# An Experimental Usability Evaluation Framework for Model-Driven Tools

Jose Ignacio Panach[1], Nelly Condori-Fernández [2], Arthur Baars[1], Tanja Vos[1], Ignacio Romeu[1], Óscar Pastor[1]

[1]Centro de Investigación en Métodos de Producción de Software ProS.
Universitat Politècnica de València
Camino de Vera s/n, 46022 Valencia, Spain
{jpanach, abaars, tvos, iromeu, opastor}@pros.upv.es
[2]Faculty of Electrical Engineering, Mathematics and Computer Science
University of Twente
Information Systems Group
7500 AE Enschede, P.O. Box 217, The Netherlands
nelly@pros.upv.es

**Abstract.** According to the Model-Driven Development (MDD) paradigm, analysts can substantially improve the software development process concentrating their efforts on a conceptual model. This conceptual model can be transformed into code by means of transformation rules applied by a model compiler. MDD tools are very useful to draw conceptual models and to automate the code generation. Even though this would bring many benefits, wide adoption of MDD tools is not yet a reality. Various research activities are being undertaken to find why and to provide the required solutions. However, insufficient research has been done on a key factor for the acceptance of MDD tools: usability. With the help of end-users, this paper presents a framework to evaluate the usability of MDD tools. The framework will be used as a basis for a family of experiments to get clear insights into the barriers to usability that prevent MDD tools from being widely adopted in industry.

**Keywords**: Usability, model-driven development, evaluation framework

## 1 Introduction

In a Model-Driven Development (MDD) process, analysts focus their efforts on building a conceptual model that is transformed into code automatically or semi-automatically (depending on the model compiler capacity). MDD tools can offer many benefits for a software developing company: reduced costs, reduced development time, higher quality, higher customer satisfaction, and, hence, improved competiveness [19]. However, the wide acceptance of MDD is not yet a reality. Introducing MDD methods and tools in a project is not simple and also requires serious changes in the organization's culture and processes. To address these issues, various research initiatives have been undertaken. MDD Maturity Models have been designed to establish capability levels towards the progressive adoption of MDD within an organization [17]. Case studies have been performed to find out which conditions should ideally be fulfilled by companies in order to successfully adopt MDD tools in their organizations [21]. Examples of these conditions are: the learning curve, the tool maturity, and the resistance to change. However, insufficient research has been done on a key factor that is fundamental for the adoption of MDD tools: usability. Both consumers and technology companies have accepted that if a product is easy to use, it sells more and is adopted quicker. For MDD tool adoption, usability is even more important since, on the one hand, they are difficult to use due to the complexity of the paradigm, and on the other hand, they are highly interactive applications offering a large number of different kinds of functionalities [15].

The main contribution of this paper is to present an empirical framework to perform a set of usability evaluations in MDD tools. There are several advantages of working with a framework. Firstly, it is very easy to replicate an experiment with an existing framework. We hope to set the basis for a family of experiments as advocated by Basili [2], since it is difficult to measure all the involved variables in the same experiment. The target of all the experiments should be to study the usability of MDD tools and provide clear insights into the barriers to usability that might prevent MDD tools from being widely accepted in industry. Another advantage of using a framework is that it helps to know clearly the required elements and the stages to perform the experiment. As a proof of concept, we carried out the evaluation of an MDD tool called OLIVANOVA [3], an industrial tool that implements a MDD software development method called OO-Method [13]. This tool has been selected due to its high profile in the context of conceptual model-based code generation. OLIVANOVA is an industrial tool that is capable of generating complete functional systems automatically from a conceptual model. This feature contributes to the validation of the usability evaluation framework beyond an academic context.

The structure of the paper is as follows. 2nd section introduces related works. In 3rd section, we describe our proposed framework to evaluate usability in MDD tools. In 4th section, we apply the framework to a specific MDD tool called OLIVANOVA. Finally, 5th section presents the conclusions of this work.

## 2 State of the Art

Several authors have proposed frameworks for measuring system usability, since the Human Computer Interaction (HCI) community insists on the importance of a framework for evaluating usability. Several authors such as Fiora [4], Kostiainen [7], Masemola [8], and Andre [1] have proposed performing usability evaluations by means of a framework. Fiora has defined an evaluation framework to evaluate the system usability automatically. Kostiainen has designed a framework to evaluate the usability in distributed applications. Masemola has also defined a framework focused on the usability evaluation of e-learning applications. Finally, Andre has designed a framework taking as input structured knowledge based on usability concepts. All these proposals have the same disadvantage: they are not dealing with features specific of MDD tools. Frameworks that aim to measure the usability in any system can be applied to MDD tools, but the results of these experiments are not precise.

Little work has been published about the usability of MDD tools, but if we extend our research to Computer Aided Software Engineering (CASE) tools in general, we find several usability evaluations. For example Senapathi [20] focuses mainly on the learnability of UML CASE tools in an educational environment. A similar work has been done by Post [16], who describes a survey with questionnaires letting respondents rate the importance of attributes and categories of characteristics of OO CASE tools. Moreover, Philips [14] has described a framework for usability evaluations of OO CASE tools. The framework consists of a hierarchy of usability criteria similar to those of Nielsen [10], focusing on the typical properties of OO case tools. Another work that has been developed in the context of CASE tools has been developed by Seffah [18]. Seffah's proposal aims to reduce the conceptual gap between the developer's mental model of the integrated software development environment and the way it can be used.

In contrast to conventional CASE tools that are oriented to software development based on design and programming, MDD tools have to cope with specific features where the modeling and the programming perspective become intertwined. Moreover, existing works in the context of CASE tools do not emphasize the design of a framework with the aim of replicating the evaluation for several tools. Studying related works, we conclude that more work must be done on the usability evaluation of MDD tools. We state that for any MDD tool, usability is an essential key in becoming a tool that is fully accepted in industrial environments.

## 3 Empirical Framework to Evaluate the Usability of MDD Tools

In order to replicate the usability evaluation of MDD tools under similar circumstances, and to facilitate knowledge building through families of experiments, it is important to define a framework to describe the experimental evaluation. The framework describes how to carry out an empirical evaluation involving laboratory observations of users interacting with the tool on a set of predefined tasks. This framework has been designed using the proposals of the empirical software engineering community [23]. This empirical framework is composed of: a usability evaluation model and an experimental process.

### 3.1 A Usability Evaluation Model

We have designed a usability evaluation model, which identifies the most relevant elements for evaluating the usability of MDD tools. These elements (concepts) and their respective relations are represented as a UML class diagram in Figure 1. According to Figure 1, we aim to evaluate the usability of MDD tools by means of: satisfaction, efficiency and effectiveness, such as ISO 9241-11 proposes [6]. A *user* interacts with a particular *MDD tool* in a specific *environment* in order to achieve an interactive *modeling task*. This environment can be characterized by both *technical* and *physical* aspects. Technical aspects are related to the software and hardware used in conjunction with the tool. Physical aspects involve the area where the experiment is performed.

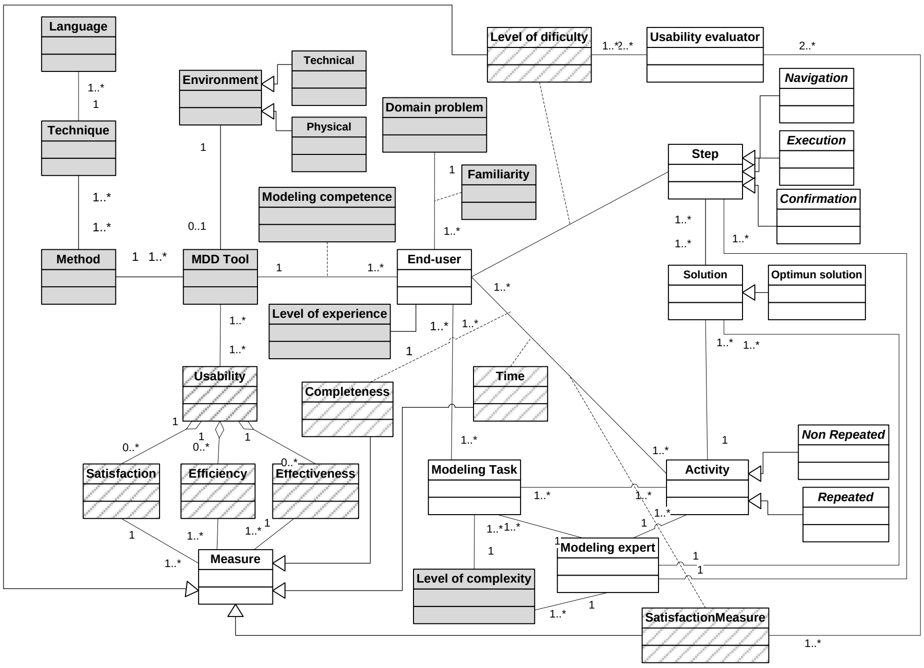


**Fig. 1.** Usability Evaluation Model

A MDD tool provides support to a specific MDD *method* that involves different *modeling techniques*[1] for representing the system using one or more languages. The user can have different *modeling competences* and *experience* in using a MDD tool. Moreover, each user is more *familiarized* with one *problem domain* than another. Since the usability is evaluated from a HCI perspective, a set of *modeling tasks* must be performed by the *end-user* (experimental subject). These tasks can be divided into *activities*, which are atomic operations inside the task. Since each modeling activity can be resolved by the user in different ways (*solution*), the expert in modeling must specify which solution is the most *optimal*. Each solution in turn is divided into *steps*, which are the actions that the end-user must follow to accomplish the activity. We identify three types of steps: (1) *Confirmation*: the end-user must confirm an execution; (2) *Execution*: the user triggers an action; (3) *Navigation*: the user navigates towards another context.

For each activity, we have to store a *satisfaction measure*, which can be extracted with different instruments. (i.e., questionnaires, tools based on emotional response, etc.). To measure efficiency, the end-user must be *timed* and the level of *completeness* of the activity must be known. With regard to effectiveness, we need to measure whether or not the end-user is performing the steps with difficulties (*level of difficulty*). To do this, we need two or more *usability evaluators* to measure the level of difficulty that each end-user has per step and the satisfaction per activity.

[1] The term method refers to a systematic way of working to obtain a desired result. The term technique refers to a recipe for obtaining a result. Methods contain techniques [22].

A *modeling expert* is responsible for defining tasks, activities and solutions. This expert is also responsible for classifying the tasks into three *levels of complexity*: *easy, medium,* and *difficult*. As some activities can be included in more than one task, we need to identify the repeated activities. This is because we will take into account the first time the task will be executed in order to avoid learning effects. The interpretation of the colored classes of the model is explained in a below section.

### 3.2 Experimental Process to Evaluate Usability

This section focuses on the steps that compose the process to perform the usability evaluation according to the usability model. There are four stages [23] (Figure 2)

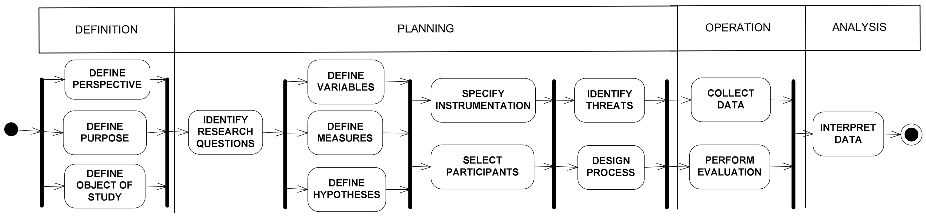


**Fig. 2.** Process to evaluate usability

**Definition**: The first step is to determine the foundation of the experiment. The goal is defined in terms of the *perspective*, *purpose*, and *object of study*. **Planning**: This step specifies how the experiment is conducted. First, the researcher specifies the *questions* to be answered with the outcomes of the experiment. Next, the researcher defines *variables, measures, and hypotheses*. Variables are divided into two types:

- **Independent variables**: Variables that can be changed and controlled in the experiment. In the usability evaluation model (Figure 1) these variables are represented as classes in grey background.
- **Dependent variables**: Variables affected by the independent variables. These variables are quantified by means of one or more measures. Dependent variables and measures are represented in Figure 1 by classes crossed by diagonal lines.

Measures are entities that specify how to measure variables. Hypotheses are statements derived from the research questions that must be accepted or rejected. The hypotheses can be defined by the combination of variables. Next, the researcher selects the *participants* and specifies the *instruments* needed to perform the usability test. Finally, it is important to design the evaluation *process* and to identify *threats*.

**Operation:** In this step, the researcher performs the experiment and collects the needed data. **Analysis**: In this step the researcher interprets the experiment data.

## 4 Applying the Usability Framework to a Specific MDD Tool

This section explains the usability evaluation that we performed with the proposed framework. The studied MDD tool was OLIVANOVA [3], an industrial tool that generates fully functional systems from a conceptual model. Next, with OLIVANOVA, we explain how the information of the usability evaluation model (Figure 1) is provided in each step of the experiment (Figure 2). We focus our study on effectiveness and efficiency, relegating the satisfaction to a future experiment.

### 4.1 Definition of the experimental evaluation

The objective of our empirical evaluation was to analyze the object model of the OLIVANOVA tool, with respect to effectiveness and efficiency. This evaluation was performed from the viewpoint of the software developer, in the context of object-oriented software development and different levels of experience with MDD tools.

### 4.2 Experimental Planning

**Definition of research questions**:

- **RQ1**: Is the users' efficiency the same for modeling tasks with different levels of complexity independently of their background in using MDD tools?
- **RQ2**: Is the users' effectiveness the same for modeling tasks with different levels of complexity independently of their background in using MDD tools?

**Identification of variables and measures:**

- **Dependent variables:** Efficiency and effectiveness with the following measures:
  - **Efficiency:** This was measured by task completion percentage in relation to the time spent to perform a task. This measure is related to a ratio scale.
  - **Effectiveness:** This is the level of completeness reached in every task. This variable was calculated by two measures: 1) the percentage of tasks carried out correctly. 2) the percentage of correctly performed activities that were carried out in an optimum[2] way. These measures are related to a ratio scale.
- **Independent variables:** The level of complexity of the tasks, the modeling competence, and the level of experience using MDD tools.

**Identification of Hypotheses:**

- **$H_{10}$:** When using the OLIVANOVA tool for modeling tasks with different levels of complexity, the efficiency is the same for the three groups of users.
- **$H_{20}$:** When using the OLIVANOVA tool for modeling tasks with different levels of complexity, the effectiveness is the same for the three groups of users.

**Selection of Participants.** We used three groups of users:

- **Type I (Experts)**: Experienced using the evaluated tool. This group was composed of researchers of the ProS center of the Technical University of Valencia.
- **Type II (Medium)**: Experienced using similar tools. This group was recruited from the regional Valencian public administration who are familiar with open source MDD tools like Moskitt [9].

---

[2] Optimum means performing the tasks correctly without any difficulty.

- **Type III (Novice)**: No experience with the tool nor with similar tools. This group was composed of engineers from the Technological Institute of Computer Science who are familiar with Object-Oriented concepts but not with any modeling tool.

Since it is important for the number of subjects in these three groups to be balanced, we used four users from each group. This step provides the information represented in the classes of the usability model *End-user*, *Level of experience* and *Modeling competence* (Figure 1).

**Experimental Instrumentation.** We used the following instruments:

- **A tool to record the user**: We used Noldus [12] to time the user and identify the completeness of each task.
- **A list of tasks**: Every participant was asked to carry out 18 tasks (classified into easy, medium and difficult) related to objects creation and manipulation.
- **A pre-test**: Every participant fulfilled a list of preliminary questions to classify her/him into one of three groups of participants.

These instruments and the guidelines used can be found at [5]. This step provides the information of the remaining classes in the usability evaluation model: *Modeling task*, *Activity*, *Solution*, *Step*, and all their inherited classes.

**Design process .** Figure 4 shows a summary of the process to evaluate the MDD tool usability. First, the subject filled in the pretest to be assigned to one group of subjects. Next, the subject tried to perform 18 tasks with the MDD tool being evaluated. The tasks had previously been divided into three groups of difficulty. All this information was stored in a database to be processed later.

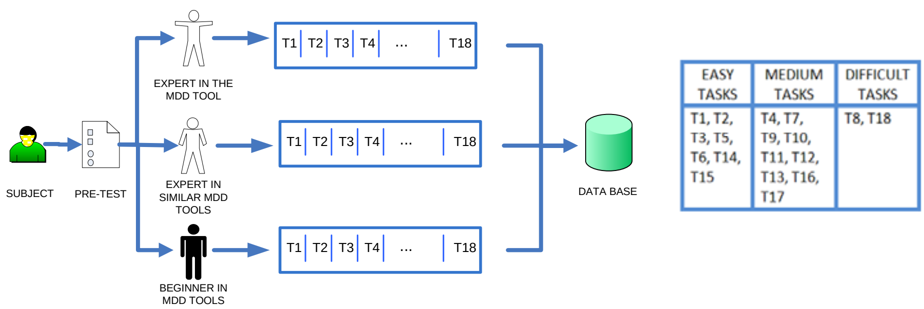


**Fig. 3.** Process of the Experiment

**Threats to validity.** Threats defined in the Wohlin's proposal [23] was minimized using a pre-questionnaire, limiting the time of the experiment, using a usability expert to coordinate the experiment and hiding the target of the experiment.

### 4.3 Analysis of the results

**Analyzing efficiency.** Efficiency was measured by task completion percentage in relation to the time spent doing a task. This time was calculated by summing the times necessary to complete each activity of the respective modeling task. The time required by the user to understand the task to be performed was not considered in the analysis. Figure 4a shows the results obtained for this variable.

According to a Kolmogorov-Smirnov test, efficiency follows a normal distribution. Since there is homogeneity of variances, ANOVA test is appropriate to verify the Hypothesis $H_{10}$. According to the results of the ANOVA test, we reject the null hypothesis $H_{10}$, which means that the efficiency using OLIVANOVA for modeling tasks with different levels of complexity is different for the three groups of users. However, if this analysis is carried out excluding the group of experts, there are no differences in the mean efficiency scores with the other two types of users (Types II and III). However, if we analyze only experts and medium users (Types I and II) a significant difference was found only for difficult tasks.

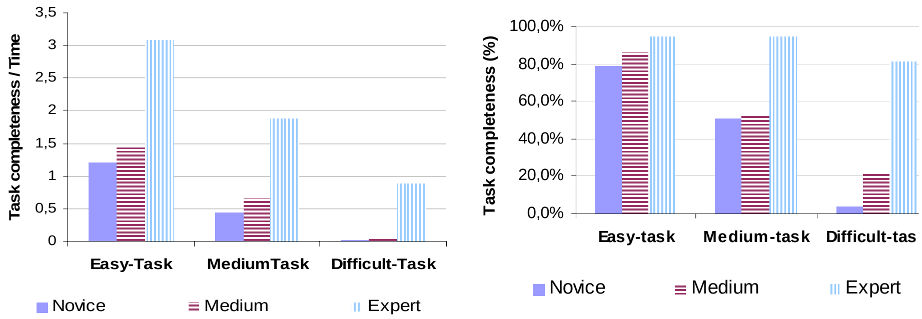


**Fig. 4.** a) Comparison of efficiency by type of user and task complexity level; b) Comparison of modeling task completeness by type of user and task complexity

**Analyzing effectiveness.** Effectiveness was measured in terms of modeling task completion percentage and percentage of correct tasks that were carried out optimally. Figure 4b shows the results obtained for the task completion percentage. Similar completeness percentage is only observed when the users performed tasks with an easy level of complexity. An ANOVA test was used (data normality and homogeneity of variances were corroborated) to know whether the type of user has an effect on overall completeness of tasks with different complexity levels. With this test, significant differences were found for both the tasks of medium level of difficulty and high level of difficulty. Studding tasks performed optimally, we noticed that there are also differences for the three groups of users. Therefore, we conclude that $H_{20}$ is not satisfied, i.e., the effectiveness is not the same for the three groups of users.

## 4.4 Problems Detected with the Usability Evaluation

Next, we detail usability problems of OLIVANOVA detected in the usability test. We have considered that an end-user has detected a usability problem with a task when the value of effectiveness or efficiency for this task could be improved. This information has been extracted from novice and medium users specially. Even though some expert users did not get good efficiency and effectiveness values for some tasks such as 7 and 12. We have classified each usability problem according to Nielsen' usability heuristics [11] (Table 1):

**Table 1.** Usability problems found by task

| Heuristic \ Task | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Guidance** | | x | x | x | | x | x | x | | | | x | | x | x | | | | x |
| **Workload** | | x | | | x | | | | | | x | | | | | | | x | |
| **User Control** | | | | x | | | | | | | | | | | | | | | |
| **Adaptability** | | x | | | | | | | x | | | | | | x | | | | |
| **Error management** | x | | | x | | x | x | x | | x | | | | | | | x | | |
| **Consistency** | x | | | | | | | | x | x | | | | x | | | | | |
| **Significance of code** | x | | | x | x | | | x | | | x | | | | | | | | |
| **Compatibility** | | | | | | | | | | | | | | | | | | | |

It can be seen from our test with OLIVANOVA that a violation of usability heuristics Guidance and Error Management have been the most detected. Moreover, the tasks for which effectiveness turned out to be less than expected (Tasks 2, 4, 6, 9, 14, 15 and 17, see the columns in light gray) suffer mostly from usability barriers like Guidance, Error Management, Consistency and Adaptability violations.

## 5 Conclusions and Future Work

The usability of MDD tools is a key factor for becoming completely accepted in industrial environments. This paper proposes an empirical framework to evaluate the usability of these tools. The framework aims to replicate the usability evaluation of MDD tools in similar conditions to increase the external and internal results. With the purpose of evaluating our framework; it was applied to study the usability of an industrial MDD tool called OLIVANOVA.

With respect to the usability test applied to OLIVANOVA, we have extracted some conclusions. Firstly, with regard to efficiency, we can state that there are no differences between medium users and novices. This fact means that OLIVANOVA does not share many features with other MDD tools. This is because OLIVANOVA has several stereotypes to extend the vocabulary of UML. Secondly, with regard to effectiveness, we can conclude that novice and medium users can only correctly perform easy tasks. Another important piece of data is that there are some common tasks where even experts had some difficulties. These tasks are related to the definition of formulas, inheritance specification, and class renaming. To solve these problems, we propose the following: improving the examples of formula definition in a wizard; allowing inheritance to be defined graphically; and allowing to edit the class name in the graphical representation of the class.

The results of the evaluation demonstrate that changes must be applied to OLIVANOVA in order to improve the users' effectiveness and efficiency. We are currently preparing a detailed internal report for the company that develops OLIVANOVA (CARE Technologies). As future work, we want to repeat our usability test with 30 users in order to obtain more significant values. In this new evaluation we will include the study of the user's satisfaction.

## Acknowledgments

This work has been developed with the support of MICINN, GVA and ITEA 2 call under the projects PROS-Req (TIN2010-19130-C02-02), ORCA (PROMETEO/2009/015) and UsiXML (20080026).

## References

1. Andre, T. S., Hartson, R., Belz, S. M., McCreary, F. The user action framework: a reliable foundation for usability engineering support tools. Int. J. Human- Computer Studies, vol. 54, (2001), 107-136.
2. Basili, V., Shull, F., Lanubile, F. Building knowledge through families of experiments. IEEE Transactions on Software Engineering. 25(4), (1999).
3. CARE Technologies S.A: www.care-t.com
4. Fiora T. W. Au, Baker, S., Warren, I., Dobbie, G.: Automated Usability Testing Framework. Proc. 9th Australasian User Interface Conference (AUIC2008) (2008).
5. Instruments: http://hci.dsic.upv.es/testOLIVANOVA/
6. ISO 9241-11 Ergonomic requirements for office work with visual display terminals - Part 11: Guidance on Usability (1998).
7. Kostiainen, K., Uzun, E.: Framework for Comparative Usability Testing of Distributed Applications. Security User Studies: Methodologies and Best Practices Workshop (2007).
8. Masemola, S.S.T., Villiers, M.R.R.D.: Towards a Framework for Usability Testing of Interactive e-Learning Applications in Cognitive Domains, Illustrated by a Case Study. Proceedings of SAICSIT 2006 (2006), 187 –197.
9. Moskitt http://www.moskitt.org/
10. Nielsen, J., Heuristic evaluation. Usability Inspection Methods. Wiley, New York, (1994), 25–62.
11. Nielsen, J.: Usability Engineering. Morgan Kaufmann (1993).
12. Noldus: www.noldus.com
13. Pastor, O., Molina, J.: Model-Driven Architecture in Practice. Springer, Valencia (2007).
14. Phillips, C., Mehandjiska, D., Griffin, D., Choi, M. D., Page, D. The usability component of a framework for the evaluation of OO case tools. Proceedings of Software Engineering, Education and Practice, IEEE, Dunedin, (1998).
15. Post, G., Kagan, A. User requirements for OO CASE tools., Information and software technology, Vol. 43, Nº 8, (2001), 509-517.
16. Post, G., Kagan, A. User requirements for OO CASE tools., Information and software technology, Vol. 43, Nº 8, (2001), 509-517.
17. Rios, E., Bozheva, T., Bediaga, A., Guilloreau, N. MDD Maturity Model: A Roadmap for Introducing Model-Driven Development. ECMDA-FA, LNCS 4066, (2006),78 – 89.
18. Seffah, A., Rilling, J. Investigating the Relationship between Usability and Conceptual Gaps for Human- Centric CASE Tools, IEEE Symposium on Human- Centric Computing Languages and Environments, Stresa, Italy, (2001).
19. Selic, B.: The Pragmatics of Model-Driven Development. IEEE software 20 (2003) 19-25
20. Senapathi, M.: A Framework for the Evaluation of CASE Tool Learnability in Educational Environments. Journal of Information Technology Education 4 (2005), 61-84.
21. Staron, M., Adopting MDD in Industry - A Case Study at Two Companies. Proceedings of the 9th ACM/IEEE International Conference on Model Driven Engineering Languages and Systems MoDELS 2006, Vol 4199, (2006), 57-72.
22. Wieringa R.J., Requirements Engineering: frameworks for understanding,Wiley, (1996).
23. Wohlin, C., Runeson, P., Höst, M. Experimentation in Software Engineering: An Introduction, Springer, (1999).