l' are expressing facts relating terms.

We propose to use a template based on Business rules of business domain to express the business rules that can be transformed later even semi-automatically to deduce the structure of the system and generate principally the domain class diagram. The template use a subset of a natural language (both syntactically and vocabulary-wise) to minimize ambiguities.

The proposed template to express these business rules is illustrated in the Table 1. We have used a template in the form of structured English based on the concept of fact and term.

Table 1: The proposed template to express businesses rules.

Template	Example of Business rules
Term	Exam, Student, Response
<i>Fact</i>	Pass, own, use
<Term> <Fact><term>	The student <i>passes</i> an exam The student <i>answer</i> questions
<Term> <i>is characterized by its</i> <Term>, <Term>,...	An exam is characterized by a date of exam, a duration and a set of questions
<Term> <i>belongs to one/many</i> <Term>	An exam <i>belong</i> to one category
<Term> <fact> many/a/an/number <term>	A question <i>has</i> four Reponses
<term> <i>may/can</i> be a <term1> <i>or</i> <term2>	An exam <i>can</i> be Multiple Choice Question <i>or</i> a direct questions
<term> has number/ is <i>types:</i> <term1>, <term2>, <term3>...	An exam has two <i>types</i> : Multiple Choice Question , direct questions

5. CIM to PIM transformation

In the transformation from a CIM to a Platform Independent Model (PIM) the purpose of the models change and the focus is on the computational complexity that is needed to describe the behavior and structure of the software. The PIM is then transformed into a Platform Specific Model (PSM) which is a concrete solution to the problem as specified by the CIM. The PSM will include information about which programming language(s) to use and what hardware to deploy the executable code on.

The Figure 6 below illustrates the proposed approach with the architecture for both the CIM and PIM levels as well as the performed CIM to PIM transformation.

5.1 High level BPM to Use case model

In this level we are interested to express the functional view of the system that can be modeled through the use case model. This can be easily done by focusing on collapsed sub-process in the high level BPM. However, before to start mapping those sub-processes we can distinguish, from the point of view how the activities are executed, three kinds of activities:

- Manual – pure human based. No computer resources are needed. The human resource executes the activity based on directions associated with it.

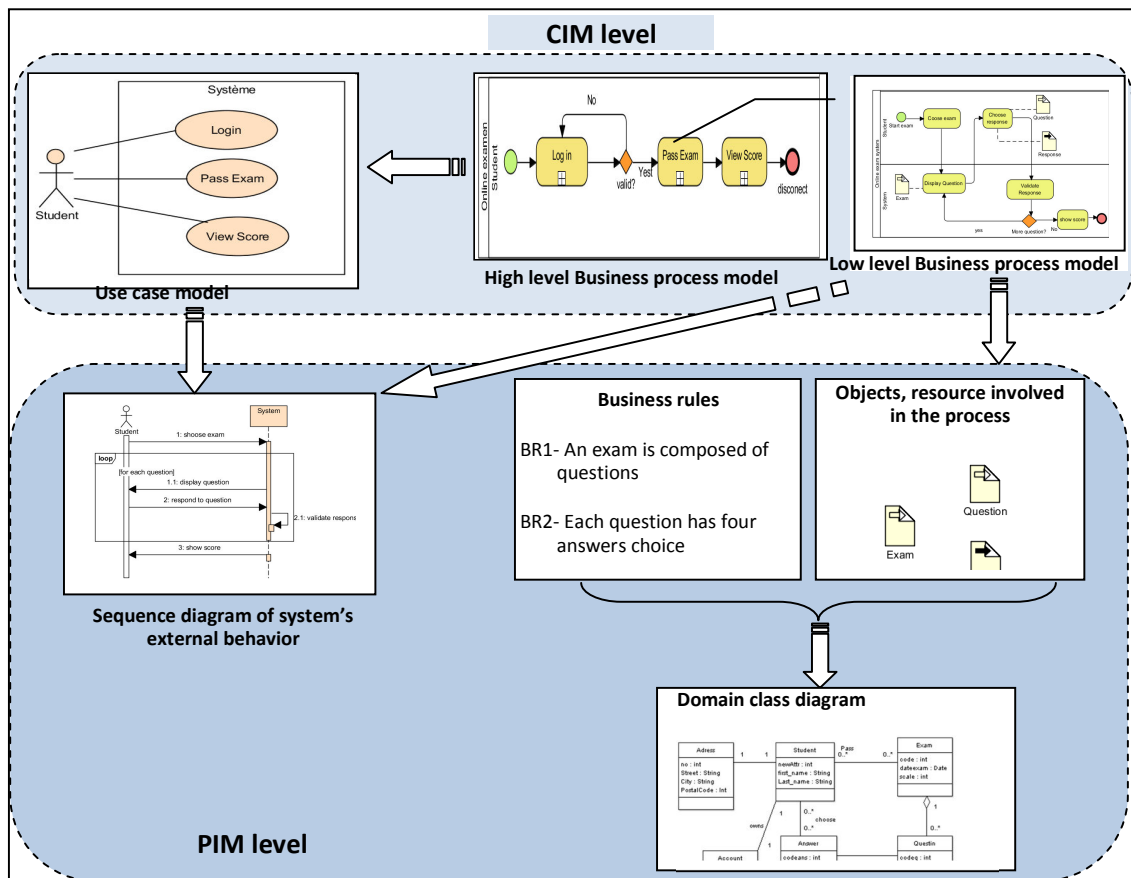


Figure 6: The proposed approach for the CIM building and its transformation to PIM

- Semi-Automatic – human activity is supported by an external application. The activity has associated not just directions but also the code that is executed by the actor. This code migrates to the actor's computer where it is executed using instances associated with the activity. For example, the activity Car Selection uses the application that searches in a database of car manufacturer. The list of available models is displayed to the Salesman and Customer.
- Automatic – pure computer based. No human resources are needed. The activity has code associated. The code uses instances and it is executed immediately. The code is executed on the same computer as the workflow engine is running.

We can see that only automatic and semi-automatic activities will be programmed, the manual one is totally performed by human actor and cannot be programmed.

The table 2 below presents the detailed mapping rule for transforming the HLBPM elements to UCM elements.

According to Larman [14], the definition of use case deals with the definition of sub-process or EBP. Thereby, each collapsed sub-process is transformed to a use case in the use case model except the manual process that will not be developed. The associate actor is represented by the swimlane within the pool where the sub-process is executed. This actor is the one who triggered the activity and it will be considered as the principal one. Secondary actors can be deduced from the expanded sub-process model corresponding to the collapsed sub-process. All the represented swimlanes in this diagram will be considered as secondary actors.

If there is a based data gateway to control the flow in a process between two activities this will be mapped with “extend” relation between those use cases corresponding to those activities. The condition of the extension point will be defined by the guard condition of the gateway.

Applying those mapping rules on the passing on line exam case study presented in Figures 2, 3 and 4 gives as results the use case model presented in Figure 7. The HLBPM of this case study presents three sub-processes: Login, Pass exam and view score that are mapped to uses cases in the UCM. The principal actor is the Student that corresponds to the unique swimlane present in the same diagram. The LLBPM presented in Figure 2 does not involve any supplement actor; thereby there will be no secondary actor for those uses cases.

Table 2: Mapping rules for HLBPM to UCM

Element in HLBPM	Corresponding element in UCM
Sub-process	Use case
Swimlane in HLBP	Principal actor
Swimlane in LLBP	Secondary actor
Sub-process within swimlane	Association between use case and actor
A based gateway between two activities sb1 and sb2	Extends association between the corresponding use case: sb2 extends sb1
Two activities sb1 and sb2 in same flow and there is only this unique flow to this activity sb2	Include association between corresponding use cases : sb2 includes sb1

5.2 Lower level BPM to SDSEB

The sequence diagram of system’s external behavior shows the dynamic view of the system even in high level of abstraction. This one can be obtained from the lower level BPM that capture the same dynamic of the system and that is built using the BPMN notation.

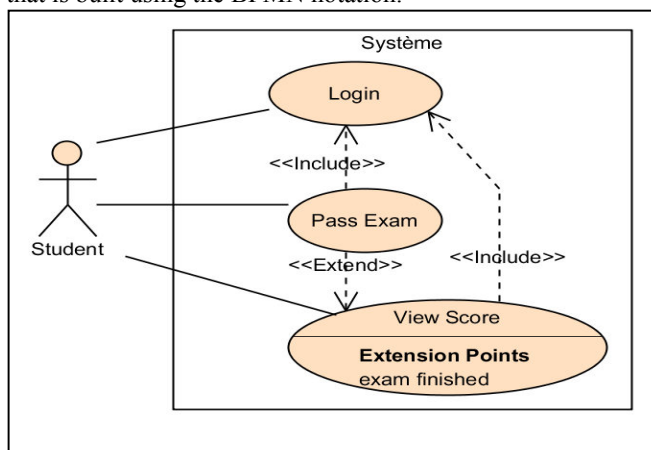


Figure 7: The use case model corresponding to the passing on line exam.

From the use case model we can get the principal actor and the secondary actors involved in the use case. LLBPM will allow getting all the actions and messages sent from those actors and the corresponding system's response. Each swimlane is mapped to an actor. While, each task performed by the actor is transformed to an action or message sent to the system (choose exam, respond a question).

Each task performed by the system and that performs a calculation or validation activity will be transformed to an internal message in the system (validate response). Meanwhile, each task performed by the system that performs a display activity or asking for information from the actor will be mapped to a response sent from the system to the actor (show score, display question).

If there is an exclusive gateway based on data, the successful flows will be represented in the SDSEB and the other flows will be considered as an alternative scenario. Otherwise, if the secondary flow is terminated with error or cancelation message it will be considered as error scenario. Both alternative and error scenario are represented in the SDSEB with notes. If the flow returns to a previous task then the messages and their response corresponding to those tasks will be enclosed in LOOP interaction operand (display question, and respond to question). Collapsed sub-processes are mapped with REF interaction operand to the use case corresponding to this sub-process. Figure 8 shows the resulting SDSEB by applying the proposed mapping rules in Table 3.

Tbale 3: Mapping rules for LLBPM to SDSEB

Element in LLBPM	Corresponding element in SDSEB
Swimlane in HLBP	Principal actor
Swimlane in LLBP	Secondary actor
Task performed by actor	Message sent to system
Task performed by the system for calculation or validation	Internal message
Task performed by the system for getting/displaying information from/for actor	Response from the system to the actor
Flow of activities after an exclusive gateway	Successful flow is mapped to actions/responses. While, other flows that terminate process correctly are mapped to alternative scenario, floes that terminate process with errors or cancelation are mapped to error scenario. Both errors and alternative are represented by notes.
Flow that returns to previous task	A Loop interaction operand
Sub-process	A REF interaction operand

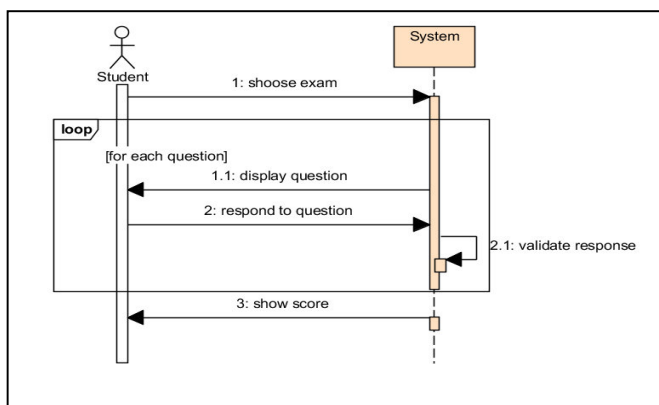


Figure 8: The resulting SDSEB for the case study.

5.3 Business rules to DCD

We have already the objects and resources used in different task in the LLBPM of different Sub-processes of the case study (see Figure 9). Those Input/output data objects are considered as terms and are mapped to classes in the DCD. Those objects can be completed with the different terms and fact deduced from the business rules according to the

mapping rules presented in Table 4 below.

Table 4: Mapping rules for the business rules.

Expression	Signification
Nouns, roles, concepts	are considered as terms =>Class
This, these, that, those, ... and synonyms	same term
Its, his, her, their...	express a relation between two concepts . the term is an attribute of the owner term if it is a simple property (atomic) , or it is a class if it is not simple.
List, set of	: an ordered constrains in OCL
The verbs belongs, composed, contains, include...	is considered as a fact that means an association of composition or aggregation.
Many, a, an, any, several, a lot of, one, numbers, plural ...	multiplicity in an association.
Is .. Or.. , may/can be .. or ...,	express a generalization/specialization relationship

While, the Terms become entities/objects (or tables on a database) or attributes/properties (database columns), facts are represented as associations, subtypes (subclass), roles, aggregates, and attributes. Table 4 below presents how DCD's elements are deduced from the BR.

Examples of some businesses rules for the passing on line exam case study:

- BR1- A student passes many exams
- BR2-An exam is characterized by a code, date of exam, and scale
- BR3- An exam is composed of questions
- BR4-Each question has four answers choice
- BR5-Each answer choice has a status of validity
- BR6-The student choose a response for each question
- BR7-A student is characterized by its first and last name, and address.
- BR8-A student has an account
- BR9-The account is characterized by a login and password

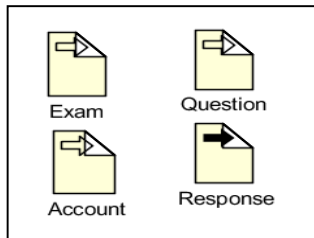


Figure 9: The input/output data object involved in the SDSEB for the case study.

The BR1 allows determining the classes Student and Exam that associated with pass association and the BR2 determine attributes of the Exam class (code, date, scale). Meanwhile, BR3 specifies that an aggregation association between exam and Question, another class of domain.

BR 4, 5 and 6 specify that a question is associated to four responses and response has an attribute validity that determines either the answer is correct or not. The Student is associated with question with an answer association that associates the student, question and the chosen answer.

While BR 7 determines some Student's attribute: first name, last name. Address is considered as Class because it is not a simple property (atomic), BR9 defines a new Class Account with two attributes: login and password; and BR8 associate it with Student class.

Figure 10 below presents the resulting DCD for the passing online exam case study

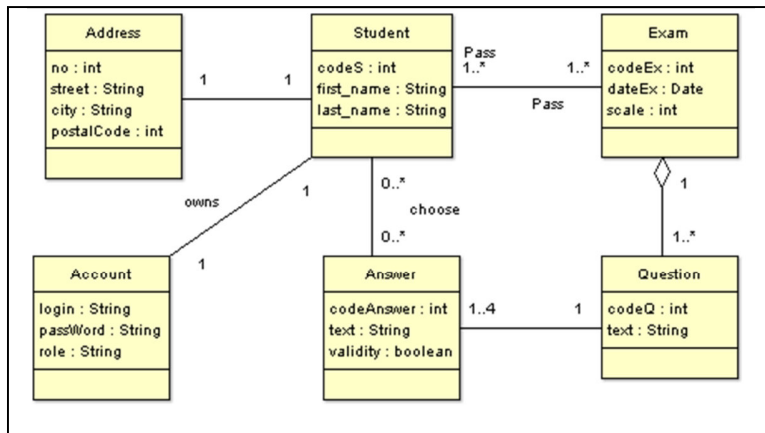


Figure 10: The resulting Domain class diagram for the case study.

6. Evaluation

The evaluation is performed by the case study that is introduced to illustrate the proposal. The second evaluation is a criterion based one.

6.1 Case study: e-Commerce web site

To evaluate the approach we propose to use a complex case study that knows an important business workflow therefore the choice of the e-commerce web site case study.

6.1.1 Case study:

Any surfer on web can access to the web site and search for product of different categories (Book, informatics....) and collect them in its cart. He can manage this cart at any time to add/remove products or to change the quantity of items. When he is convinced he can check out the order and pay the command that will be shipped to his address. Web Surfer must login with their account or subscribes if it is their first visit for the web site.

6.1.2 Business Process model

To implement the proposed approach for the chosen case study we start with high level sub-process business model represented in Figure 11 that is a set of collapsed sub-process organized in work flow and represented using the BPMN notation.

For an example we have chose two collapsed sub-processes check out and payment that were detailed with a Lower level business model that represents a expanded sub-process with a set of task organized in workflow controlled by a set of gateways. Figures 12 and 13 show those two diagrams.

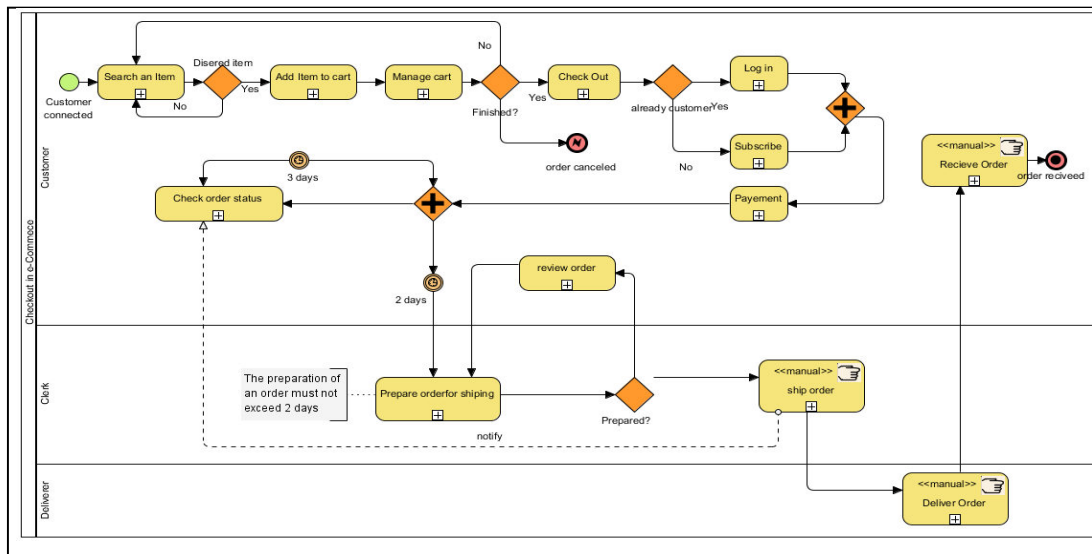


Figure 11 : Collapsed sub-process for the e-commerce case study.

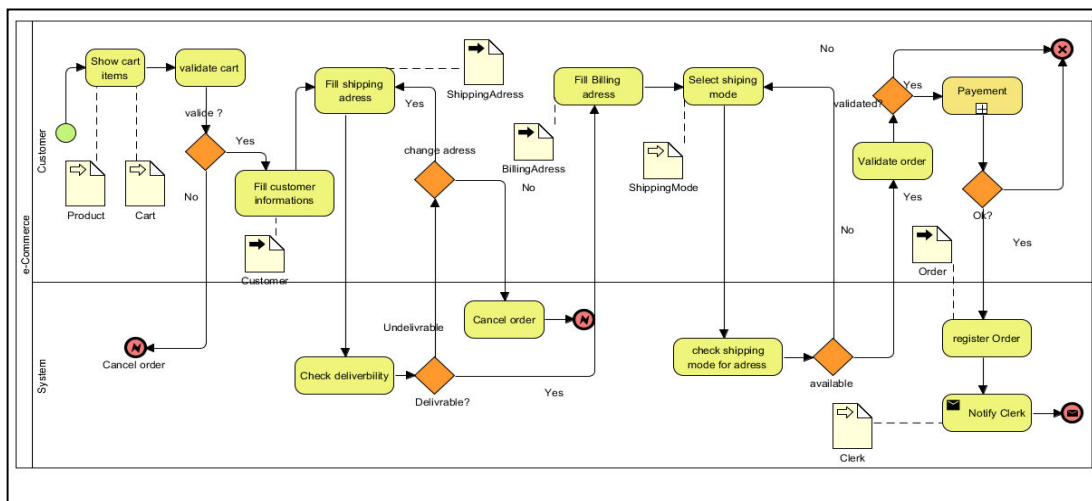


Figure 12 : Expanded sub-process checkout.

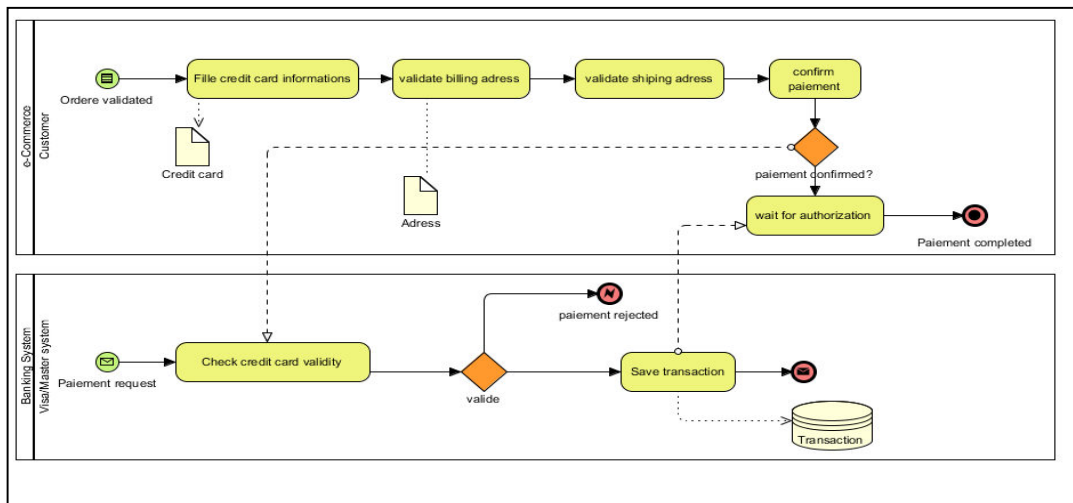


Figure 13 : Expanded payment sub-process.

Applying the mapping rules proposed to generate the UCM, we can identify the folwing use cases : *subscribe*, *search an item*, *add item to cart*, *manage cart*, *login*, *checkout*, *check order status*, *review order*, *prepare order for shipping*. The sub-processes : “receive order”, “ship order” and “deliver order” are manual therefore thy will not be mapped to a use case. We can also identify two actors : *Customer* and *Clerk* and a secondary actor for the payment use case : the *bankin system*. Figure 14 illustrates the generate use case model.

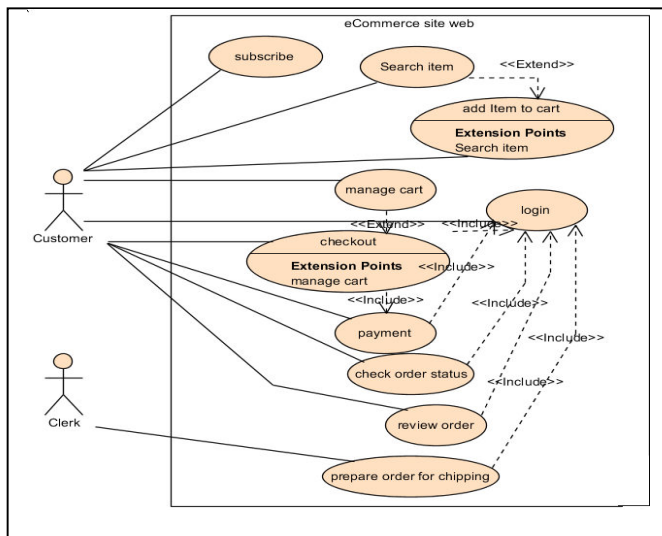


Figure 14 : Use case model for the e-commerce case study.

6.1.3 Resource and data object

From the lower level business process model we can deduce the involved resourecs or data objects in each activity. The Figure 15 presents those entities.

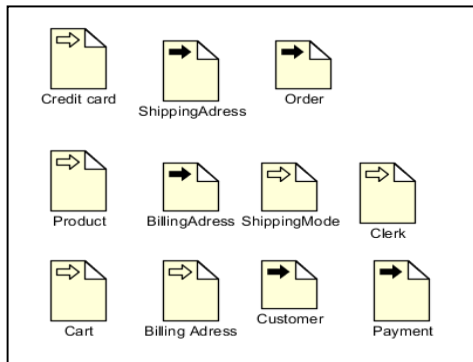


Figure 15: Resources used in activities

6.1.4 Generating the SDSEB corresponding to the checkout use case

From the use case model we can identify the principal actor that is the customer. The lower level BPM does not show any suppliers secondary actors. We can identify three internal messages: validate shipping address, check delivery mode and save order. The other tasks are transformed to actions/response from/to system.

The collapsed sub-process payment is mapped to a REF interaction operand for the use case Payment. The Figure

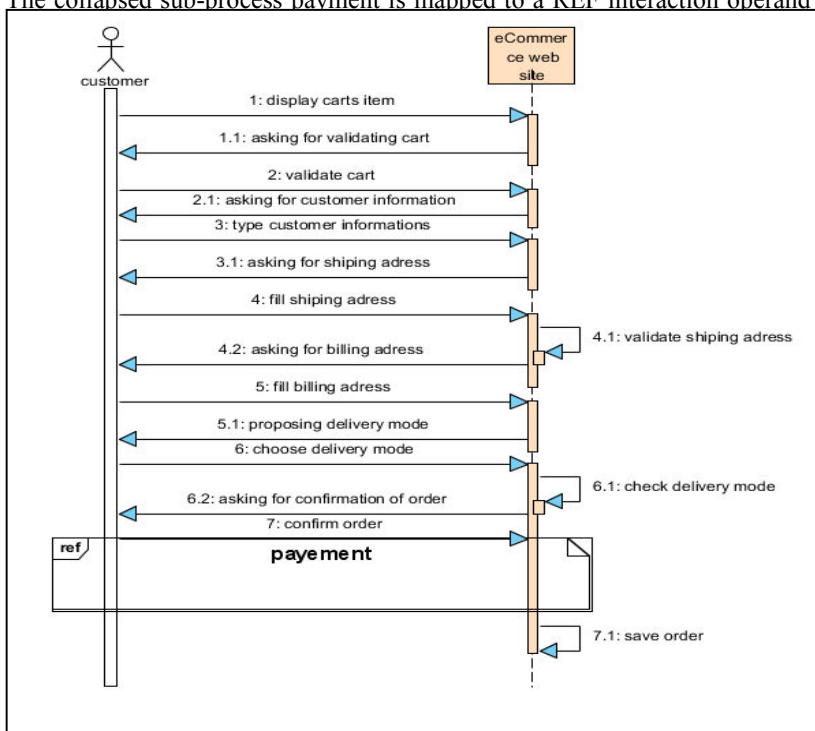


Figure 16 : Generated SDESB for the check out use case.

6.1.5 Business rules to Domain class diagram

We will present here the important Businesses rules concerning the case study.

- BR1- A customer passes many order
- BR2- An Order concerns at least one product
- BR3- An Order has a billing address and a shipping address.
- BR4- Product belongs to one category
- BR5- Product is characterized by a reference, description and a price
- BR6- A Customer is characterized by a code, first name, last name, an email address
- BR7- An order has a Status
- BR8- an Order is characterized by a date, and reference
- BR9- For each item in the cart we specify the quantity
- BR10- A Customer has an account
- BR11- An account is characterized by a login, password and role.
- BR12- An Order has a payment
- BR13- A payment indicate a Credit Card and a amount
- BR14- A Credit card is characterized by a Number, validity date.
- BR15- An Order has a Shipping mode
- BR16- A Customer can review order
- BR17- A Customer can cancel the order

In this step we will uses those businesses rules to complete the entities presented in Figure 15.

From BR 1, 2, 3, 4, 5 and 6 we can identify the classes Customer, Order, Address, Product and Category. The class Customer has three properties: Code, first name, last name and email address; and it is associated to the class Order with the passes association. The order class is associated with address with two roles: shipping address and billing address. The Order is associated also to the Product, which has the attributes: reference, description and price and associated to the category class, with the concerns association giving place to the OrdLine association class with attribute quantity (BR9). We can also deduce the flowing attributes: reference, date (BR8) and a status (BR7).

The customer class is associated to the class Account that owns the attributes: login, password and role (BR 10 and 11).

From BR 12, 13, 14 and 15 we can identify others classes: Payment, Credit card and shipment mode. The payment uses a credit card that has the attributes: number and validity date and it is associated to one order specifying the amount pied for the order. The BR 16 and 17 identify two supplement associations between Order and Customer: review and Cancel.

The Figure 17 presents the complete resulting domain class diagram for the whole case study.

6.2 Criterion evaluation

We evaluate the CIM with respect to two evaluation criteria. The first one is "CIM creation" and the second one is "Coverage of CIM". The evaluation criterion "Coverage of CIM" is derived from the Taxonomy of CIM. Another criterion is type of transformation. Also we have tested if the approach proposes a manual, semi-automatic or automatic CIM to PIM transformation.

The table 5 below shows the evaluation of fours approaches and our proposal based on the criteria cited before. The comparison of those four approaches was extracted from [37]. We can see that regarding to CIM coverage only the approach proposed by the Kheraf covers both aspects Business model and use Case model. Meanwhile, the other approaches cover only one aspect of them.

Regarding the PIM coverage, we can see that two out of four approaches can derive only structural model elements (e.g., objects, classes, associations) from CIM. One approach can generate behavioral features of a system (e.g., sequence diagrams, state machines, and/or activity diagrams). One approach is capable of generating PIM including both structural and behavioral aspects of a system but still requires a total human intervention. We can see that our

approach covers both business model and requirement model in the CIM level and furthermore, that it can derive both structural and behavioral aspect from this CIM. Thus it is considered as complete approach according to the criteria mentioned above.

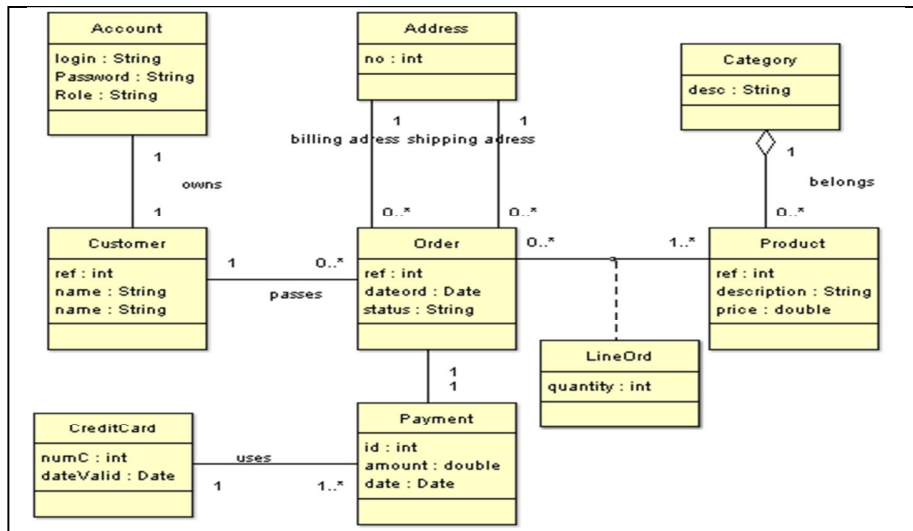


Figure 17: The resulting domain class diagram

Table 5: Criteria based evaluation

Approaches	CIM architecture	CIM coverage		PIM coverage		Automation
		Business model	Requirement model	Structural	Behavioral	
Wei et al. [4]	Feature Model		Yes	Yes		Semi-automatic
Kardoš et al. [6]	DFD	Yes		Yes	Yes	Manual
Kherraf et al. [2]	Activity Diagram Use Case Diagram	Yes	Yes	Yes		Manual
Rodríguez et al. [7]	BPMN	Yes			Yes	Semi-automatic
BOUSETTA et al.	BPMN Use Case Model Business rules	Yes	Yes	Yes	Yes	Semi-automatic

7. Conclusion

The paper proposes an approach for CIM level modeling and its transformation to a PIM within a model driven architecture approach. We have proposed an architecture for CIM modeling that covers functional and behavioral view of the system using business process model based on BPMN and requirement model represented by a use case model. The CIM is built so it can be transformed later to a PIM that is represented by Domain class diagram that represents the static view of a system and SDSEB that represents the behavioral one. The DCD is built using businesses rules that were expressed using structured English template. The proposal was evaluated using a case study that concerns an e-commerce web site and criteria based evaluation.

The proposal completes our previous works [33, 34, 35, 36] that subscribe in global approach that aims automating the whole development process. Other works to develop a tool that support all the transformations performed in our earlier works is undergo.

References

- [1] J. Miller and J. Mukerji. MDA Guide Version 1.0.1. Technical report, Object Management Group (OMG),2003.
- [2] Kherraf, S; Lefebvre, E; Suryan, W. Transformation from CIM to PIM Using Patterns and Archetypes, 19th Australian Conference on Software Engineering, 2008,<http://www2.computer.org/portal/web/csdl/doi/10.1109/ASWEC.2008.63>, downloaded: February, 27th 2010.
- [3] Rodríguez, A; Fernández-Medina, E; Piattini, M. Towards CIM to PIM Transformation: From Secure Business Processes Defined in BPMN to Use-Cases. In: Lecture Notes in Computer Science, Publisher: Springer Berlin / Heidelberg, ISBN 978-3-540-75182-3, pages 408-415, 2007.
- [4] Zhang, W; Mei, H; Zhao, H; Yang, J. Transformation from CIM to PIM: A Feature-Oriented Component-Based Approach. Work presented at MoDELS 2005, Montego Bay, Jamaica, 2005.
- [5] QVT: Meta Object Facility (MOF) 2.0 Query/View/Transformation Specification, 2008, <http://www.omg.org/spec/QVT/1.0/PDF/>, downloaded: February, 27th 2010.
- [6] Martin. Kardoš, Matilda. Drozdová , “Analytical method of CIM to PIM transformation in Model Driven Architecture (MDA),” JOURNAL OF INFORMATION AND ORGANIZATIONAL SCIENCES, vol. 34, pp. 89-99, 2010.
- [7] Alfonso. Rodríguez, Eduardo. Fernández-Medina, and Mario. Piattini, “Towards CIM to PIM transformation: from Secure Business Processes defined in BPMN to Use-Cases,” Business Process Management, vol. 4714, pp. 408-415, 2007.
- [8] “BPMN: Business Process Modeling Notation specification,” OMG, 2006.
- [9] Marite. Kirikova, Anita. Finke, and Janis. Grundspenki, “What is CIM: an information system perspective”, Advances in Databases and Information Systems, vol. 5968, 2010, pp. 169-176.
- [10] Steffen. Becker, “Coupled model transformations”, in Proceedings of the 7th international workshop on Software and performance. Princeton, New Jersey, USA: ACM, 2008, pp. 103-114.
- [11] Stefan. Biff, Richard. Mordinyi, and Alexander. Schatten, “A Model-Driven Architecture approach using explicit stakeholder quality requirement models for building dependable information systems”, in Proceedings of the 5th International Workshop on Software Quality. Washington, DC, USA: IEEE Computer Society, 2007, pp. 6-.
- [12] Janis. Osis, Erika. Asnina, and Andrejs. Grave, “Computation Independent Modeling within the MDA”, in IEEE International Conference on Software-Science, Technology & Engineering, Los Alamitos, 2007, pp. 22-34.
- [13] I. Jacobson, G. Booch, and J. Rumbaugh, The Unified Software Development Process. MA: Addison-Wesley, 1999.
- [14] C. Larman, Applying UML and Patterns, 3 ed: Prentice Hall, 2005.
- [15] I. Jacobson and Pan-Wei Ng, Aspect-Oriented Software Development with Use Cases: Addison-Wesley, 2005.

-
- [16] E. Lefebvre and E. Gagnon, "Evaluation of Visual vs. Textual Use Case Modeling," To be published.
- [17] Bandara, W., G. G. Gable, and M. Rosemann (2005) "Factors and Measures of Business Process Modelling: Model Building Through a Multiple Case Study", *European Journal of Information Systems* (14)4, pp. 347-360
- [18] Curtis, B., M. I. Kellner, and J. Over (1992) "Process Modeling", *Communications of the ACM* (35)9, pp. 75-90
- [19] Davenport, T. H. (2005) "The Coming Commoditization of Processes", *Harvard Business Review* (83)6, pp. 100-108
- [20] Scheer, A.-W. (2000) *ARIS - Business Process Modeling*, 3rd edition, Berlin, Germany: Springer
- [21] Aguilar-Savèn, R., Olhager, J., 2002. Integration of product, process and functional orientations: Principles and a case study. Preprints of the International Conference on Advanced Production Management System, APMS 2002, IFIP, September, The Netherlands.
- [22] Christie A. 1995. *Software Process Automation*. Springer-Verlag
- [23] DeMarco T. 1979. *Structured Analysis and System Specification*. Prentice-Hall, Englewood Cliffs, New Jersey
- [24] Peterson J.L. 1977. "Petri Nets." *ACM Computing Surveys*, vol.9, no.3 (Sept): 223-251
- [25] Standish, The Standish Group International, Inc. 2004 Third Quarter Research Report, West Yarmouth, MA, 2004.
- [26] C. Larman, *Applying UML and Patterns*, 3 ed: Prentice Hall 2004.
- [27] J. Martin, J. Odell, *Object-Oriented Methods: a Foundation—UML Edition*, Prentice Hall PTR, New Jersey, 1998.
- [28] D. Rosca, S. Greenspan, C. Wild, H. Reubenstein, K. Maly, M. Febowitz, Application of a Decision Support Mechanism to the Business Rules Lifecycle, 10th Knowledge-Based Software Engineering Conference (KBSE95), Boston, MA, 1995.
- [29] H. Herbst, *Business Rule Oriented Conceptual Modelling*, PhD Thesis, Physica-Verlag, 1996.
- [30] M.I. Krammer, Business rules: automating business policies and practices, *Distributed Computing Monitor* May (1997) 1997.
- [31] B.V. Halle, Back to business rule basics, *Database Programming and Design* (October 1994).
- [32] D. Hay, K.A. Healy, *Defining Business Rules What Are They Really?* Technical Report Rev 1.3, the Business Rules Group, July 2000
- [33] Omar El Beggar, Brahim Bousetta, Taoufiq Gadi, Generating methods signatures from transition state diagram: A model transformation approach. *Information Science and Technology (CIST)*, 2012 IEE Colloquium in Fez, 4-9
- [34] BOUSETTA Brahim, EL BEGGAR Omar, GADI Taoufiq. Automating software development process: Analysis-PIMs to Design-PIM model transformation. Under review.
- [35] EL BEGGAR Omar, BOUSETTA Brahim, GADI Taoufiq. Automatic code generation by model transformation from sequence diagram of system's internal behavior. *International Journal of Computer and Information Technology (IJCIT)* December 2012 Volume 1, Issue: 2. p129-146.
- [36] BOUSETTA Brahim, EL BEGGAR Omar, GADI Taoufiq. Generating operations specification from domain class diagram using transition state diagram. *International Journal of Computer and Information Technology (IJCIT)* January 2013 Volume 2, Issue: 1. p29-36.
- [37] Hamid Reza Sharifi et al , CIM to PIM Transformation: An Analytical Survey. *Int.J.Computer Technology & Applications*, Vol 3 (2), 791-796

Authors



Brahim BOUSETTA received his Degree in Informatics Engineering from the Hassan II Institute of Agronomy and Veterinary (IAV Hassan II), Statistics and informatics Department in 2007. He later prepared his PhD degree in the Hassan 1st University, faculty of science and technologies of Settat (FSTS) since 2010. His main interests of research are: Software Engineering, Model Driven Engineering and Development on JEE platform. Currently, he is teaching Software Engineering at the University Hassan 1st FST of Settat and a member of LAVETE Laboratory. He has published papers in some prestigious scientific and professional journals and magazines, contributing to the theory and practice of model-driven development and UML. He is the co-author of the book "Guide de la modélisation en UML". You may contact him at ibbousetta@gmail.com.



Omar EL BEGGAR received his Degree in Informatics Engineering from the National School of Computer Science and Systems Analysis (ENSIAS) in 2002. He later prepares his PhD degree in the university Hassan 1st faculty of science and technologies (FST) since 2010. Currently, he is teaching Software Engineering at the University Hassan 1st FST and at the Moroccan office of professional training and employment promotion (OFPPT). He is a member of LAVETE Laboratory and co-author of the book "Guide de la modélisation en UML". His research interest focuses on conceptual modeling, software process, agility systems and modernization of legacy systems.



Taoufiq GADI received his PhD degree from the university Sidi Mohamed Ben Abdellah Faculty of sciences in 1999. Currently, he is Professor at the university Hassan 1st faculty of science and technologies (FST) in Settat and responsible of many engineering formation sections at the same university. He is founder member of the Mediterranean Telecommunication Magazine (RTM), reviewer in many relevant journals and chair in many national and international conferences. He is a director of the 2IDGL Laboratory, author of many books in software engineering and informatics science such as "UML Modeling Guide", "Object Oriented Programming" and leadership of research's teams on Software engineering and 3D Data base indexing. He has valuable contributions and publications in those topics of research. You may contact him at gtaoufiq@yahoo.fr

This academic article was published by The International Institute for Science, Technology and Education (IISTE). The IISTE is a pioneer in the Open Access Publishing service based in the U.S. and Europe. The aim of the institute is Accelerating Global Knowledge Sharing.

More information about the publisher can be found in the IISTE's homepage:

<http://www.iiste.org>

CALL FOR PAPERS

The IISTE is currently hosting more than 30 peer-reviewed academic journals and collaborating with academic institutions around the world. There's no deadline for submission. **Prospective authors of IISTE journals can find the submission instruction on the following page:** <http://www.iiste.org/Journals/>

The IISTE editorial team promises to review and publish all the qualified submissions in a **fast** manner. All the journals articles are available online to the readers all over the world without financial, legal, or technical barriers other than those inseparable from gaining access to the internet itself. Printed version of the journals is also available upon request of readers and authors.

IISTE Knowledge Sharing Partners

EBSCO, Index Copernicus, Ulrich's Periodicals Directory, JournalTOCS, PKP Open Archives Harvester, Bielefeld Academic Search Engine, Elektronische Zeitschriftenbibliothek EZB, Open J-Gate, OCLC WorldCat, Universe Digital Library, NewJour, Google Scholar

