



## AN AUTOMATIC TRANSFORMATION METHOD FROM THE E3VALUE MODEL TO UML2 SEQUENCE DIAGRAMS: AN MDA APPROACH

Nassim Kharmoum <sup>1)</sup>, Soumia Ziti <sup>1)</sup>, Yassine Rhazali <sup>2)</sup>, Fouzia Omary <sup>1)</sup>

<sup>1)</sup> Department of Computer Science, Faculty of Sciences, Mohammed V University in Rabat, Morocco, nassim.kharmoum@um5s.net.ma, ziti.soumia@gmail.com, omaryfouzia@gmail.com

<sup>2)</sup> Department of Computer Science, Higher School of Technology, Moulay Ismail University in Meknes, Morocco, dr.yassine.rhazali@gmail.com

### Paper history:

Received 3 April 2019

Received in revised form 20 August 2019

Accepted 10 September 2019

Available online 30 September 2019

### Keywords:

Model Driven Architecture;

Model transformation;

Automatic transformation;

CIM to PIM Transformation;

Meta-model;

E3value;

UML2 Sequence Diagrams.

**Abstract:** Since the Model Driven Architecture (MDA) approach appearance, model's transformation has been considered as one of the main computing keys in the software industry, which grows exponentially to ensure competitiveness. For that, the Object Management Group (OMG) offers for the MDA approach three abstraction levels, which are Computation Independent Model (CIM), Platform Independent Model (PIM) and Platform Specific Model (PSM). Most researchers put much focus on the PIM to PSM transformation because these levels have multiple common points, but they disregard the CIM to PIM transformation. So, our defiance in this paper is to propose a disciplined method that generates the UML2 Sequence diagrams automatically from the E3value model. Therefore, we produce the UML's behavior and interaction model in the PIM level, from the value-based requirements model in the CIM level. The transformations are done automatically using the Atlas transformation language (ATL). The purpose is to reduce the effort, time, and the development cost of projects as all transformations is automated, increasing the chance of being more competitive in the software industry.

Copyright © Research Institute for Intelligent Computer Systems, 2019.

All rights reserved.

## 1. INTRODUCTION

Nowadays, the majority of organizations are looking for aligning its business with the information system. Therefore, The Object Management Group (OMG) proposes the Model Driven Architecture (MDA), at the beginning of the 21st century [1]. This approach put the models at the center of the development process of any software system [2] that aims to facilitate the development process [3] based on a variety of keys, among them, we mention the transformation of model's task. For that, the OMG proposes three abstraction levels for the MDA approach, which are the higher, the average, and the lower abstraction levels.

The higher level of abstraction called Computation Independent Model (CIM). In this level, the model does not provide any technical consideration or system implementation. The average level called the Platform Independent Model (PIM), models at this level do not contain any

implementation and technical details. Thus, the lower level, which is the Platform Specific Model (PSM) or can be called the model of code, is directly related to the platform execution.

In practice, most researchers put much focus on the PIM to PSM transformation because these levels have multiple common points. However, the transformation from CIM to PIM is rarely discussed; because they are two distinct levels that can contain higher abstraction model nature, such as Requirements models [4, 5]. In this paper, we emphasize on the construction and the transformation from CIM level to PIM level. The CIM level is represented by the E3value model, while the generated PIM level is represented by UML2 Sequence diagrams [6]. So, the E3value source model presents the value-based requirements engineering [7, 8], models and unifies the e-business models notations from a value viewpoint, by showing how the economic value is created, exchanged and consumed within a network of actors

[9]. Thus, the Business Value Model makes models more comprehensible by “Business Executives” and “Business Value Analysts”. The UML’s behavior and interaction model that is generated automatically will be presented by the second version of UML Sequence diagrams. We choose the UML model, as it is the modeling standard for OMG [6]. Those UML diagrams are normally created by “Developers” and “Systems Engineer” to describe and better understand the system behavior. The purpose is to represent in a standard way the interaction of messages between different objects and system actors, taking into consideration the time order of different messages. So, our method will allow stakeholders who do not know about UML2 creation to generate their UML2 Sequence diagrams.

Moreover, we generate the models using ATLAS Transformation Language (ATL) automatically [10] that uses the following OMG standards; such as OCL (Object Constraint Language) [11], MOF (Meta Object Facility) [12] and XMI (XML Metadata Interchange) [13], which will allow us to accelerate the development of the e-Business information systems process by reducing effort and time; therefore, reducing the cost to guarantee competitiveness in the software industry.

To get to the heart of the method, the remainder of the paper is structured as follows: section two shows the related work that generates the UML Sequence diagrams. Section three explains the proposed method, demonstrates the source and the target meta-models, and explains all transformation rules used in the current method. A case study illustrating our method is shown in section four. Section five provides analysis, and discussion of all the results obtained. In the last section, we conclude this contribution and introduce some future work.

## 2. RELATED WORK

This section put much emphasis on the different researches conducted over the last decade. The selected researches generate the PIM level models from CIM level models in the MDA approach, particularly those which generate the UML2 Sequence diagrams in their PIM level. Until now, few methods focus on this generation, but we succeeded in detecting some work done in this regard.

Starting with Kardoš et al. in [14]. They generate the UML Sequence diagram with other UML diagrams; which are Use Case diagram, Activity diagram and Domain diagram in the PIM level, from the Data Flow Diagram (DFD) [15] in the CIM level.

In [16], Bousetta et al. propose a methodology for modeling the CIM level with UML Use Case

diagram, high and low levels of BPMN (Business Process Model and Notation) [17]. Also, its transformation to the UML Domain Class diagram and Sequence diagram of External Systems behavior, which presents the PIM level.

Generating System Sequence behavioral diagram automatically in the PIM level from both the Business Process Model and Notation (BPMN) and the UML Use Case diagram textual description (UCUML) in the CIM level; is proposed by Addamssiri et al. in [18], using QVT [19] transformation language.

Moreover, Kriouile et al. in [20] obtain the behavioral model in the PIM level from the CIM level. They build their CIM level by transforming the Business Process Model and Notation (BPMN) to Use Cases Diagram. After that to represent the behavioral aspect of the PIM-level, they based their work on the achieved Use Cases Diagram to have the System Sequence diagram.

In [21] Yue et al. propose an automated method to generate a UML analysis model comprising UML Class diagram, UML Activity diagram and UML Sequence diagram from a use case model; using their tool called a Toucan, which is implemented with SNLP (Stanford Natural Language Parser) [22].

Furthermore, Zhou et al. apply in [23] an affinity analysis based on the transformation from UML Use Case diagram in the CIM level to UML State Chart diagram, UML Class diagram as well as UML Sequence diagram in the PIM level, by implementing their algorithm with NLP (Natural Language Processing).

Essebaa et al. in their proposition [24] use to model the CIM through UML’s Business Use Case diagram and Activity diagram to cover the structural, functional, and dynamic aspects. As a result, in the PIM level, they generate UML’s Class diagram automatically and Sequence diagram to cover the static and dynamic view, using the QVT (Query/View/Transformation) transformation language.

Last but not least, a Domain Specific Language (DSL) [25] used by Miranda et al. in [26] for automating the generation of UML’s Use Case diagrams, Class diagrams, and Sequence diagrams, from textual Use Cases, based on LUCAM tool.

After studying the above work, we notice that for the source model creation manner, most of the work is based on one or more models in a graphical representation for their CIM level. Also to generate their sequence diagram in the PIM level they make several transformation stages, which are based on a non-automatic way using human language, other methods automatically generate their models using their algorithm or a transformation language.

So, since all of these studies approaches do not

meet our proposal. In this paper, we propose a new method that automatically generates UML’s Sequence diagrams from a model that has value-based requirements with a graphical representation.

The following section will take a closer look at the proposed method.

### 3. PROPOSED METHOD

In this section, we will shed more light on the disciplined proposed method, which is intended to construct and transform the CIM level automatically to the PIM level for e-business information systems, respecting the MDA approach. We stress the fact that the proposal results from our last analytical studies [4, 40, 41].

We generate automatically in this method the UML2 Sequence diagrams from the E3value model (Fig. 1). The source E3value model represents the requirements and business value model in the CIM level. Hence, UML2 Sequence diagrams present the behavior and interaction model in the PIM level and are used to concretize interactions between two entities (objects in a model of software, or sub-

correct E3value model and UML2 Sequence diagrams. We aim to use the models of our method as a base for the system’s understanding, and facilitating communication between technical and e-business stakeholders such as “Developers”, “Systems Engineer”, “Business executives” and “Business Analysts”. Also, to complete the development process of our e-business information system by generating other models in the lower abstraction level PSM.

To illuminate the proposed method according to the MDA approach, Fig. 2 illustrates its transformation process. We start with the definition of meta-models; the purpose is to create and generate correct models and to perform automated model transformations [27]. So, the source meta-model is E3value meta-model, describing the structure of the Business Value Model and requirements on the higher abstraction level. After that, we determine the targets UML2 Sequence meta-model representing the structure of the generated UML’s behavior and interaction model in the average abstraction level. Then, we specify the

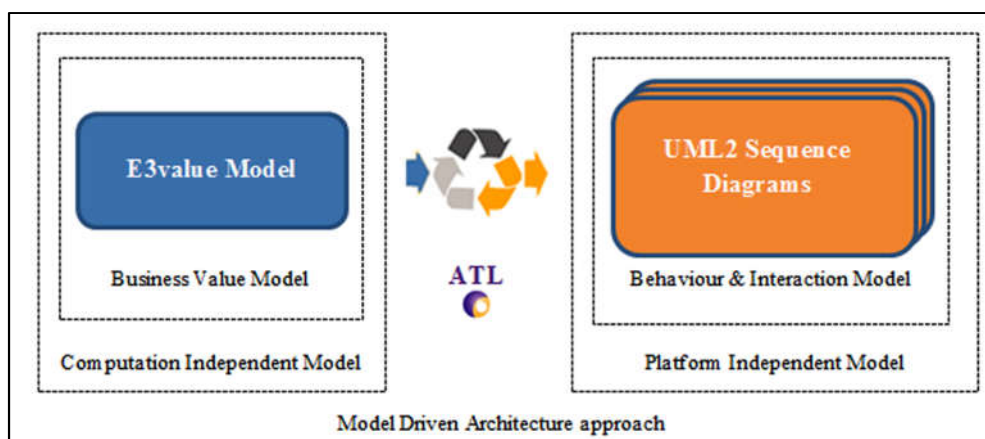


Figure 1 – The Proposed method

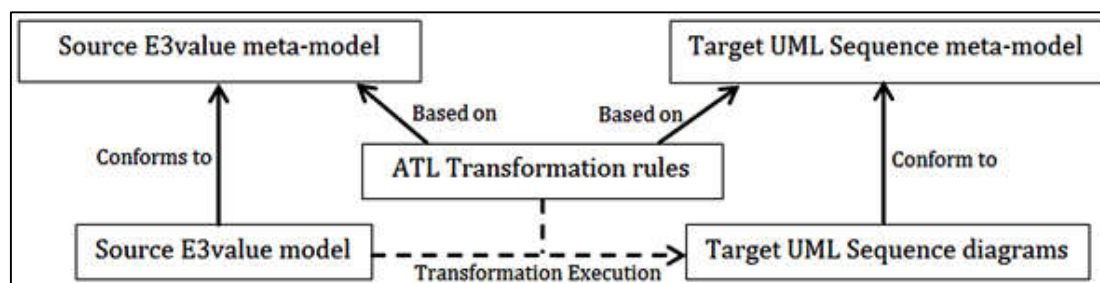


Figure 2 – The Transformation process of the proposed method

systems in a model of a complete system) in the PIM level. We add that the transformation is done automatically via ATL transformation language. The purpose is to have an e-business system containing

constraints during the source model construction. Moreover, we explain the set of mapping rules implemented via ATL transformation language to

generate the target models automatically from the source model.

### 3.1 SOURCE E3VALUE META-MODEL

This part describes the structure of the E3value source meta-model that any created E3value model must conform to. The purpose is to create a correct model explaining the manner that economic value is created, exchanged, and consumed in a network of several actors. Therefore, the E3value meta-model (Fig. 3) is made of the following elements, which are taken from [28–30] and proposed by Gordijn et al.:

- «Actor»: an independent economic entity can be a consumer or a producer of the economic value. We can have two types of actors «Elementary Actor»: represents an individual actor and «Market Segment»: represents a set of actors that have common properties;
- «Value Object»: is the economic value exchanged by the «Actors», it can be a service, a good, an experience, or money;
- «Value Port»: An actor that provides «Value Objects» via «Port Out» or requests them via «Port In» «Value Objects» to or from other «Actors». The «Port Out» and «Port In» are used to extend «Value Port»;
- «Value interface»: assembles several «Value Ports» and showing economic reciprocity;
- «Value Exchange»: connects two «Value Ports» to exchange potential «Value Objects». For the meta-model, the «Value

Exchange» can have type «request» or «response»;

- «Value Activity»: presents one or more operational activities that an «Actor» can process to yield a profit;
- «Value Transaction»: groups a set of «Value Exchanges» that participate in the same economic transaction;
- «Dependency Element»: various kinds exist, «Element AND» and «Element OR»; these two elements accept as behavior: «Join», «Fork» or both «Fork\_Join». We also have «Stimulus Element» that can be «Stimulus Start» or «Stimulus End»;
- «Connect Element»: connects all «Dependency Elements» between them or with «Value interface» by specifying the source and target for each «Connect Element»;
- «Comment»: used to add a comment or a note for all the elements mentioned above;
- Finally, to rearrange the E3value meta-model, all elements are grouped in «E3valueModel» which are all of the type «E3valueObject», and all its instances are identified by the «Name» attribute.

### 3.2 TARGET UML2 SEQUENCE META-MODEL

We explain in this part the construction of the UML2 target meta-model that any generated UML2

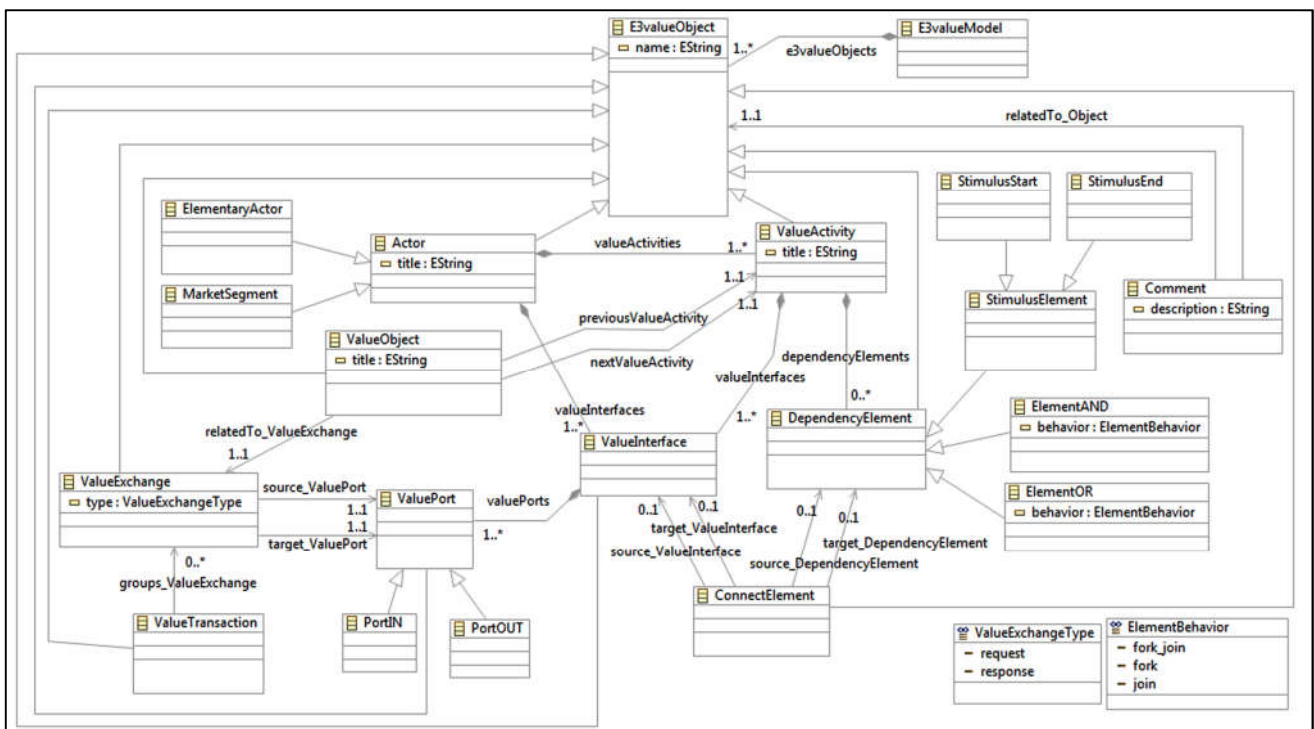


Figure 3 – The E3value meta-model

diagram must conform to. The UML2 Sequence diagrams are generated automatically from the E3value source model. They explain the system behavior, by representing in a standard way the messages' interaction between different objects and system actors, taking into consideration the time order of different messages. The UML2 Sequence meta-model (Fig. 4) elements are extracted from [31], which are proposed by OMG:

- «Actor»: represents an individual participant in a sequence diagram, in this case, it represents an independent economic entity maybe a producer or consumer of the proposed economic values;
- «Lifeline»: represents an object that interacts with other objects in a model of software, or sub-systems in a model of a complete system. The purpose is to exchange messages during the sequence. Moreover, Jacobson [32] specified new roles for the element «Lifeline», and we find: «Boundary», «Control» and «Entity»;
- «Boundary»: a type of «Lifeline» which is used to model the interactions between the system and its actors;
- «Control»: a type of «Lifeline» which is used to represent coordination, linking and controlling other objects;
- «Entity»: a type of «Lifeline» which is served to model durable and often persistent information;
- «Message»: defines a particular communication between «Lifeline» or in general between objects. It can have

«MessageParameter» and/or «MessageReturn». And every «Message» has a type that expresses its behavior. It can be synchronous «Message» or asynchronous «Message», and also has the attribute «order» to take into consideration the time order of different messages;

- «Fragment»: is a new element in the UML2 Sequence diagram that describes the execution modes of the «Messages». Every «Fragment» has a type that expresses its behavior, and accepts «Conditions»;
- «Condition»: is a new element in the UML2 Sequence diagram, which is integrated with «Fragments»;
- «Gate»: is an element that connects messages outside a fragment with a message inside a fragment;
- «Endpoint»: reflects found or lost «Message» in a sequence.
- Finally, to rearrange the UML2 Sequence meta-model, all elements are grouped in «UMLSequenceModel», which are all of the type «SequenceObject», and all its instances are identified by the «Name» attribute.

### 3.3 SOURCE E3VALUE MODEL CONSTRUCTION RULE

For the source model creation, the only constraint is to respect the meta-model that is previously explained and illustrated in Fig. 3 to create a correct source E3value model.

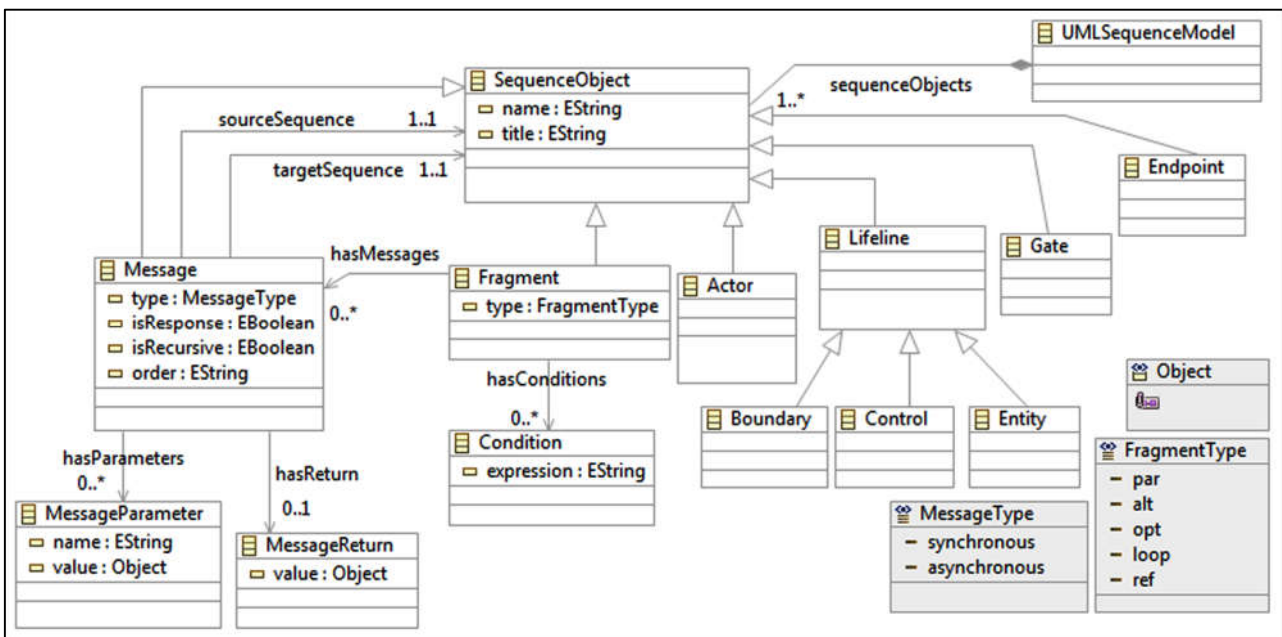


Figure 4 – The UML2 Sequence diagrams meta-model [6]

### 3.4 TRANSFORMATION RULES FROM E3VALUE MODEL TO UML2 SEQUENCE DIAGRAMS

To transform the Business Value Model automatically to the UML's behavior and interaction model, this sub-section lists the E3value model transformation rules toward the UML2 Sequence diagrams based on their meta-models.

Moreover, to generate a well-detailed e-business information system, we will not generate all sequences in a single diagram, but we will generate sequence diagrams based sub-function system. In our proposal, the «Value Activity» element is considered as a sub-function for each e-business system. So, this part details the transformation rules for each generated UML2 Sequence diagram via simple diagrams, and ATL transformation language (Figs. 5 and 6).

We start by specifying the name of E3value «Value Activity» to be generated to UML2 Sequence diagram using the variable «selectedValueActivity» (see Fig. 5).

“Rule 1”: presents the transformation from E3value «Actor» (it can be «Elementary Actor» or «Market Segments») to UML2 Sequence diagram «Actor». We select just the «Actor» who performs the selected «Value Activity»; via the helper «isActorInclude()». Thus, the source «Actor» has the same «name» and «title» as the target one (Fig. 5).

“Rule 2”: allows transforming the selected «Value Activity» to «Lifeline» and we add suffix «sub-system» to the «name» and «title» because we model a complete system. To show the interaction between Sequence diagram elements, we add to this rule four «Messages» types between the generated «Actor» and the current «Lifeline», which are: «messageRequest», «messageRecursive», «messageResponse\_favorable» and «messageResponse\_unfavorable». All of these messages are synchronous message because the sender waits for the receiver to process the message and return before carrying on with another message. Those messages have the following attributes: «order» specifies the execution order of the message, «name» gives the name of each message, «title» specifies the name of each message, «sourceSequence» mentions the source element of each message and «targetSequence» mentions the target element of each message. The «order» attribute's value is calculated according to two helpers: getCounter() and getCounterForLastMessages().

- The getCounter() helper used to specify the time execution for all messages. We initialize it with the value "1" for the first message after

we increment its value for each new message generated;

- The getCounterForLastMessages () helper, used especially for the last two messages of each sequence diagram "messageResponse\_favorable" and "messageResponse\_unfavorable". Both messages have the same value.

The «Message» «name» accepts «parameters» and «message\_return». Thus the message structure is: (*message\_title (parameters): message\_return*). While the message\_title construction accepts several cases:

- We add the prefix "need" for the requested message;
- We add the prefix "verify" for the recursive message;
- We add the prefix "give" for the favorable response message;
- We add the prefix "stop" for the unfavorable response message.

Also, in the “Rule 2”, we generate the element «Fragment» that groups the generated two messages «messageResponse\_favorable» and «messageResponse\_unfavorable», with «type» «alt» (alternative operator) (Fig. 5).

“Rule 3”: transforms «Value Activities» that connect the selected «Value Activity» (using helper «isValueActivityInclude()»), to «Lifeline» and we add suffix «service» to the «name» and «title» because we model a complete system (Fig. 6).

“Rule 4”: generates the interaction messages between the generated «Lifeline» in “Rule 2”, with the generated «Lifeline» in “Rule 3”. So, we also generate for that four «Messages» which are: «messageRequest», «messageRecursive», «messageResponse\_favorable» and «messageResponse\_unfavorable»; all of these messages have the following attributes: «order», «name», «title», «sourceSequence» and «targetSequence». In this rule, the value of the «order» attribute is calculated according to two helpers: getCounter() (as used in “Rule2”) and getCounterWithoutIncrement().

○The getCounterWithoutIncrement() helper used especially for messages type "messageResponse\_unfavorable" so as not to increment its value of the time order.

Also, in the “Rule 4”, we generate the element «Fragment» that groups the generated two messages «messageResponse\_favorable» and «messageResponse\_unfavorable», with «type» «alt» (Fig. 6).

<p><b>Initialization:</b> We initialize the name of «Value Activity» to be generated, for example in our case study we choose: 'accessing_library_resources' :</p> <pre>helper def : selectedValueActivity : String = 'accessing_library_resources';</pre>
<p><b>Rule1:</b> transforms e3value «actor» to UML Sequence «actor»</p> <pre>rule Rule1 {   from     actor : MME3value!Actor(actor.oc1IsKindOf(MME3value!Actor) and                           actor.isActorInclude())   to     sequenceActor : MMUMLSequence!Actor (       name &lt;- actor.name,       title &lt;- actor.title     ) }</pre>
<p><b>Rule2:</b> Transforms selected «valueActivity» to «lifeline», and generate: «messageRequest», «messageRecursive», «messageResponse_favorable», and «messageResponse_unfavorable». Also generate «fragments».</p> <pre>rule Rule2 {   from     valueActivity : MME3value!ValueActivity(valueActivity.name =   thisModule.selectedValueActivity)   to     lifeline : MMUMLSequence!Lifeline (       name &lt;- valueActivity.name + '_sub-system',       title &lt;- valueActivity.title + ' sub-system'     ),     messageRequest : MMUMLSequence!Message (       order &lt;- thisModule.getCounter(),       name &lt;- 'need_' + valueActivity.name,       title &lt;- messageRequest.order + ': need ' + valueActivity.title,       sourceSequence &lt;- valueActivity.refImmediateComposite(),       targetSequence &lt;- valueActivity     ),     messageRecursive : MMUMLSequence!Message (       order &lt;- thisModule.getCounter(),       name &lt;- 'verify_' + valueActivity.name,       title &lt;- messageRecursive.order + ': verify ' +             valueActivity.title,       sourceSequence &lt;- valueActivity,       targetSequence &lt;- valueActivity,       isRecursive &lt;- true     ),     messageResponse_favorable : MMUMLSequence!Message (       order &lt;- valueActivity.getCounterForLastMessages() ,       name &lt;- 'give_' + valueActivity.name,       title &lt;- messageResponse_favorable.order + ': give ' +             valueActivity.title,       sourceSequence &lt;- valueActivity,       targetSequence &lt;- valueActivity.refImmediateComposite(),       isResponse &lt;- true     ),     messageResponse_unfavorable : MMUMLSequence!Message (       order &lt;- valueActivity.getCounterForLastMessages(),       name &lt;- 'stop_' + valueActivity.name,       title &lt;- messageResponse_unfavorable.order + ': stop ' +             valueActivity.title,       sourceSequence &lt;- valueActivity,       targetSequence &lt;- valueActivity.refImmediateComposite(),       isResponse &lt;- true     ),     fragment : MMUMLSequence!Fragment (       name &lt;- messageRequest.name + '_response',       title &lt;- 'need ' + valueActivity.title + ' response',       type &lt;- #alt,       hasMessages &lt;- Sequence{messageResponse_favorable,                              messageResponse_unfavorable}     ) }</pre>

Figure 5 – Transformation rules from E3value model to UML2 Sequence diagram –Part1

<p><b>Rule3: Transforms other «valueActivity» to «lifelines».</b></p> <pre> rule Rule3 {   from     valueActivity : MME3value!ValueActivity(valueActivity.name &lt;&gt;  thisModule.selectedValueActivity and  valueActivity.isValueActivityInclude())   to     lifeline : MMUMLSequence!Lifeline (       name &lt;- valueActivity.name + '_service',       title &lt;- valueActivity.title + ' service'      ) } </pre>
<p><b>Rule4: Generates all «messages» between «lifelines» created in Rule3.</b></p> <pre> rule Rule4 {   from     valueObject : MME3value!ValueObject(       valueObject.relatedTo_ValueExchange.type = #response and       valueObject.nextValueActivity.name =       thisModule.selectedValueActivity )   to     messageRequest : MMUMLSequence!Message (       order &lt;- thisModule.getCounter(),       name &lt;- 'need_' + valueObject.name,       title &lt;- messageRequest.order + ': need ' + valueObject.title,       sourceSequence &lt;- valueObject.nextValueActivity,       targetSequence &lt;- valueObject.previousValueActivity     ),     messageRecursive : MMUMLSequence!Message (       order &lt;- thisModule.getCounter(),       name &lt;- 'verify_' + valueObject.name,       title &lt;- messageRecursive.order + ': verify ' +valueObject.title,       sourceSequence &lt;- valueObject.previousValueActivity,       targetSequence &lt;- valueObject.previousValueActivity,       isRecursive &lt;- true     ),     messageResponse_favorable : MMUMLSequence!Message (       order &lt;- thisModule.getCounter(),       name &lt;- 'give_' + valueObject.name,       title &lt;- messageResponse_favorable.order + ': give ' +         valueObject.title,       sourceSequence &lt;- valueObject.previousValueActivity,       targetSequence &lt;- valueObject.nextValueActivity,       isResponse &lt;- true     ),     messageResponse_unfavorable : MMUMLSequence!Message (       order &lt;- thisModule.getCounterWithoutIncrement(),       name &lt;- 'stop_' + valueObject.name,       title &lt;- messageResponse_unfavorable.order + ': stop ' +         valueObject.title,       sourceSequence &lt;- valueObject.previousValueActivity,       targetSequence &lt;- valueObject.nextValueActivity,       isResponse &lt;- true     ),     fragment : MMUMLSequence!Fragment (       name &lt;- messageRequest.name + '_response',       title &lt;- messageRequest.title + ' response',       type &lt;- #alt,       hasMessages &lt;- Sequence{messageResponse_favorable,         messageResponse_unfavorable}     ) } </pre>
<p><b>Rule5: Groups all generated elements in Rule1, Rule2, Rule3 and Rule4 in «UML Sequence Model».</b></p> <pre> rule Rule5 {   from     e3valueModel : MME3value!E3valueModel   to     UMLSequenceModel : MMUMLSequence!UMLSequenceModel (       sequenceObjects &lt;- MMUMLSequence!Actor.allInstances()         -&gt; union(MMUMLSequence!Lifeline.allInstances())         -&gt; union(MMUMLSequence!Fragment.allInstances())         -&gt; union(MMUMLSequence!Message.allInstances())     ) } </pre>

Figure 6 – Transformation rules from E3value model to UML2 Sequence diagram –Part2



“Rule 5”: generates the « UMLSequence Model» from «E3value Model». The generated model groups all Fig. 6 generated elements, and in this case, we have: «Actors», «Lifelines», «Fragments» and «Messages».

#### 4. CASE STUDY

To illustrate our method, this section shows the

«.ecore\_diagram»), the second contains models (with «.xmi» extensions) while the third folder covers the transformation rules (with extensions «.asm» and «.atl»). For more ergonomics, each folder contains two sub-folders; one for the source (E3value), while the other is for targets (UML2 Sequence diagrams).

To better position the Fig. 7 in the proposed method, the table 1 links each file in Fig. 7 with its

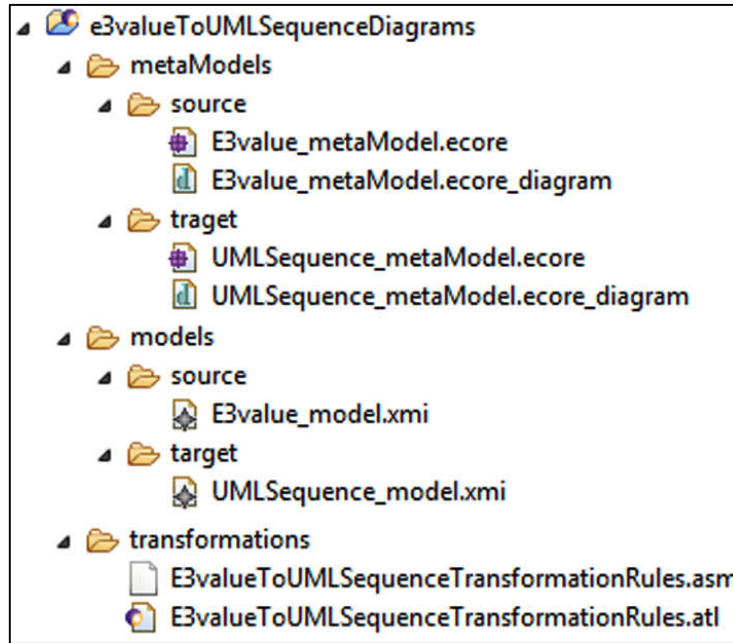


Figure 7 – The Practical case structure

Table 1. The case study files and its corresponding figures

Files	Explanation	Corresponding figures
E3value_metaModel.ecore and E3value_metaModel.ecore diagram	illustrate the E3value source meta-model	Fig. 3
UMLSequence_metaModel.ecore and UMLSequence_metaModel.ecore diagram	illustrate the UML2 Sequence target meta-model	Fig. 4
E3valueToUMLSequenceTransformationRules.atl and E3valueToUMLSequenceTransformationRules.asm	present the transformation rules from the E3value model to UML2 Sequence diagram	Figs. 5 and 6
E3value_model.xmi	represents graphically the proposed E3value-source model in XMI format [13]	Fig. 8
UMLSequence_model.xmi	represents graphically the generated UML2 Sequence diagram in XMI format	Fig. 9

theoretical and practical aspect of the “University Library management” case study transformation through ATL Eclipse plugin, as a standard tool for transformation [33], to exemplify the transformation from Business Value model to UML Behavior and interactions model. Fig. 7 illustrates the practical case study in three main folders; the first contains meta-models (has extensions «.ecore» and

corresponding created and generated figures.

We suggest the following «Actors»: «Patrons», «Library», «Publishing Companies» and «Research Databases». We have that every «Patrons» can access «Library» resources. For instance, we have «Books» and «Online Research Articles» by paying fees for the chosen access. From its part, the

«Library» is always concentrating to the «Patron's» requests; it lends «Books», browses «Research Articles» if the requested need exists, if not, it manages its needs by communicating them to suppliers who are «Publishing Companies» to offer the requested «Books» and «Research Databases» to give «Databases» desired access. The purchase of each «Book» or «Research Database» access automatically requires fees.

The rest of the section illustrates the «University Library Management» case study source and target models.

#### 4.1 E3VALUE MODEL PROPOSAL

We focus, in the source model, on the E3value model offering the requirements and business value model of this case study, as illustrated in Fig. 8.

The «Actors» can be «Elementary Actors» such as «library» or can be «Market Segments» like «patrons», «publishing companies» or «research databases». Each «Actor» can have at least one «Value Activity». In this model the «patrons» can «access library resources», the «library» has «lend books», «browsing research articles» and «managing library needs», the «publishing companies» have «sell books», and «research databases» have «sell access to research databases». The «Value Objects» can be a request like «access to books registration

The E3value model exploits «Value Exchange» to explain the dependency path direction. These dependency paths connect E3value dependency elements, which can be «Start Stimulus», «End Stimulus», «And-Join», «And-Fork», «Or-Join» or «Or-Fork».

#### 4.2 GENERATED UML2 SEQUENCE DIAGRAM

In this section, we describe the generating model, based on the set of defined elements that constitute the source model, the target models, and the set of previous transformations explained.

As mentioned before and to properly obtain detailed UML2 Sequence diagrams, we will generate them according to their system sub-functions. In our case, we will generate a sequence diagram per E3value «Value Activity», so we can have six UML2 Sequence diagrams, which correspond to the following «Value Activities»: «accessing library resources», «lending books», «browsing research articles», «managing library needs», «selling books» and «selling access to research databases».

Seen that all the sequence diagrams will be generated in the same way, in this case, we shed more light on the first «Value Activity» which is «accessing library resources» Fig. 9. As a result, the chosen UML2 sequence diagram will be generated

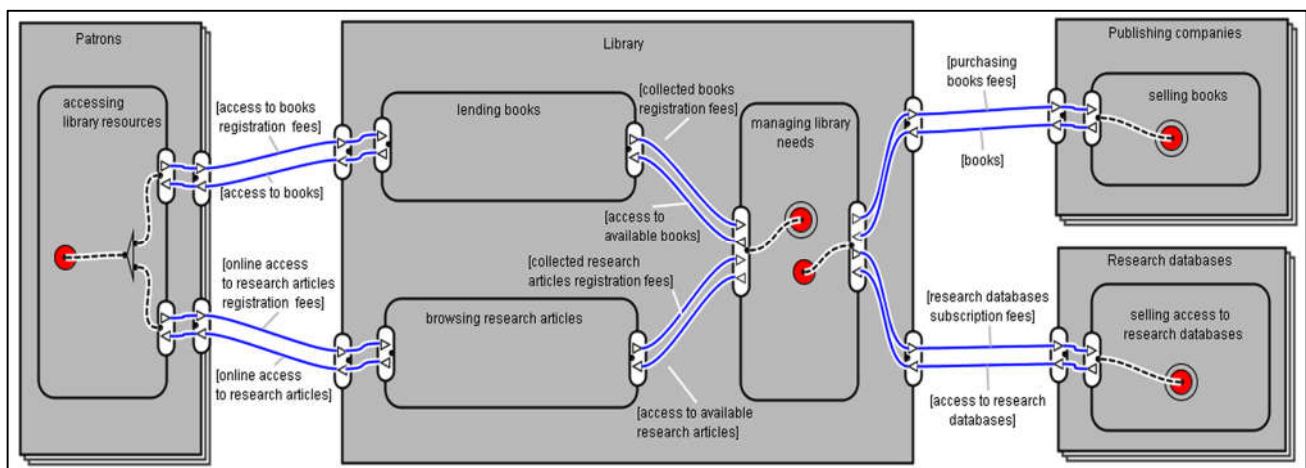


Figure 8 – The E3value-source model

fees», «online access to research articles registration fees», «collected books registration fees», «collected research articles registration fees», «purchasing books fees» and «research databases subscription fees». Moreover, «Value Objects» can be a response such as «access to books», «online access to research articles», «access to available books», «access to available research articles», «books», and «access to research databases».

as follows: the «Market Segments»: «library» is transformed to «Actor», to present the individual participant in a sequence diagram. Then the selected «Value Activity»: «accessing library resources» will be transformed to main «Lifeline» in this sequence with the name «Accessing library resources subsystem». «Value Activities» that connect the selected «Value Activity»: «accessing library resources» are transformed to «Lifelines» and in this case, we have two «Lifelines»: «Lending books

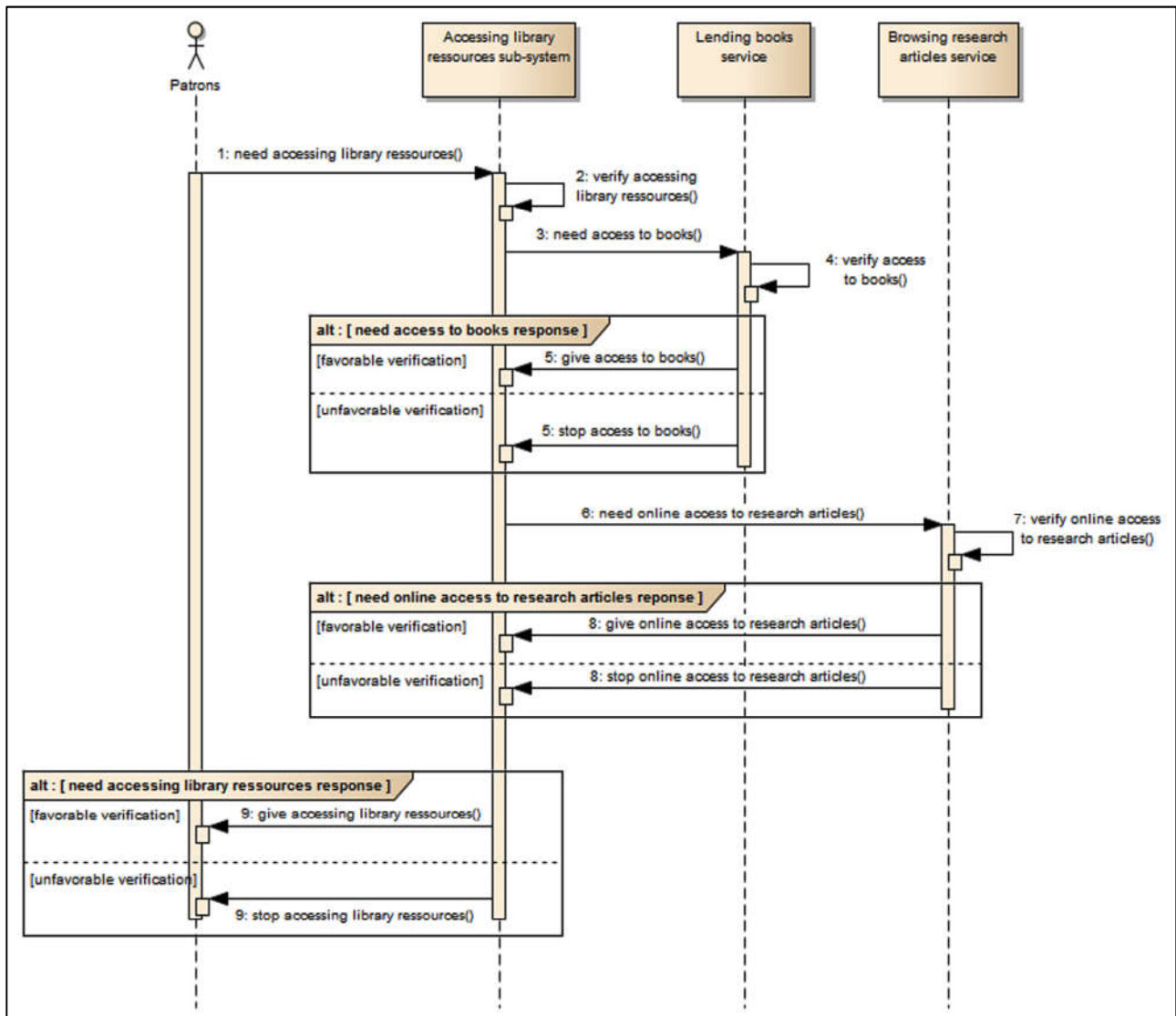


Figure 9 – The Generated UML2 Sequence diagram for “accessing library resources.”

service» and «Browsing research articles service». To ensure the interaction between the generated elements, we use message elements. For this method, we propose four message types: the first is to request a service, so we add the prefix «need» for example: «need accessing library resources». The second message is for verifying what is asked in the previous message, for this, we propose to add the prefix «verify», for example: «verify accessing library resources». The third and the fourth message types are grouped in the same «Fragment» whose type is «alt» (alternative operator) because this type of «Fragments» accept conditions. Thus, we suggest two conditions; the first is favorable (carries the third message, like «give accessing library resources»), the second condition is unfavorable (carries the fourth message, for example, «stop accessing library resources»). Every «Message» has the attribute «order» to take into consideration the time order of different messages. We place the «order» value before every «Message».

## 5. ANALYSIS AND DISCUSSION

In this section, we analyze and discuss the proposed method based on an analytical survey of all studied related work. For that purpose, we focus on the OMG recommendations [1] and following work [33, 34–37] to deduce the evaluation criteria.

So, we group the deduced criteria according to their model level construction, the model transformation manner, and the use of assessment methodology.

- Model construction criteria: present the coverage of the graphical representation for each model. And the participation in one of the MDA approach levels;
- Transformation criteria: Verify transformation automation, and meta-models based transformation to ensure the correctness of the transformation, which are one of the essential

Studied papers	Source model				Generated UML Sequence diagram		Transformation					Assessment Methodology	
	Graphical representation	MDA level	Representation	Requirements model coverage	Graphical representation	MDA level	Based Meta-models	Is Automatic	Type	Representation language	Mapping rules Definition	Theoretical	Practical
Kardoš et al. [14]	Yes	CIM	Data Flow diagram (DFD)		Yes	PIM			Human language			Case study	
Bousetta et al [16]	Yes	CIM	Business Process Model and Notation (BPMN) & UML Use Case diagram	Yes	Yes	PIM			Human language		Yes	Case study	
Addamssiri et al. [18]	Yes	CIM	Business Process Model and Notation (BPMN)	Yes	Yes	PIM	Yes	Yes	Transformation language	QVT (Query / View / Transformation)	Yes	Case study	
		CIM	UML Use Case textual description										
Kriouile et al. [20]	Yes	CIM	Business Process Model and Notation (BPMN) & UML Use Case diagram	Yes	Yes	PIM	Yes		Human language		Yes	Case study	
Yue et al. [21]	Yes	CIM	UML Use Case diagram	Yes	Yes	PIM	Yes	Yes	Algorithm (Based aToucan tool)	SNLP (Stanford Natural Language Parser)	Yes	Case study	Personal Tool (aToucan)
Zhou et al. [23]	Yes	CIM	UML Use Case diagram	Yes	Yes	PIM		Yes	Algorithm (Based personnel tool)	NLP (Natural Language Processing)	Yes	Case study	Personal tool
Essebaa et al. [24]	Yes	CIM	UML Use Case diagram & UML activity diagram	Yes	Yes	PIM	Yes	Yes	Transformation language	QVT (Query / View / Transformation)	Yes	Case study	
Miranda et al. [26]			UML Use Case textual description	Yes	Yes			Yes	Algorithm (Based LUCAM tool)	DSL (Domain Specific Languages)	Yes	Case study	Personal tool (LUCAM)
Our proposal	Yes	CIM	E3value model	Yes	Yes	PIM	Yes	Yes	Transformation language	ATL (Atlas Transformation Language)	Yes	Case study	Standard tool (Eclipse)

Figure 10 – Studied papers Comparison via Evaluation criteria

concepts in MDA [1, 38]. Also, we check the existence of the mapping rules definition;

- Evaluation approach criteria: Present the occurrence of the assessment methodology for theoretical and practical ways.

The rest of this section analyzes the studied work with the proposed method based on an analytical survey from the concluded criteria.

### 5.1 ANALYSIS

We present in Fig. 10, the result of the evaluation criteria analysis. The figure columns give the deduced criteria, whereas the figure lines show studied papers.

We start with model construction criteria; on the one hand, we have all methods do the transformation from CIM to PIM to generate their UML Sequence diagram, except the authors of the method [24], they did not locate their models in the MDA approach. On the other hand, most methods prefer the use of a graphical representation for their source and destination models; except both methods [18] and [26]; they use a textual representation for their UML Use Case to generate their UML Sequence diagram.

Concerning the transformation criteria, we have just five methods which are based on the meta-model such as [18], [20], [21], [24] and our proposed method. Thus, for the transformation automation, we notice some methods; use their algorithm [39] like [21], [23] and [26] while other methods transform their models relying on a transformation language; for instance, QVT for [18] and [24] or ATL in our proposal. However, we have other methods that transform their models manually relying on Human language like [14], [16] and [20]. Plus, for the mapping rules definition, most of the studied methods clearly explain their transformation rules, except the [14] method.

Besides, for the evaluation approaches criteria; all methods use a case study for their theoretical assessment methodology. However, just four methods use a practical assessment methodology; for example, methods [21], [23] and [26] use their tool, while, in this method, we use a standard tool «Eclipse» [33].

### 5.2 DISCUSSION

After analyzing all the studied methods, we have that the paper [21] validates all the proposed criteria. The authors generate their UML Sequence diagram

automatically in the PIM level from UML Use Case diagram in CIM level using an algorithm based on their tool called aToucan, and they use a theoretical and practical assessment methodology based on a case study in their aToucan tool.

In the proposed method, we validate all proposed criteria, which is the unique method that generates the second version of UML Sequence diagram in PIM level starting with a straightforward model which is the E3value model in CIM level. Also, the executions of the proposed transformations rules are meta-model based; forasmuch these diagrams are generated automatically using ATL Transformation language. We use theoretical and practical assessment methodology based on a case study with a standard tool.

Finally, using our method for e-business systems will allow generating correct behavior and interaction models in the PIM level based on a simple method which can be created by non-technical stakeholders who do not know about computer modeling. On the one hand, it will reduce the effort, time, and therefore, the development cost of e-business projects as all transformations is automated increasing the chance of being more competitive in the software industry. On the other hand, relying on the generated UML2 Sequence diagrams will facilitate the construction and the generation of other models for other lower levels such as PSM, which is one of MDA experiments.

## 6. CONCLUSION

Proposing a disciplined method for e-business information systems was our challenge in this paper. To overcome this challenge, we transform the E3value model automatically to the UML Sequence diagrams. The source model serves to model value-based requirements in the CIM level. Whereas the target models reproduce the UML's behavior and interaction model in the PIM level, using the ATL transformation language.

In future work, we will complete the e-business Information System construction, focusing on the generated UML Sequence diagrams by proposing a new disciplined method that produces models in the PSM level, while respecting the MDA approach.

## 7. REFERENCES

- [1] MG-MDA, *MDA Guide Version 2.0*, 2014.
- [2] J. Bézévin, "Sur les principes de base de l'ingénierie des modèles," *RSTI-L'Objet*, vol. 10, no.4, pp. 145–157, December 2004. (in French)
- [3] E. Maatougui, C. Bouanaka and N. Zeghib, "Towards a meta-model for quality-aware self-adaptive systems design," *Proceedings of the 3rd International Workshop on Interplay of Model-Driven and Component-Based Software Engineering*, October 2016, vol. 1723, pp. 12–18.
- [4] N. Kharmoum, S. Ziti and F. Omary, "An alytical study of the CIM to PIM transformation in MDA," *Proceedings of the International Workshop on COmputing Sciences (WCOS'16)*, Kenitra, Morocco, December 2016, pp. 14–19.
- [5] Y. Rhazali, Y. Hadi, I. Chana, M. Lahmer and A. Rhattoy, "A model transformation in model driven architecture from business model to web model," *IAENG International Journal of Computer Science*, vol. 45, no 1, pp 104–117, 2018.
- [6] OMG, *Unified Modeling Language (UML)-Version 2.5.1*, December 2017.
- [7] J. Gordijn and J. M. Akkermans, "Value-based requirements engineering: exploring innovative e-commerce ideas," *Requir. Eng.*, vol. 8, no. 2, pp. 114–134, Jul. 2003.
- [8] I. Garrigós, J.N. Mazón, N. Koch, and M. Escalona, "Editorial: web and requirements engineering," *IET Softw.*, vol. 6, no. 2, pp. 83–84, 2012.
- [9] J. Gordijn, J.M. Akkermans, "Designing and evaluating e-business models," *IEEE Intell Syst-Intelligent e-Business*, vol. 16, no. 4, pp. 11–17, 2001.
- [10] F. Jouault, F. Allilaire, J. Bézivin, I. Kurtev, and P. Valduriez, "ATL: A QVT-like transformation language," *Proceedings of the 21st ACM SIGPLAN Symposium on Object-oriented Programming Systems, Languages, and Applications*, New York, NY, USA, 2006, pp. 719–720.
- [11] OMG, *Object Constraint Language (OCL)-Version 2.4*, February 2014.
- [12] OMG, *Meta Object Facility (MOF)-Version 2.5*, June 2015.
- [13] OMG, *XML Metadata Interchange (XMI)-Version 2.5.1*, June 2015.
- [14] M. Kardoš and M. Drozdová, "Analytical method of CIM to PIM transformation in Model Driven Architecture (MDA)," *Journal of Information and Organizational Sciences*, vol. 34, no.1, pp. 89-99, 2010.
- [15] T. Hathaway and A. Hathaway, *Data Flow Diagrams – Simply Put!: Process Modeling Techniques for Requirements Elicitation and Workflow Analysis*, CreateSpace Independent Publishing Platform, 2016.
- [16] B. Bousetta, O. El Beggar and T. Gadi, "A methodology for CIM modeling and its transformation to PIM," *J. Inf. Eng. Appl.*, vol. 3, no. 2, pp. 1–21, 2013.

- [17] BPMN2 OMG, *Business Process Model and Notation (BPMN)-Version 2.0*, January 2011.
- [18] N. Addamssiri, A. Kriouile, Y. Balouki and T. Gadi, "Generating the PIM behavioral model from the CIM using QVT," *Journal of Computer Science and Information Technology*, vol. 2, no. 3 & 4, pp. 55–81, December 2014.
- [19] OMG, *Meta Object Facility (MOF) 2.0 Query/View/Transformation Specification, OMG Adopted Specification Version 1.3*, June 2016.
- [20] A. Kriouile, T. Gadi, N. Addamssiri and A. El Khadimi "Obtaining behavioral model of PIM from the CIM," *Proceedings of the 2014 IEEE International Conference on Multimedia Computing and Systems (ICMCS)*, 2014, pp. 949–954.
- [21] T. Yue, L.C. Briand and Y. Labiche, "aToucan: An automated framework to derive UML analysis models from use case models," *ACM Transactions on Software Engineering and Methodology*, vol.24, no.3, article number 13, 2015.
- [22] L. Kof, "Natural language processing: mature enough for requirements documents analysis?," In: Montoyo A., Muñoz R., Métais E. (eds) *Natural Language Processing and Information Systems. NLDB 2005. Lecture Notes in Computer Science*, vol 3513. Springer, Berlin, Heidelberg, 2005, pp. 91–102.
- [23] D. Zhou, X. Chen, Q. Jin, Z. Kuang and H. Yang, "An affinity analysis based CIM-to-PIM transformation," *Multiagent and Grid Systems*, vol. 13, no. 3, pp. 269–286, 2017.
- [24] I. Essebaa and S. Chantit, "QVT Transformation Rules to Get PIM Model from CIM Model," *Europe and MENA Cooperation Advances in Information and Communication Technologies. Springer, Cham*, pp. 195–207, 2017.
- [25] A. Van Deursen, P. Klint and J. Visser, "Domain-specific languages: An annotated bibliography," *ACM Sigplan Notices*, vol. 35, no. 6, pp. 26–36, 2000.
- [26] M.A. Miranda, M. G. Ribeiro, H.T. Marques-Neto and M. A. J. Song, "Domain-specific language for automatic generation of UML models," *IET Software*, vol. 12, no. 2, pp. 129–135, 2018.
- [27] A. Rodríguez, I. G.-R. de Guzmán, E. Fernández-Medina, and M. Piattini, "Semi-formal transformation of secure business processes into analysis class and use case models: An MDA approach," *Inf. Softw. Technol.*, vol. 52, no. 9, pp. 945–971, Sep. 2010.
- [28] J. Gordijn and J. M. Akkermans, "Value-based requirements engineering: exploring innovative e-commerce ideas," *Requir. Eng.*, vol. 8, no. 2, pp. 114–134, Jul. 2003.
- [29] J. Gordijn, E. Yu and B. van der Raadt, "e-Service Design Using i\* and E3value Modeling," *IEEE Software*, vol. 23, no. 3, pp. 26–23, 2006.
- [30] J. Gordijn and H. Akkermans, *Value Webs: Understanding e-Business Innovation*, The Value Engineers B.V. Soest, NL, 2018.
- [31] OMG, *Unified Modeling Language (UML)-Version 2.5.1*, December 2017.
- [32] I. Jacobson, M. Christerson, P. Jonsson and G. Overgaard, *Object-Oriented Software Engineering: A Use Case Driven Approach*, Addison-Wesley, 1992.
- [33] Y. Rhazali, Y. Hadi, and A. Mouloudi, "A Methodology of Model Transformation in MDA: from CIM to PIM," *International Review on Computers and Software (IRECOS)*, vol. 10, no. 12, p. 1186–1201, Dec.2015.
- [34] Y. Rhazali, Y. Hadi and A. Mouloudi, *Transformation Des Modèles Depuis CIM Vers PIM Dans MDA*, Saarbrücken, Germany, Noor Publishing, 2016.
- [35] A. Kriouile, T. Gadi, Y. Balouki, "CIM to PIM Transformation: A criteria Based Evaluation," *International Journal Computer Technology & Applications*, vol. 4, no. 4, pp. 616-625, 2013.
- [36] T. Yue, L. Briand and Y. Labiche "A systematic review of transformation approaches between user requirements and analysis models," *Require. Eng*, vol. 16, no. 2, pp. 75–99, 2011.
- [37] R. Mokhtar, A. A. Rahman and S.H. Othman, "Towards model driven architecture in academic quality assurance information system development," *Journal of Telecommunication, Electronic and Computer Engineering*, vol. 9, no. 1-3, pp. 95–100, 2017.
- [38] D. Cetinkaya and A. Verbraeck, "Metamodeling and model transformations in modeling and simulation," *Proceedings of the Winter Simulation Conference (WSC 2011)*, United States, December 2011, pp. 3043–3053.
- [39] L. Krawczyk, and E. Kamsties, "Hardware models for automated partitioning and mapping in multi-core systems using mathematical algorithms," *International Journal of Computing*, vol. 13, no. 4, pp. 340-347, 2014.
- [40] N. Kharmoum, S. Ziti, Y. Rhazali, K. Elbouchti and R. W. Abdelaziz, "Analytical study of requirements models construction and their transformations in mda approach," *Proceedings of the 5th JDSIRT Conference on Information*

*Systems, Networks and telecommunications JDSIRT, Meknes, Morocco, 2018, pp. 12.*

- [41] N. Kharmoum, K. Elboucheti, S. Ziti and F. Omary, "Descriptive analysis of business value models' transformation in mda approach," *Proceedings of the 3rd International Conference Scientific Days in Applied Sciences JSSA'19, Larache, Morocco, 2019, pp. 63.*



**Nassim Kharmoum** received his Master Degree in Engineering Information Systems from the Cadi Ayyad University Marrakech, Morocco in 2010. Currently, he is a Software Engineering and Development Officer at the National Center for Scientific and Technical Research in Morocco. He is a member of the IPPS (Intelligent Processing Systems & Security) team, and he prepares his Ph.D. degree at the Faculty of Sciences, Mohammed V University in Rabat, Morocco. His research interest focuses on analysis and conceptual modeling, software requirements, and modernization of legacy systems.



**Prof. Dr. Soumia Ziti**, Ph.D. in Computer Science is a habilitated professor in the faculty of sciences in Rabat. Her research domains include Special Classes of Graphs: Structural and Algorithmic Aspects. Her applications cover Information System Engineering, Information System Mode-

ling, Meta-modeling, Big Data, Internet of things, Data, Graphs, metaheuristics, etc. Professor Soumia is an author and a co-author of several papers published in international journals and conferences.



**Prof. Dr. Yassine Rhazali**, Ph.D. in Computer Science, he obtained his doctorate in software engineering, he proposed validated approaches in about thirty scientific articles published in scientific conferences and in scientific journals. He is a member of international committees in several conferences and scientific journals. He is a professor at the Moulay Ismail University, Meknes, Morocco. His area of research is model-driven engineering and big data.



**Prof. Dr. Fouzia Omary** obtained her first Ph.D. in 1988 in Computer Sciences at Mohammed V University in Rabat and the second in 2006 in Security, namely Cryptography in the same University. Dr Fouzia Omary is a Professor in the Computer Sciences Department of the Faculty of sciences in Mohammed V University in Rabat since 1984. She is Director of the Computer Sciences Research Laboratory (LRI) from 2012 to 2016. Now she is Director of the Intelligent Processing Systems & Security team since 2016. Her research interests include cryptography, information security and information system.