Transformation of the system sequence diagram to an interface navigation diagram

William Germain DIMBISOA
PhD Student
Laboratory of Computer Science and Mathematics Applied to Development (LIMAD),
University of Fianarantsoa,
Fianarantsoa, Madagascar
dwilliamger@gmail.com

Thomas MAHATODY
PhD
Laboratory of Computer Science and Mathematics Applied to Development (LIMAD)
University of Fianarantsoa,
Fianarantsoa, Madagascar
hasinathomas@hotmail.com

Josvah Paul RAZAFIMANDIMBY
Full Professor
Laboratory of Computer Science and Mathematics Applied to Development (LIMAD),
University of Fianarantsoa,
Fianarantsoa, Madagascar
jp_josvah@yahoo.com

ABSTRACT

The automatic generation of the computer systems such as ergonomic Human Computer Interactions (HCI) always fascinates a computer project designers. For example, the MACAO (Méthode d'Analyse et de Conception d'Applications Orientées-objets) method, which is a participatory method, allows the development of complex object-oriented applications in accordance with the fundamental principles of software engineering. It is based on four models: organizational, structural, behavioral and HCI models. Within HCI models, MACAO offers two types of models: the first model is the Interactive Navigation Diagram (IND) which allows to design and to model the logic of sequence of the interactions between the user and the software; the second model is the Logical Model of HCI which is a translation of the Interactive Navigation Diagram in a particular technology (Windows, WEB, Multimodal ...). In this paper, we perform automatic generation of the Interactive Navigation Diagram from the System Sequence Diagram noted SSD. We use an architecture conforming to the Model Driven Architecture approach (MDA). This study aims at the automatic creation of new models from already existing models. We adopted the Model Driven Engineering (MDE) Approach. Thus, transformation rules are created to automatically generate models.

Author Keywords

Human Computer Interaction, Interface Navigation Diagram, Model Driven Architecture, Model Driven Engineering, transformation, System Sequence Diagram.

INTRODUCTION

The objectives of MACAO method are firstly help in the discovery of all the information necessary for the design and development of a software, secondly design the overall and detailed structure of the software in terms of object classes.

This method also define the HCI which is best suited to the needs of users. MACAO Optimize too the maintenance of the delivered software.

It allows to develop reliable, scalable and time-based object-oriented programs. MACAO relies on four-step approach such as: comprehensive analysis, overall design, development and finalization. The four types of models adapted to each area concerned by the software: organizational, structural, behavioral and HCI models. A typical documentation allowing to describe the software in different aspects: technical, maintenance, user and contractual.

In the second step, the method advocates designing the architecture of the software on several levels: functions, business classes, HCI general structure, cutting into prototypes. During HCI general structure design, MACAO uses Interactive Navigation Diagram model.

It is possible to automatically generate this Interactive Navigation Diagram [5] from the sequence diagram. This will facilitate its tasks and minimize the time and load of designing and developing a software. This paper aims to help computer scientist minimize the cost of software development. This article is able to simplify the task of designer while respecting the delay of the project realization.

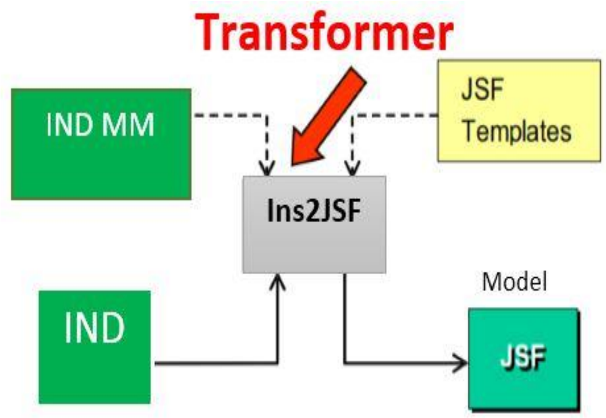
We will use the MDE approach for the automatic generation of the sequence diagram to the Interactive Navigation Diagram by creating transformation rules. The sequence diagram is the best diagram to highlight the interaction between user and the system. This study is structured as follows: in the first we will talk about the MACAO approach, the MDE approach [6] and its tools. Then we will automatically transform and create the Interactive Navigation Diagram from the sequence diagram. Finally, we will analyze the transformation model from the sequence model to the Interactive Navigation Diagram model. We will test our tool.

MACAO AND MDE

In this section, we briefly present the MACAO method and the MDE approach.

Approach by MACAO

MACAO approach [12] is equipped with a graphic editor of the Interactive Navigation Diagram called VisualSNI who is allowing to manipulate these models. MACAO also allows to transform this Interactive Navigation Diagram model in order to generate semi-automatically a specific model of the HCI. Nicolas Ferry has presented in his research work [12] about the transformation of Interactive Navigation Diagram into Java Server Faces (JSF). The Figure 1 shows this transformation.



MM: Meta model

IND: Interactive Navigation Diagram

Figure 1. A template-based transformer requires the IND metamodel (IND-MM), and a set of templates of transformation in the JSF target technology. By passing a source IND model, we get a mockup with the HCI generated in the form of web pages [12] (modified).

MACAO has informed a participatory approach from the beginning of the development to the finalization of the software and its maintenance. This method has taken over widely in the design phase the Unified Modeling Language (UML) [7] notation and its diagrams. It has participated in the modeling and the mechanism of the transformation of the models into the MDE [16]. MACAO takes into account the risk factor throughout the software manufacturing process and proposes a criterion for evaluating workloads for the software development [12]. The Figure 2 shows the architecture of the HCI axis for the MACAO approach presented by Nicolas Ferry.

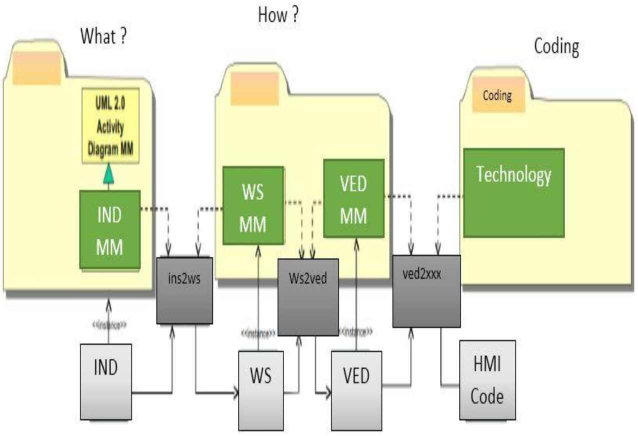


Figure 2. Overview Architecture of the HCI axis with transformers for the passage of diagrams of different levels. Here MM means Metamodel. This diagram shows the design of a HMI from different levels models of abstraction [12] (modified).

IND: Interaction Navigation Diagram

WS: Windows Scheme

VED: Visual Element Definition

MM: Meta model

In this Figure 2, to answer the WHAT question, the designer uses the Interactive Navigation Diagram model.

The Interactive Navigation Diagram makes it possible to design, model and represent the sequence of a dialogue between the software and the user taking into account the supposed behavior of the user. It is a conceptual model with a high level of abstraction to represent the user's needs and user's functional requirements. The user will be able to navigate between the information offered to him in a seemingly free way, but being constrained to a logic imposed by the designer of the application [12]. The Interactive Navigation Diagram has two types of dialogue unit:

- Basic or elementary dialogue unit;
- Composed dialogue unit by juxtaposition

The Figure 3 shows the structure of the Interactive Navigation Diagram model, the elementary dialogue unit identified by its name.

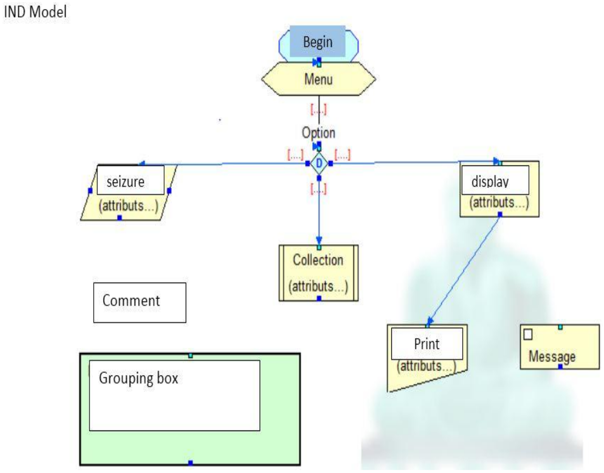


Figure 3. Structure of the IND model, dialogue unit.

MDE approach

The Model Driven Engineering approach [6] is the field of computer science that provides tools, concepts and languages to create and automatically transform a models. The architecture directed by the Model Driven Architecture models [10] is a software development approach proposed and supported by the Object Management Group (OMG) which is normalize all of approach. This approach is a particular variant of the MDE. It is structured in three levels. The transition from one model [15] to an another model is ideally carried out automatically. MDA [9] is an approach to implementing an application.

It is also proposed and supported by the OMG. The basic principle of the MDA approach [9, 5] is the development of platform-independent requirement a models and the transformation of these first models into platform dependent models. The first level corresponds to the conceptual model was called the Computational Independent Model (CIM). This model is completely independent of the techniques and the platforms.

It captures the needs and the requirements of users. The second level corresponds to the logical and technical model who is called Platform Independent Model (PIM). It is platform independent and obtained from transforming the CIM level model using transformation rules. The third level is called Platform Specific Model (PSM). It is explicitly the result of the transformation of the PIM model. It represents the final application adapted to a specific platform.

GETTING INTERACTIVE NAVIGATION DIAGRAM (IND) FROM SEQUENCE SYSTEM DIAGRAM (SSD)

The creation of sequence diagram [13] is a common and mandatory activity in the design and the development of the software. We aim to reduce the load of carrying out a computer project. The main objective of this article is to automatically transform the sequence diagram to IND.

The approach [12] who is indicated by Nicolas Ferry therefore requires additional tasks for a computer designers and a developers in particular because of:

- semi-automatic transformation of models: existence of human intervention,
- modeling was begun in the Interaction Navigation Diagram model: the trend of the designer is the orientation towards the use of diagrams of the UML notation to model the user interface,
- notion of correspondence between the Interactive Navigation Diagram model and the UML diagram.

In this approach, the transformation of the Interactive Navigation Diagram into an interface model, that is to say the transformation of the PIM level into PSM, has already been carried out by Nicolas Ferry. So this transformation was not taken into account. And, we start designing and modeling directly through the development of the Interactive Navigation Diagram.

Synoptic diagram of the transformation of sequence diagram to IND

The manipulation of the Interactive Navigation Diagram model has required additional skills for computer scientists. This results in the loss of time during the realization of a computer project. We have developed appropriate rules to automatically generate models. To support the transformation of sequence diagram into Interactive Navigation Diagram, ATLAS Transformation Language (ATL) [4, 17] is used with its tools. The synoptic diagram of the approach adopted to conduct the research project is presented in the Figure 4.

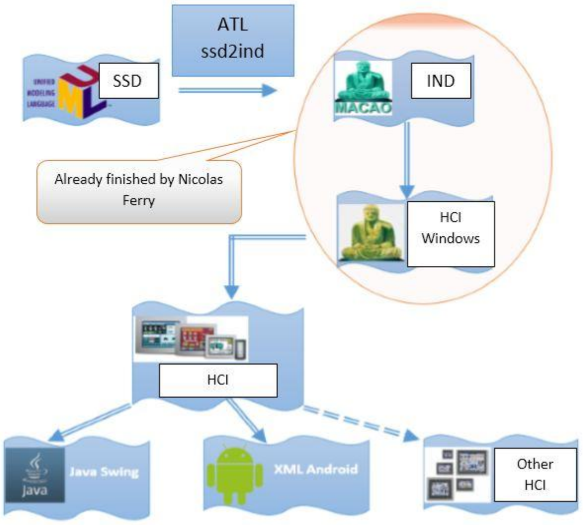


Figure 4. Synoptic diagram of the approach.

In this figure 4, we start the conception by creating SSD and generate the IND. The follow transformation is already finished by Nicolas Ferry to obtain Logical Model of HCI from IND.

ATL is a model transformation language inspired by Query View Transformation. It allows translation between XML Metadata Interchange (XMI) files. It is essential to have an initial and a target metamodels, as well as a mapping file. In this synoptic diagram of the approach, the modeling of an application was begun with the development of the sequence diagram which will be automatically generated in Interactive Navigation Diagram. This Interactive Navigation Diagram will be transformed into an HCI model, and this transformation is already carried out by Nicolas Ferry. In order to minimize the designer's workload, we have created the ***ssd2ind.atl*** tool to automatically generate the requirements model in logical models of the user interface model.

Transformation with ATL based on "ecore"

The transition from one model level to an another must follow the transformation rule. These models must be expressed in a modeling language who is defined by a metamodel. A transformation takes as input one or more models of a left part called Left-Hand Side. And it outputs one or more models of a right part called Right-Hand Side. The Left-Hand Side expresses access to the source models, and the Right-Hand Side indicates the expansions so the creation, updating of the target models. Each transformation takes one or more input models and produces one or more output models until obtainable an executable artefacts [6]. An overall view of the model transformation is presented in the Figure 5.

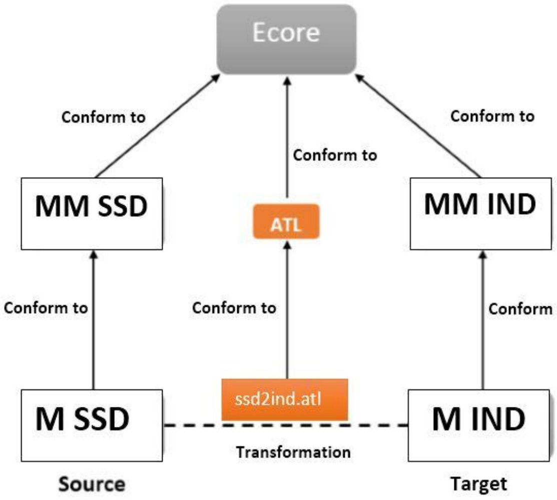


Figure 5. Overview of ATL transformation of sequence system diagram to IND.

M: Models, **MM**: Metamodels, **MMM**: Metametamodels

ATL is part of the Eclipse Model-to-Model project. It consists of a transformation language, a compiler with a virtual machine and a tool based on Eclipse.
In the Figure 2, we have the source model conforming to a source metamodel. Modeling is based on the metametamodel "ecore". The development of the sequence diagram metamodel begins the transformation. Compliant with this first metamodel, we will create the sequence diagram model as a source model. We have created transformation rules built into ***ssd2ind.atl*** to automatically generate the INS model as a target model. To manipulate the "ecore" metametamodel, the Eclipse Modeling Framework (EMF) is used to process models. This Framework lets us to store templates as files to ensure the persistence. It allows to process the different types of files: conforming to the XMI standard and also in specific forms (Java code).

Generation of sequence system diagram to Interactive Navigation Diagram

In this study, we need to use the research work [12] of Nicolas Ferry, but instead of designing the IND model, we will automatically generate it from a sequence diagram model. The process of designing and developing software is generally required the creation of the sequence diagram. This study creates a tool for automatically transforming sequence diagram to IND. It highlights the design of the elements of an interaction. To implement the transformation, we will firstly create the sequence diagram metamodel as a source metamodel. The Figure 6 shows a part of the sequence diagram metamodel.

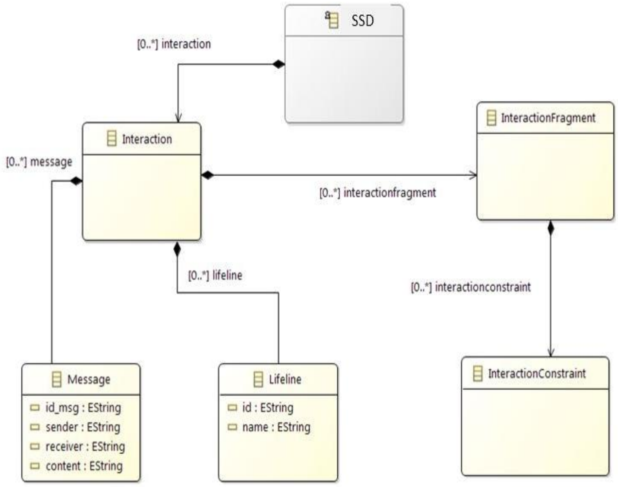


Figure 6. Part of the sequence system diagram metamodel.

This sequence diagram metamodel serves to create a model that is an instance of this metamodel. Secondly, we will create the Interactive Navigation Diagram metamodel as a target metamodel.

The Interactive Navigation Diagram allows to model and to conceive the logic of sequence of the interactions between the user and the software [6]. It is a conceptual model with a high level of abstraction to represent the needs and functional requirements of users. Interactive Navigation Diagram allows to model the sequence of interaction between the software its user.

Its main objectives are to model the user interface in terms of navigation between the various basic elements, access rights and functional coverage related to the HCI [12]. The Figure 7 shows a part of the Interactive Navigation Diagram metamodel.

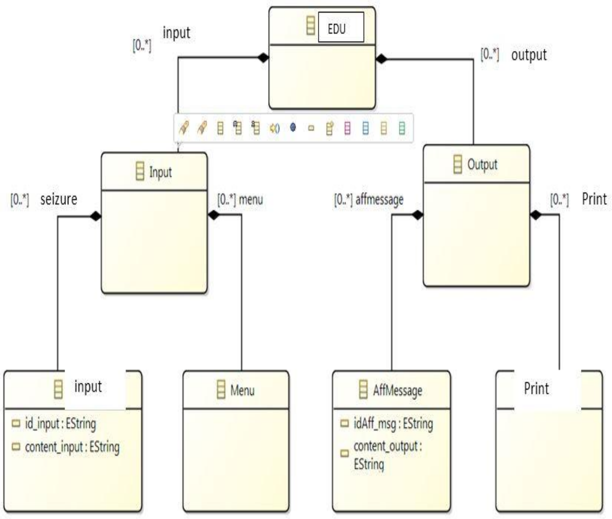


Figure 7. Part of the Interactive Navigation Diagram metamodel.

This IND metamodel serves to generate a model which is thus an instance of this metamodel. We will create in the end the sequence diagram model which is an instance of the metamodel of itself. And this model will be automatically transformed into an IND model. Appropriate rules are created to support transformation. The Figure 8 shows the source model called the System Sequence Diagram.

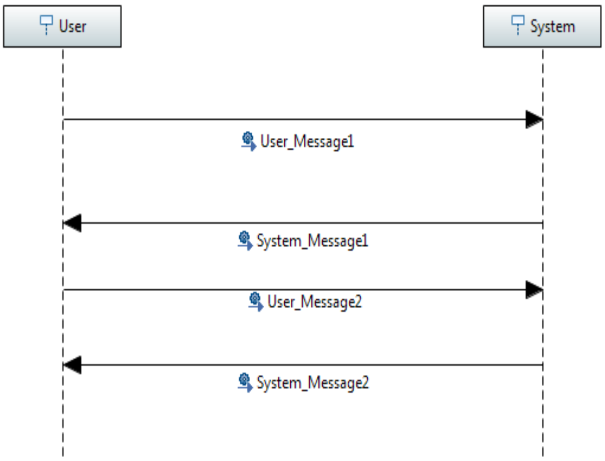


Figure 8. Model of the System Sequence Diagram.

The Figure 9 shows the target model called the Interaction Navigation Diagram.

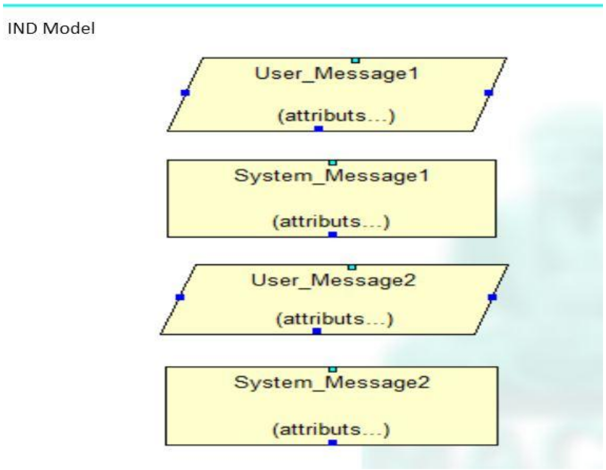


Figure 9. Model of the Interaction Navigation Diagram.

From the point of view of the HCI, an interaction takes place either as an *entry in the system* or by *sending to the user*. Dialog box categories are represented with their properties: Display, Collection display, Input, Menu ... [12]. An interaction is mainly represented in two ways:

- Either a message *from the user and directed to the System*: this message is transformed into a dialogue unit called *Input* in IND;
- A message *from the system to the user*: this message is transformed into a dialogue unit called *Output* or AffMessage in IND.

The representation in the Figure 8 and the Figure 9 implements the correspondence between the interaction message of the sequence diagram model and the dialogue unit of the IND model.

DISCUSSION

The purpose of modeling an interactive application is to allow software developers to define the models that they use for an application. We obtain the Interaction Navigation Diagram model as a result of this study. This model is automatically generated from the sequence system diagram to minimize loads and facilitate the tasks of interactive application designers. The purpose of this article is to save time by automatically creating the Interaction Navigation Diagram from the sequence system diagram. This sequence system diagram is not enough to design an user interface, but to highlight the interaction between the user and the system it is interacting with, it is better to first build the sequence diagram.

The modeling of HCI according to the architecture proposed by Nicolas Ferry was begun directly with the development of the Interaction Navigation Diagram. This study encourages the start of design by creating the sequence system diagram to capture the needs of users. On the one hand, the modeling of HCI can be started by directly developing the Interaction Navigation Diagram model using the VisualSNI tool. But, the direct creation of the Interaction Navigation Diagram was required additional tasks for the computer scientists. Then, we solicit to automatically generate the Interaction Navigation Diagram model from the sequence diagram model. The aim of this study is to encourage the designer and the software developer to realize evolving programs on time. We are used the UML [10] notation to model the development. The Interaction Navigation Diagram model have many advantageous interaction elements for HCI.

Nicolas Ferry has presented only a tool for transforming the CIM level to the PIM level, in this case the transformation of Interaction Navigation Diagram into an interface model. For the transformation of the sequence system diagram to Interaction Navigation Diagram, the elements of the Interaction Navigation Diagram are no longer specified to the interactions validated from the keyboard or mouse. For the automatic generation of the Interaction Navigation Diagram from the sequence system diagram by the ATL tool, we used the XMI format of the Papyrus, VisualSNI. There are a lot of Framework but we adopt EMF [2] for usefull especially in model transformation in Eclipse development environment.

There are a lot of similar approach such as approach [1] by Arnaud Brossard based on PERsonalization and COnceptual MOdeling Method (PERCOMOM) [3] and approach based on MACAO.

The table 1 shows the comparison between the similar approaches.

Approach	Transformation rules	Tools	Transformation
[1] based on PERCOMOM	Does not having rules	Insufficient tools	Semi-automatic
[12] based on MACAO	Having rules	Insufficient tools	Semi-automatic
Proposed approach	Having rules	Use of created tools	Automatic

Table 1. Comparison of the similar approaches.

Thus, this article offers the following opportunities:

- Time saving: our study makes it possible to obtain Interaction Navigation Diagram immediately from sequence system diagram, it also makes it possible to minimize the costs of carrying out a computer project;
- Reusability: Transformation rules are customizable and templates are reusable, the generated model is specific for a platform;

We have tested our tool to realize the transformation.

CONCLUSION

The specific architecture of this study conforms to the MDA approach. It allows to automatically generate target models from existing conceptual models. The research strategy is based on the design, the modeling and the realization of a Logical Model of HCI. We have defined a HCI more adapted to the needs of the users by using approach which based on MACAO method. The approach makes it possible to simplify as much as possible the modelization of an application. The main objective is to solicit the designer to simplify the process of the realization of computer project. The approach consists in helping computer scientist to facilitate tasks, reduce and minimize the time required to complete an application. This study allows to automatically generate Interaction Navigation Diagram from the sequence diagram to facilitate the tasks of the HCI designers. In this way, it reduces loads and saves the time needed to produce an application. On this study, we will consider the retro engineering in the future work.

ACKNOWLEDGMENTS

I would like to express my sincere thanks to all those who helped us in designing and carrying out this paper.

REFERENCES

1. Arnaud, B., Mourad, A. and Christophe, K. Context Awareness and Model Driven Engineering: A Multi-Level Approach for the Development of Interactive Applications in Public Transportation, 2008.
2. Cédric, B. and Alfonso, P. Model Differences in the Eclipse Modelling Framework, 2008.
3. Firas, B. and Mourad, A. Tool support for conceptual models of the PERCOMOM platform applied to transportation domain.
4. Frédéric, J., Freddy, A., Jean, B. and Ivan K. ATL: A model transformation tool, 2017.
5. Igor, S., J., V. Introduction to Model Driven Architecture (MDA), 2007.
6. Jean, B. In Search of a Basic Pronciple for Model Driven Engineering, 2004.
7. Jean, B., C. and Nicolas, F. Modélisation PIM des IHM : VisualSNI, 2007.
8. Jean, B., C. and Nicolas, F. SNI: High Level Model for HMI Design and Mock-up, 2008.
9. Jean, B., From Object Composition to Model Transformation with the MDA, 2001.
10. Kuznetsov, M. B. UML Model Transformation and Its Application to MDA Technology, 2007.
11. Li, D. and Li, D. An Approach to Formalize UML Sequence Diagrams in CSP, 2012.
12. Nicolas, F. Formalisation des modèles de la méthode MACAO et réalisation d'un outil de génie logiciel pour la création d'interfaces homme-machine, 2008.
13. Sawprakhon, P. and Limpiyakorn, Y. Sequence Diagram Generation with Model Transformation Technology, 2014.
14. Tisi, M., Martinez, S. and Choura, H. Parallel Execution of ATL Transformation Rules, 2013.
15. Tom, M., A Taxonomy of Model Transformation, 2006.
16. Wided, B., Adel, M., Lassaad, B., Sirine, R. and Mourad, A. An MDE Approach for User Interface Adaptation to the Context of Use. HCSE 2010, LNCS 6409, pp. 62–78, 2010.
17. Yassine, R., Youssef, H. and Abdelaziz, M. Model Transformation with ATL into MDA from CIM to PIM Structured through MVC, 2016.