

An Empirical Approach for evaluating the Usability of Model-Driven Tools

Nelly Condori-Fernández¹, Jose Ignacio Panach², Arthur Iwan Baars³, Tanja Vos³, Óscar Pastor³

¹Faculty of Electrical Engineering, Mathematics and Computer Science
University of Twente
Information Systems Group
7500 AE Enschede, P.O. Box 217, The Netherlands
n.condorifernandez@utwente.nl

²Escola Tècnica Superior d'Enginyeria, Departament d'Informàtica, Universitat de València
Avenida de la Universidad, s/n, 46100 Burjassot, Valencia, Spain
joigpana@uv.es

³Centro de Investigación en Métodos de Producción de Software ProS.
Universitat Politècnica de València
Camino de Vera s/n, 46022 Valencia, Spain
{abaars, tvos, opastor}@pros.upv.es

Abstract. MDD tools are very useful to draw conceptual models and to automate 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. To illustrate the applicability of our framework, we instantiated it for performing a usability evaluation of a tool named INTEGRANOVA. Furthermore, we compared the outcome of the study with another usability evaluation technique based on ergonomic criteria.

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

1 Introduction

The Model-Driven Development (MDD) paradigm aims to develop software by means of a conceptual model that abstractly represents the system to develop. This conceptual model is the input for a model compiler that generates the code implementing the system. Transformations from model to code can be done automatically or semi-automatically (depending on the model compiler's capabilities). Even though the paradigm is not restrictive, it tends to automate as many transformations as possible to improve the analyst's efficiency. MDD tools promise many benefits for a software developing company: reduced costs, reduced development time, higher quality, higher customer satisfaction, and, hence, improved competitiveness [27]. However, the wide acceptance of MDD is not yet a reality.

There are several works that have studied MDD disadvantages to justify the reason why MDD is not widely used [30]: the learning curve of MDD tools is high; there is no development standard; relations among models that compose the conceptual model are not clear; difficult transformations. All these disadvantages share the same problem: a lack of usability in the tool that supports the MDD method.

In general, developers of MDD tools do not take care of usability when developing tools, since these tools are going to be used by people with high level of knowledge in computer science. However, this is a mistake, since MDD tools have specific features that are not shared with others. For example, some tools use a Class Diagram to represent the data persistency while other tools use an Entity Relationship Diagram. Therefore, when an analyst starts to work with a MDD tool, she/he needs much time for learning the specific features of the tool for

being efficient [29]. Usability is a key characteristic to obtain a good acceptance of the software for users that do not work in the area of the software development, and this is not less so in the context of the software development. For MDD tool adoption, usability is even more important than in other tools, since, on the one hand, MDD tools 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 [22].

The step previous to improvement of the usability of MDD tools is to define a process that evaluates their usability. This paper presents a framework for evaluating usability of MDD tools with end-users. MDD tools are more than single utensils to draw models. These tools are the front-end of a software development method, and they should guide and help the analyst throughout the method [22]. By experimental framework we mean the definition of a process, the elements involved and how to aggregate these elements. The framework aims to perform evaluations unambiguously. We propose performing the usability evaluation in terms of efficiency, effectiveness and satisfaction, such as ISO 9241-11 proposes [11].

There are several advantages of working with an evaluation framework. First, the framework helps to replicate the study and set the basis for a family of experiments as advocated by Basili [5], 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. Second, the framework helps to know clearly the required elements and the stages to perform the experiment.

In this paper, as proof of concept of our proposed framework, we have chosen INTEGRANOVA [10] to evaluate its usability. INTEGRANOVA is an industrial tool that implements a MDD software development method called OO-Method [20]. This choice is based on the capacity of the tool to generate code from a conceptual model. INTEGRANOVA can generate fully functional interactive systems without writing a single line of code. This feature contributes to the validation of the usability evaluation framework beyond an academic context.

This paper is the continuation of previous work published in [19]. In this previous work we presented an initial version of our framework and a preliminary proof of concept measuring efficiency and effectiveness of a MDD tool. In this paper, the novelties with regard to our previous work are: we have extended the framework with an aggregation method; we have measured the end-user's satisfaction while they interact with the MDD tool; and we have compared the precision of the framework in detecting usability problems with regard to an evaluation based on ergonomic criteria.

The structure of the paper is as follows. Section 2 introduces related works. In the third section, we describe our proposed framework to evaluate usability in MDD tools. In the fourth section, we apply the framework to a specific MDD tool called INTEGRANOVA, by evaluating the usability in terms of efficiency, effectiveness and satisfaction within an experimental context. In the fifth section we compare a usability evaluation based on the proposed framework with an evaluation based on ergonomic criteria. Finally, section 6 presents the conclusions of this work.

2 State of the Art

Several authors have proposed experimental frameworks for measuring system usability, since the Human Computer Interaction (HCI) community insists on the importance of a framework for evaluating usability. Authors such as Fiora [8], Kostiainen [13], Masemola [14], Andre [3] and Al-Wabil [2] have proposed performing usability evaluations by means of an experimental framework. Fiora has defined a 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. Andre has designed a framework taking as input structured knowledge based on

usability concepts. Finally, Al-Wabil [2] proposes a framework for integrating usability evaluation methods by matching the method's capabilities and limitations with a classification of usability problems. All these proposals have the same disadvantage: they are not dealing with the specific features of MDD tools, such as the end-users' modeling competence and the method behind the tool. Experimental 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 on 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 [28] focuses mainly on the learnability of UML CASE tools in an educational environment. His studies involve questionnaires and observation of 150 students in the classroom. A similar work has been done by Post [22], who describes a survey with questionnaires letting respondents rate the importance of attributes and categories of characteristics of OO CASE tools. One of the conclusions of the study was that the usability characteristic "ease of use" is important. Moreover, Philips [21] has described a framework for usability evaluations of what he calls Object-Oriented (OO) case tools (e.g. MetaEdit+ [15], Rational Rose [23], etc.). The framework consists of a hierarchy of usability criteria similar to those of Nielsen [17], focusing on the typical properties and characteristics of OO case tools. Their usability evaluations consist of passing questionnaires to users containing a question for each criterion in the framework.

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. MDD tools have as peculiarity that behind the tool there is a method to develop software that must be easily understandable by the end-user. 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.

There are also several usability studies based on user observation experiments in the context of software development, just as the work by Seffah [26]. However, these studies concentrate on IDEs for Java and C++ programming. As far as authors know, currently, there are no observation experiments with end-users to evaluate the usability of MDD tools.

Studying all these 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 for becoming fully accepted in industrial environments. This is our main motivation for proposing a usability evaluation framework for MDD tools.

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 observation 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 [32]. This empirical framework is composed of: a usability evaluation model, an aggregation method and an experimental process.

3.1 A Usability Evaluation Model

The ISO 9241-11 standard defines usability as the "Extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use." [11]. Although there is another ISO standard for usability (ISO 9126-1 [12]) that is strongly related to software engineering, we define an evaluation model according to ISO 9241-11 because it focuses on tasks and environment factors that can be researched more easily in an experimental context. The usability evaluation model 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.

As shown in Figure 1, we aim to evaluate the usability of MDD tools, which is measured by means of three elements: satisfaction, efficiency and effectiveness such as ISO 9241-11 proposes. A user interacts with a particular *MDD tool* in a specific *environment* in order to achieve an interactive *modeling task*. The environment can be characterized by both *technical* and *physical* aspects. Technical aspects are related to the software and hardware that are used in conjunction with the tool. Physical aspects involve the area where the experiment is performed. For example, loud noise in the workplace may prevent the users from receiving feedback from the tool.

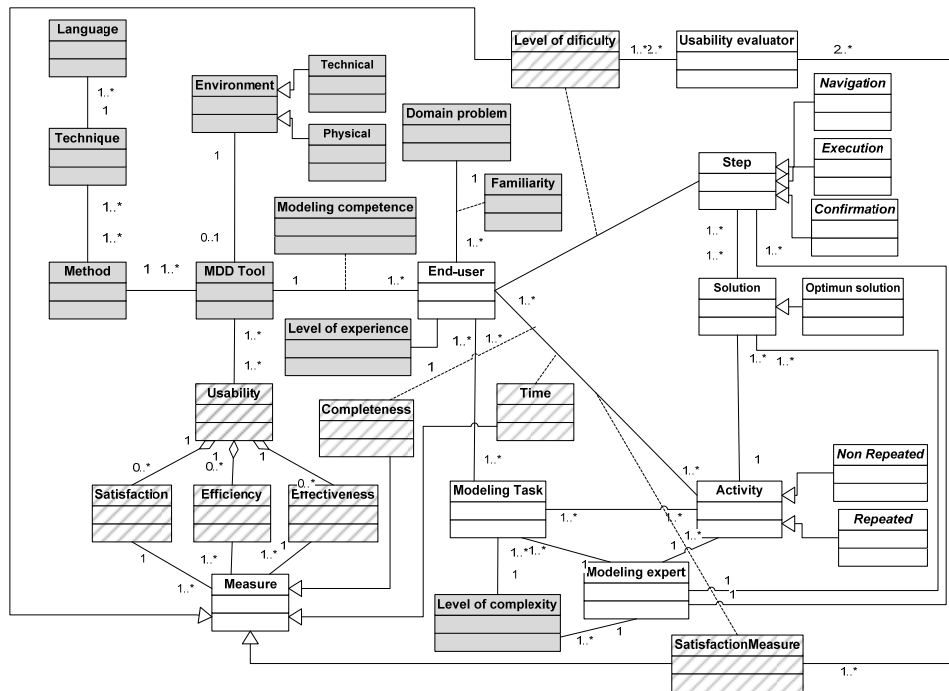


Fig. 1. Usability Evaluation Model

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. For example, although a user can be a beginner with ArgoUML tool [4] or Rational Rose [24], he may have a high capability of modeling on paper. Moreover, each user is more *familiarized* with one *problem domain* than another. For example, end-users that are familiarized with the development of banking systems are more comfortable modeling a spending application rather than a car renting system. Since the usability is evaluated from a human-computer interaction perspective, a set of modeling *tasks* must be performed by the *end-user* (experimental subject). These modeling 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. An optimum solution means performing the activity correctly with the minimum number of steps required, and without any difficulty. Tasks, activities, and steps are the entities on which the usability evaluation will be carried out. With regard to the steps, we have

¹ As the term method refers to a systematic way of working in order to obtain a desired result. As the term technique refers to a recipe for obtaining a certain result. Methods contain techniques [31].

identified three different types: (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.). The user's satisfaction is measured using this information. 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* for measuring the level of difficulty that each end-user has per step and the satisfaction measure per activity. 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*. We have classified the tasks into *easy*, *medium*, and *difficult*. As some activities can be included in more than one task, we need to identify the activities that are repeated. 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 Section 4.

3.2 An Aggregation Method

The evaluation of effectiveness is performed using two measures. One measure is by means of the completeness of the task and another measure is with respect to whether or not the user performs the steps of a task in an optimal way. To measure the latter, a level of difficulty is assigned to each step. This level is expressed as ordinal values: null (0), low (1), medium (2), and high (3). According to these values, we considered a step as optimum if the step was performed without any difficulty (null). Consequently, to determine this optimum value at the activities and tasks level, an aggregation method is required to aggregate the ordinal values assigned to each step. The method includes two phases as shown in Figure 2.

The first aggregation consists of dividing the number of steps performed by the users with the same difficulty level by the total number of steps that are involved in the activity to be aggregated. For example, in Figure 2, since the number of steps with a low level of difficulty (1) for performing the activity (A1,1) in task 1 (T1) is 3 and the total number of steps that are involved in this activity (A1,1) is 4, the ratio for this case is 3/4, and for the null level (0) is 1/4

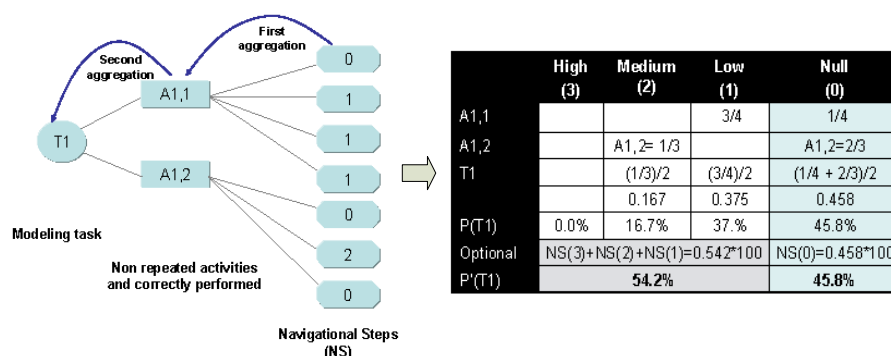


Fig. 2. Aggregating ordinal values: an example

The second aggregation consists of dividing the sum of ratios (proportions) obtained for each difficulty level by the number of activities that are involved in the task. For example, see in the third row of the table (Figure 2) the sum of ratios for the null level divided by the number of activities involved in the task 1 (two activities in our example). Finally, the proportions obtained for each level of difficulty are multiplied by 100 (see the fifth row of the table). The percentage for “null difficulty” represents the percentage of activities performed correctly with a high optimal level (this percentage in our example is 45.8%).

In Figure 2, we have grouped the values with high, medium, and low difficulty in order to compare the percentage of task performed without any difficulty with regard to the percentage of task performed with some difficulty. It is important to note that the evaluator can choose how to group the levels of difficulty. For example, in other cases it could be interesting grouping the percentages of null and low difficulty, and grouping the percentages of high and medium. This way we can compare the percentage of task performed with little difficulty with regard to the percentage of task performed with much difficulty.

3.3 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 [32] (Figure 3).

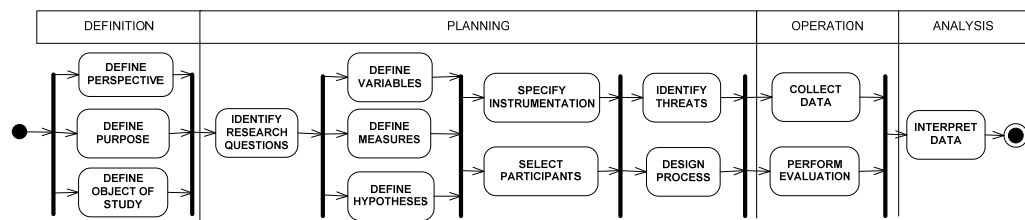


Fig. 3. Process to evaluate usability

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 evaluation. 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 INTEGRANOVA [10], an industrial tool that generates fully functional systems from a conceptual model. Next, with INTEGRANOVA, we explain how the information of the usability evaluation model (Figure 1) is provided in each step of the experiment (Figure 3).

4.1 Definition of the experimental evaluation

The objective of our empirical evaluation was to analyze the object model editor of the INTEGRANOVA tool, with respect to effectiveness, efficiency and satisfaction. 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?
- **RQ3:** Is the users' satisfaction 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, effectiveness and satisfaction 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 way*.
 - **Satisfaction:** This is defined as the extent to which users are free from discomfort, during the use of the product. Satisfaction can be measured by different ways such as subjective rating on scales, acceptability of the workload when carrying out different tasks, etc. In this paper, we measure the users' satisfaction in terms of their emotional response, by using the Emocards tool [1].
- **Independent variables:** Applying our usability evaluation model, the following independent variables were considered: the *level of complexity of the tasks*, which consists of three levels (low, medium and high) defined by one expert at OO modeling; *and the level of experience using MDD tools*.

Identification of Hypotheses:

- **H₁₀:** When using INTEGRANOVA for modeling tasks with different levels of complexity, the efficiency is the same independently of users' experience level.
- **H₂₀:** When using INTEGRANOVA for modeling tasks with different levels of complexity, the effectiveness is the same independently of users' experience level. Since we have two measures for effectiveness, we can divide H₂₀ into two different hypotheses:
 - **H_{2A0}:** When using INTEGRANOVA for modeling tasks with different levels of complexity, the completeness percentage is the same independently of users' experience level.
 - **H_{2B0}:** When using INTEGRANOVA for modeling tasks with different levels of complexity, the optimum solutions percentage is the same independently of users' experience level.
- **H₃₀:** When using INTEGRANOVA for modeling tasks with different levels of complexity, the emotional response is the same independently of users' experience level

Selection of Participants. We utilize the convenience sampling method because of availability of subjects. Considering the level of experience with MDD tools, we used three groups of users:

- **Type I (Experts):** Experienced using the evaluated tool (INTEGRANOVA). 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 [16].
- **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*, and *Level of experience* with MDD tool (Figure 1). The *modeling competence* was not manipulated as a control variable, but it was considered in the selection process, since we consider that all the participants should have a minimum modeling knowledge.

Experimental Instrumentation. We used the following instruments:

- **A tool to record the user:** We used Noldus [18] in order to: 1) time the user and identify the completeness of each task; and 2) capture the facial expressions for evaluating the emotional response of users when they perform the tasks.
- **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. Tasks are listed in Table 1.

Table 1. List of tasks

1	Create a class called "rent"	10	Add a constraint in a class
2	Create a variable attribute and another constant	11	Edit a relationship created previously
3	Create a derived attribute	12	Create a profile for the administrator
4	Define a formula of the derivation.	13	Create a user profile with limited access
5	Create a service in a class.	14	Edit the name, help text and comments of an existing class
6	Create a relationship between two classes (car and rent) with the respective cardinality: A car can be rented zero or more times and a rent can only include a car.	15	Show agent relationships in the object conceptual schema
7	Create a class called "minibus" that inherits from car.	16	Define a precondition for an existing service
8	Specify the methods of creation and destruction for the class minibus	17	Create a dynamic relationship
9	Create an entry and an exit attribute for a method	18	Create a transaction

- **A pre-test:** Every participant fulfilled a list of questions to confirm our preliminary classification into one of three groups of participants. This way, we corroborated that the participants of the second and third group really had enough knowledge of OO concepts.
- The **Emocard tool**², which consists of 16 emocards, where there are eight male and eight female faces that together depict eight emotional expressions (See Figure 4.a). In addition, as these emotional expressions vary on the basis of pleasantness and arousal, which are the two most accepted dimensions of emotions in psychology, we calculated mainly two measures (See Figure 4.b): Deviation in pleasantness, measured by the difference on horizontal axis; Deviation in arousal, measured by the difference on vertical axis [1].

² An effective nonverbal measurement of emotional response, which uses human-like representations of emotion [7].

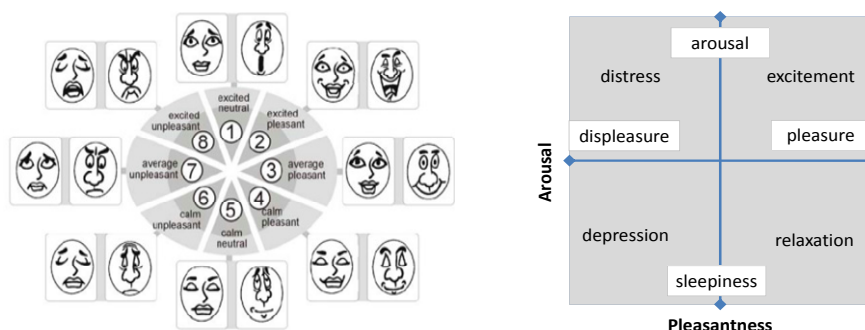


Fig. 4. a) The Emocards tool [7], b) Russell's circumplex model of affect [26]

These instruments and the guidelines used can be found at [9]. This step provides the information of the remaining classes in the usability evaluation model: *Modeling task, Activity, Solution, Step*, and all their inherited classes.

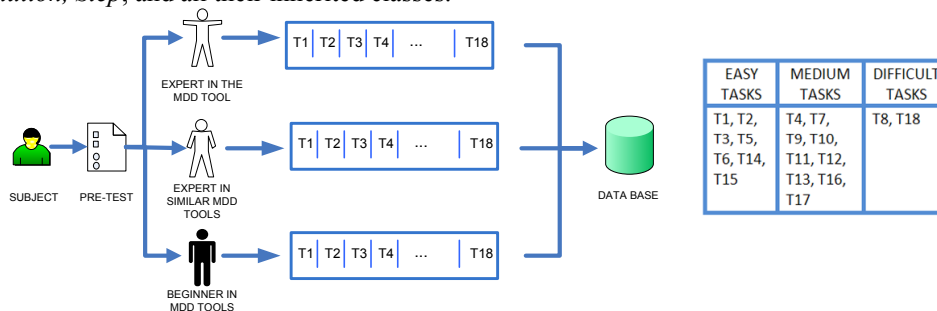


Fig. 5. Process of the Experiment

Process Design. Figure 5 shows a summary of the process to evaluate the usability of the MDD tool. First, the subject filled in the pre-test 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.

Threats to validity. According to [32] there are four kinds of validity that must be protected from threats:

- **Conclusion validity** is concerned with issues that affect the ability to draw the correct conclusion about relations between the treatment and the outcome of an evaluation. Our evaluation had the threat called *Random heterogeneity of subjects*. This threat appears when, within each user group, some users have more experience than others in the development of software systems. This threat was resolved with a pre-test that obtains the knowledge of each participant beforehand.
- **Internal validity** is about the influences that can affect the independent variables with respect to causality, without the researcher's knowledge. Our evaluation had the threat called *Maturation*: the effect that users react differently as time passes. We minimized this threat limiting the evaluation to one hour. Another internal validity threat that our evaluation suffered was *Instrumentation*: even though tasks and questionnaires were the same for all subjects, a wrong interpretation of the task may affect the results. This threat was minimized by the researcher, who helped the subject to understand the tasks.
- **Construct validity** concerns generalizing the results to the concept of theory behind the evaluation. Our evaluation had the threat called *Hypothesis guessing*: users may guess the purpose and intended result of the evaluation and base their behavior on those guesses. This threat was resolved by hiding the goal of the evaluation from the users. Another threat that appears in our evaluation was *Experiment expectancies*: people who have done the

evaluation can talk to future subjects about their experience. This can bias the results based on what the future subjects expect from the evaluation. This threat was resolved by warning subjects against talking to future subjects. Moreover, our evaluation had also the threat called *Evaluation apprehension*: This means that the performance of some users (e.g. experts) can win a positive evaluation from the evaluator (experimenter). To reduce this subjectivity we recommend involving at least two evaluators for the evaluation of effectiveness and satisfaction, where evaluators have to use qualitative measures.

- **External validity** is related to conditions that limit our ability to generalize the results of our evaluation to industrial practice. Our evaluation might suffer from *Interaction of selection and treatment*: the subject population might not be representative of the population we want to generalize. We have tried to use a confidence interval where conclusions are 95% representative. This means that if they followed a normal distribution, results would be true for 95% of the times the evaluation would be repeated. Moreover, we chose all our participants randomly among the possible ones. However, as the number of participants by group is very low, we plan to replicate the study, scaling-up our sample.

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 6a shows the results obtained for this variable.

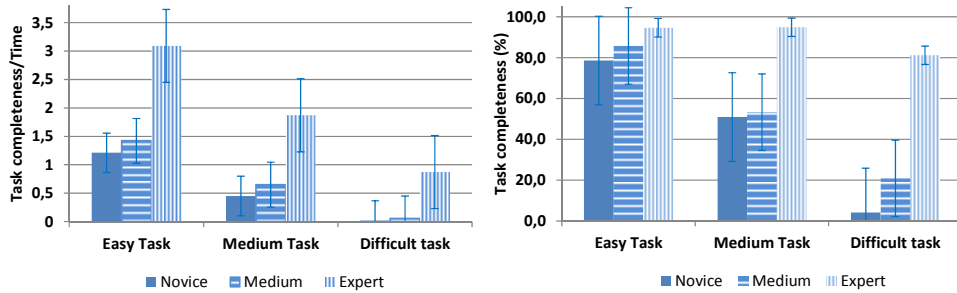


Fig. 6. a) Comparison of efficiency by user type and task complexity level; b) Comparison of modeling task completeness by user type and task complexity

According to a Kolmogorov-Smirnov test, efficiency follows a normal distribution. A paired-samples t-test was conducted to compare the efficiency (task completeness/time) between levels of task complexity. Such as Table 2 shows, there is a significant difference in the efficiency mean values obtained for the three groups of tasks (easy, medium and difficult). These results corroborate that the level of task complexity has an effect on the efficiency to perform the task.

Table 2. Paired Samples test for the efficiency in different task complexity levels

Efficiency	Paired Differences		t	df	Sig. (2-tailed)
	Mean	Std. Deviation			
Easy tasks – Medium tasks	0.915	0.434	7.291	11	0.000
Medium tasks –Difficult tasks	0.673	0.554	4.210	11	0.001
Easy tasks-Difficult tasks	1.589	0.813	6.766	11	0.000

An ANOVA test was also conducted to compare the effect of type of user (experience level) on the efficiency of performing the modeling tasks with different level of complexity (Hypothesis H_{10}). There was a significant effect of the experience level on efficiency in performing easy tasks at the $p < 0.05$ level for novice, medium, and expert users [$F(2,9)=8.76$,

$p=0.008$]. There was also a significant effect of the experience level on efficiency of performing medium tasks [$F(2,9)=9.28$, $p=0.007$] and difficult tasks [$F(2,9)=187.76$, $p=0.000$].

We then compute all possible comparisons between means, by using the Bonferroni post hoc test (See Table 3). These results suggest that a high level of experience (experts) really do have an effect on efficiency of performing modeling tasks with the INTEGRANOVA tool. Medium and Novice levels do not affect the efficiency significantly. However, we need to replicate this study with more users to confirm these results.

Table 3. Pairwise multiple comparisons between users experience levels for tasks complexity levels

Dependent Variable		(I) Experience level	(J) Experience level	Mean Difference (I-J)	Std. Error	Sig.
Efficiency mean values	Easy tasks	Novice	Medium	-0.216	0.491	1.000
			Expert*	-1.879	0.491	0.012
		Medium	Novice	0.2169	0.491	1.000
			Expert*	-1.662*	0.491	0.024
		Expert	Novice*	1.879	0.491	0.012
			Medium*	1.662*	0.491	0.024
	Medium tasks	Novice	Medium	-0.207	0.356	1.000
			Expert*	-1.422*	0.356	0.010
		Medium	Novice	0.2070	0.356	1.000
			Expert*	-1.214*	0.356	0.023
		Expert	Novice*	1.422*	0.356	0.010
			Medium*	1.214*	0.356	0.023
	Difficult tasks	Novice	Medium	-0.027	0.049	1.000
			Expert*	-0.8519	0.049	0.000
		Medium	Novice	0.027	0.049	1.000
Expert*			-0.824	0.049	0.000	
Expert		Novice*	0.851*	0.049	0.000	
		Medium*	0.824*	0.049	0.000	

*The mean difference is significant at the 0.05 level

Applying the GLM repeated measures test for efficiency, analysing the within-subjects effects, we found that there is an effect of the task complexity level ($F=45.186$; $p=0.000$), but there is not a significant interaction between the two variables task complexity/user experience level ($F=1.785$; $p=0.176$). Analysing the between-subjects effects, we also found a significant effect of user experience level ($F=102.67$; $p=0.000$). From all these results we reject the null hypothesis H_{10} and we conclude that using INTEGRANOVA for modelling tasks with different levels of complexity, only expert users have an effect on efficiency.

Analyzing effectiveness. Effectiveness was measured in terms of modeling task completion percentage and the percentage of correct tasks that were carried out optimally. Figure 6b shows the results obtained for the task completion percentage. A similar completeness percentage is only observed when the users performed tasks with an easy level of complexity. Moreover, we note that some experts did not manage to complete correctly all easy tasks, such as T6 and T14 (See Table 1). For example, two experts partially performed the task T6, where the common mistake was the definition of the cardinality in a relationship. Our explanation is that the description of the task T6 (where the term “cardinality” was not explicitly specified) might have caused a misunderstanding and consequently an incomplete completion of this task. However, with respect to the task T14, the low frequency of use of this functionality could be the reason of the bad completeness percentage when experts performed this task. So, if some users (e.g. experts) could not complete this task, it seems to be that there is a usability problem with the tool to change the name of existing classes (T14).

We also observed in Figure 6b that novice and medium users had a similar completeness percentage (55%) for tasks with medium level of complexity. A greater difference with respect to the completeness percentage between the type of users is more noticeable for difficult tasks.

In order to know whether the user type has an effect on overall completeness of tasks with different complexity levels, an ANOVA test was used (data normality and homogeneity of variances were corroborated). With this test, significant differences were found only for both the tasks of medium level of difficulty [$F(2,9)=8.524$, $p=0.008$] and high level of difficulty [$F(2,9)=13.562$, $p=0.002$]. Post-hoc pairwise comparisons (Bonferroni test) showed that only expert users do have an effect on the completion percentage of medium and difficult tasks performed with the INTEGRANOVA tool (See Table 4).

Table 4. Pairwise multiple comparisons between user experience levels for the task completion percentage

Dependent Variable		(I) Experience level	(J) Experience level	Mean Difference (I-J)	Std. Error	Sig.
Task completion percentage	Medium tasks	Novice	Medium	-2.315	11.98	1.000
			Expert	-43.980*	11.98	0.016
		Medium	Novice	2.315	11.98	1.000
			Expert	-41.667*	11.98	0.021
		Expert	Novice	43.982*	11.98	0.016
			Medium	41.667*	11.98	0.021
	Difficult tasks	Novice	Medium	-16.667	15.55	0.936
			Expert	-77.000*	15.55	0.002
		Medium	Novice	16.667	15.55	0.936
			Expert	-60.332*	15.55	0.011
		Expert	Novice	77.000*	15.55	0.002
			Medium	60.332*	15.55	0.011

*The mean difference is significant at the 0.05 level

Applying the GLM repeated measures test, there is a within-subjects effect of the complexity task level ($F=32.675$; $p=0.000$) and medium significance of the interaction between the complexity task and user experience variables ($F=4.579$; $p=0.01$). A very highly significant effect of the user experience level was also found ($F=339.290$; $p=0.000$). Considering all these results, the null hypothesis H_{2A0} is rejected. Therefore we can conclude that the task completeness percentage depends on users' experience level.

With respect to our second effectiveness measure, the aggregation method presented in the third section (See Figure 2) was applied in order to calculate the percentage of activities carried out in an optimum way. Then, the percentages obtained were averaged over the number of tasks classified as easy (7 tasks), medium (9 tasks), and difficult (2 tasks).

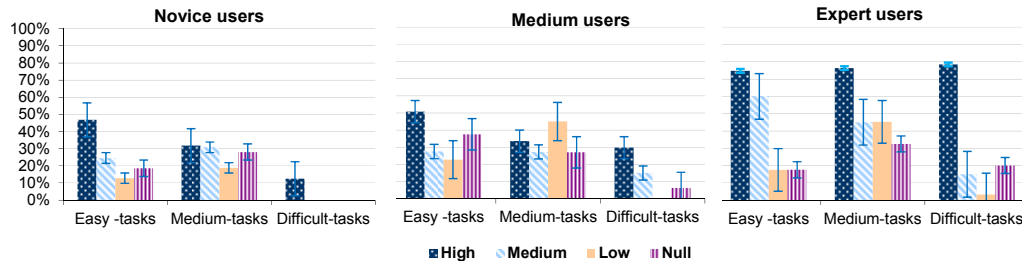


Fig. 7. Comparison of optimum level of activities performed by task complexity and type of user

As shown in Figure 7, in novice mode, the average percentage of activities performed with a high-optimal level (blue bar) decreases while the complexity of the tasks increases. Moreover, we also note 1 out of 4 novice users achieved to perform one difficult task with a high-optimal level (Task 8). However, we cannot confirm this evidence as significant. In expert mode, the average percentage of activities performed with a high-optimal level is almost constant (about 80%). However, we also note 33% of activities with medium complexity were carried out with

a null-optimal level. 2 out of 4 expert users had high difficulty to perform the task 7 and task 12.

By applying an ANOVA test, it is suggested that different types of users tend to affect overall percentage of activities performed optimally (easy tasks), with a low significance level, [F(2,9)=3.560, p=0.073]. However, for the comparison of tasks with medium and difficult complexity, a higher significance was observed, [F(2,9)=8.259, p=0.009] and [F(2,9)=4.872, p=0.03] respectively. Analysing the Post-hoc pairwise comparisons, we corroborated that only expert users have an effect on the percentage of optimal solutions for medium and difficult tasks. These results suggest that the INTEGRANOVA tool tend to be less usable for novice and surprisingly also for users with medium experience level. This can be due to some functionalities offered by INTEGRANOVA are not entirely intuitive (ease of use).

Applying the GLM repeated measures test, two effects, within and between-subjects, were analysed. Considering the test of within-subjects effects, we found that there is no main effect of the task complexity level (F=2.048; p=0.158), nor the interaction of both variables, task complexity and user experience (F=0.937; p=0.465). According to these results, the null hypothesis H_{2B0} cannot be rejected.

Analysing the test of between-subjects effects, we confirm that there is a main effect of user's experience level on the optimum solutions percentage. As shown in Table 5, this effect is mainly due to the performance of experts users, that make a big difference from the other users.

Table 5. Multiple Comparisons-Bonferroni test

(I) Type of user	(J) Type of user	Mean Difference (I-J)	Std. Error	Sig.
Novice	Medium	-0.0767	0.10603	1.000
	Expert	-0.4629*	0.10603	0.005
Medium	Novice	0.0767	0.10603	1.000
	Expert	-0.3862*	0.10603	0.016
Expert	Novice	0.4629*	0.10603	0.005
	Medium	0.3862*	0.10603	0.016

Analyzing satisfaction. Three evaluators were used to observe the videos where the user interacts with the tool in order to gather the Emocard data, (Figure 8a). We used an Emocard for each of the 18 tasks (Figure 8b). For each Emocard, the evaluator marked the cells that best fitted subject's feelings throughout the task.

We count the total number of subjects with a response on the respective "emotional expression" for each one of tasks across the 12 subjects (4 experts, 4 mediums and 4 novices). Then, the number of subjects that most frequently occurs in any emotional expression, for the respective type of tasks, is also calculated (mode).

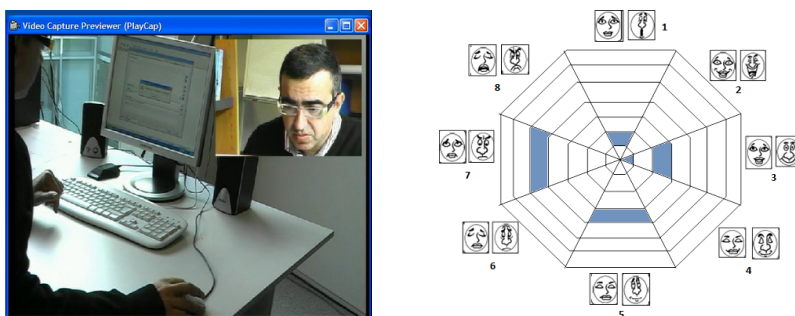


Fig. 8. a) Display visualized using the Noldus tool, b) Emocard used to gather emotional responses perceived by the evaluator

The radar charts (Figure 9) show us how these mode values changes from one to other “emotional expression” for each type of task.

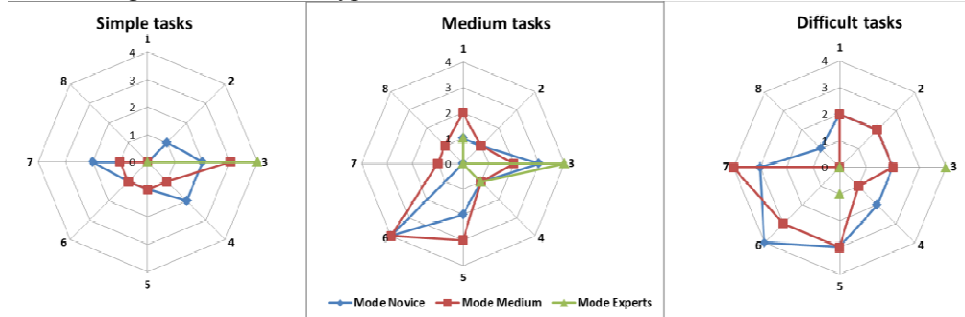


Fig. 9. Distribution of the emotional responses that most frequently occur for the three types of tasks within the circumplex of emotions

For example, we can see an increase of number of subjects for the third and fourth quadrant. According to the Russell’s circumplex model (Figure 4b), it means that the subjects tend to experience more depression and distress, when the task complexity increases. This is observed for the medium and difficult tasks, mainly when the subjects have a novice and medium level of experience.

Moreover, as these expressions vary on the basis of pleasantness and arousal, the standard deviation on these two dimensions and the average emotional responses can be seen in Figure 10.

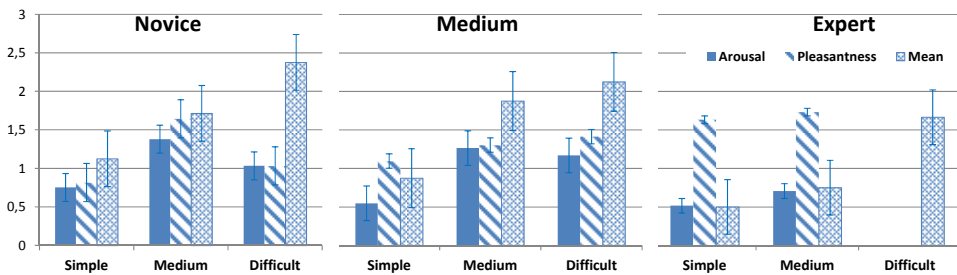


Fig. 10. Deviation from perceived emotion by task complexity and type of user

4.4 Detected Usability Problems

This section details usability problems of INTEGRANOVA detected with our proposed framework. We consider 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, since they got the poorest marks. Even though some expert users did not get good efficiency and effectiveness values for some tasks such as 7 and 12.

Next, we describe how usability problems were detected. We watched the videos that record the interaction between the end-user and INTEGRANOVA. For each task, we measured the time spent to finish, the percentage of task successfully completed and the level of optimality to fulfill the target. The time was extracted from the subtraction of the starting time and the ending time. Moreover, we divided the tasks into activities (according to Figure 1). The percentage of task successfully completed was calculated as the percentage of activities successfully performed. In its turn, activities were divided into steps (according to Figure 1). The level of optimality per tasks was measured counting the optimum steps performed by the end-user. Time, percentage of completed tasks and optimality were used to calculate the effectiveness and efficiency. Tasks with bad results in effectiveness or efficiency were specially studied. We tried to classify the end-users’ mistakes with regard to the Bastien and

Scapin ergonomic criteria classification. Table 6 shows the tasks grouped by difficulty. For each task, we have marked the usability problems detected according to the classification of Bastien and Scapin' ergonomic criteria [6].

Table 6. Usability problems found by task

Task	Easy							Medium							Difficult			
	1	2	3	5	6	14	15	4	7	9	10	11	12	13	16	17	8	18
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	

5 Comparison with an evaluation with ergonomic criteria

The evaluation with our framework and end-users has provided a list of usability problems, but we cannot ensure that these problems are all the existing problems in the tool. In order to evaluate the capacity of our framework for detecting usability problems, we have also performed a usability evaluation of INTEGRANOVA based on ergonomic criteria [6]. This way, we can compare the usability problems detected with our framework and the usability problems detected with these criteria.

Table 7. Usability problems detected in the evaluation with ergonomic criteria

Guidance	Novice users are not guided
	Some elements can confuse novice users
	Help windows do not work properly in every context (+)
	Some interfaces do not save the changes if the user does not save them explicitly
Workload	Interfaces are not visually consistent among them (+)
	Some interfaces provide too much information with regard to the available space
User Control	The tool does not provide undo and redo facilities (+)
	Some elements do not provide a menu when the user clicks with the right button of the mouse
	Some interfaces are obtrusive (they do not allow showing the window below) (+)
Adaptability	There is no option to customize the tool
Error management	Error messages do not help the user to solve the mistake
Significance of code	Icons are not self-explicative
	Some interface titles are confusing (+)
	Some functions are only reachable by means of icons, but not through the menu (+)

Next, we compare the results of the evaluation with ergonomic criteria with the results of our framework. Figure 11 shows a comparative graph of all the usability problems detected with our proposed framework (Table 6) and with the ergonomic criteria (Table 7). Some problems were detected both with the framework and with the ergonomic criteria, while others problems were detected exclusively with one of both methods. Problems detected exclusively with our framework can be consulted in [9], and problems detected exclusively with the ergonomic criteria have been marked with “(+)” in Table 7. Figure 11 shows the amount of problems detected per method and which ones are exclusive of the method. In general, we can state that most of the usability problems detected with ergonomic criteria have been also detected with our framework. Moreover, our framework has detected some usability problems that the evaluation with ergonomic criteria did not discover. Next, as example, we explain

some usability problems that were not detected with the evaluation based on ergonomic criteria:

- Guidance: Interfaces do not save data between tabs.
- Workload: Elements in a list are not editable.
- User control: The tab key does not work properly.
- Error management: There is no early validation of inserted elements.

A few usability problems detected with ergonomic criteria were not detected with our proposed framework. For example, two problems of the User control criteria (Table 7) were not detected: some elements do not provide a menu when the user clicks with the right button of the mouse; some interfaces are obtrusive. The usability experts who performed the evaluation with ergonomic criteria were specialist in interaction not in software development, and they considered that both features are useful for end-users. However, the subjects did not need contextual menus or non-obtrusive interfaces to perform any task; this is the reason why they were not detected in the evaluation with subjects.

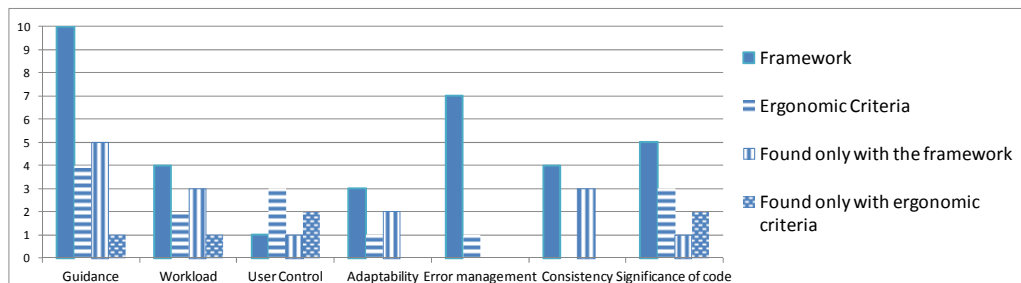


Fig. 11. Comparison of errors obtained with our framework and with the ergonomic criteria

It is important to mention that the comparison between the evaluation performed with our framework and with ergonomic criteria does not ensure that our framework detects all usability problems. However, the comparison shows that the framework can obtain better marks than an evaluation based on ergonomic criteria; the number of usability problems detected is higher and they are more real according to end-users' needs. It is also important to compare the cost of our proposal with regard to an evaluation based on ergonomic criteria. The evaluation with criteria took two hours per usability expert (2 persons), while the application of our framework took almost two hours per subject (12 persons). If we focus our comparison only on spent time, the evaluation with ergonomic criteria is clearly better than our proposal. However, there are other elements to consider apart from the cost before choosing an evaluation method. Usability problems with ergonomic criteria are usually far from the real necessities of the end-users. If we would like to obtain a usability evaluation close to real problems and to detect a high percentage of usability problems, the evaluation with end-users obtains better marks, even though the required time to perform the evaluation is clearly longer.

6 Conclusions and Future Work

There are several key factors for the acceptance of MDD tools in industrial systems, such as the learning curve, the tool maturity, and the resistance to change. This paper focuses on one of these factors: the usability. For this aim, we propose 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. However, the accuracy of the results depends on how the evaluation is designed. For example, a bad selection of Modeling Tasks could concentrate all the evaluation in a small part of the tool, and bias the evaluation, or a bad choice of End-users could bias the results (if all the End-Users have similar background).

For the purpose of illustrating the applicability of our framework, we evaluated a MDD tool called INTEGRANOVA, by using analysts with different level of experience using MDD

tools. Interesting conclusions were extracted from this empirical evaluation. Firstly, with regard to efficiency, we found that there are no significant differences between medium users (familiarized with other MDD tools) and novices (no familiarized). This fact means that INTEGRANOVA does not share many features with regard to MDD tools used by medium users. This is because INTEGRANOVA 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. 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.

Thirdly, with regard to satisfaction, the evaluation technique selected, Emocard, was fast and easy for the evaluator to understand. We calculated the deviation in perceived pleasantness and arousal. Clear differences about how users (novice, medium and experts) reacted to the interfaces when they performed tasks with different complexity difficulty were identified. Novice and medium users tend to experience more distress when the task complexity increases. However, experts showed a greater emotional stability than other end-users.

The results of the evaluation demonstrate that changes must be applied to INTEGRANOVA to improve the users' effectiveness and efficiency. We are currently preparing a detailed internal report for the company that develops INTEGRANOVA.

As future work, we plan to repeat our evaluation with more users and more complex tasks. Moreover, we want to assess the applicability of our approach with other MDD tools different from INTEGRANOVA. The evaluation model proposed in this paper is based on the ISO 9241-11 which is focused on the usability attributes: efficiency, effectiveness and satisfaction. Since the learning curve of MDD tools is high, we plan to extend in future works our proposal with other usability attributes such as learnability.

Acknowledgments

This work has been developed with the support of, Intra European Marie Curie Fellowship Grant 50911302 PIEF-2010, MICINN (TIN2008-00555, PROS-Req TIN2010-19130-C02-02), GVA (ORCA PROMETEO/2009/015), and co-financed with ERDF. We also acknowledge the support of the ITEA2 Call 3 UsiXML (20080026) and financed by the MITYC under the project TSI-020400-2011-20. Our thanks also to Ignacio Romeu for the video data gathering setup.

References

1. Agarwal A. and Meyer A.: Beyond usability: evaluating emotional response as an integral part of the user experience. In Proceedings of the 27th international conference extended abstracts on Human factors in computing systems (CHI 2009). ACM, New York, NY, USA, 2919-2930.
2. Al-Wabil, A., Al-Khalifa, H.: A framework for integrating usability evaluations methods: The Mawhiba web portal case study. Current Trends in Information Technology (CTIT), Dubai (2009).
3. 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.
4. ArgoUML: <http://argouml.tigris.org/>
5. Basili, V., Shull, F., Lanubile, F. Building knowledge through families of experiments. *IEEE Transactions on Software Engineering*. 25(4), (1999).
6. Bastien, J.M., Scapin, D.: Ergonomic Criteria for the Evaluation of Human-Computer Interfaces. *Rapport technique de l'INRIA* (1993) 79
7. Desmet P. A.. Measuring emotions - development and application of an instrument to measure emotional responses to products. In M. A. Blythe, A. F. Monk, K. Overbeeke, and P. C. Wright, editors, *Funology: from usability to enjoyment*, pages 111–124. Kluwer Boston, Inc., 2003.
8. Fiora T. W. Au, Baker, S., Warren, I., Dobbie, G.: Automated Usability Testing Framework. *Proc. 9th Australasian User Interface Conference (AUI2008)* (2008).
9. Instruments: <http://hci.dsic.upv.es/testINTEGRANOVA/>

10. INTEGRANOVA: www.integranova.es/
11. ISO 9241-11 Ergonomic requirements for office work with visual display terminals - Part 11: Guidance on Usability (1998).
12. ISO/IEC 9126-1: Software engineering - Product quality - 1: Quality model. (2001)
13. Kostiainen, K., Uzun, E.: Framework for Comparative Usability Testing of Distributed Applications. Security User Studies: Methodologies and Best Practices Workshop (2007).
14. 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.
15. Metaedit: <http://www.metacase.com/mep/> Last visit: June 2009
16. Moskitt <http://www.moskitt.org/>
17. Nielsen, J.: Usability Engineering. Morgan Kaufmann (1993).
18. Noldus: www.noldus.com
19. Panach, J.I., Condori-Fernández, N., Baars, A., Vos, T., Romeu, I., Pastor, O.: An Experimental Usability Evaluation Framework for Model-Driven Tools. In: Congreso de Interacción Persona-Ordenador (INTERACCIÓN 2011). Lisbon (Portugal) (2011) 67-76.
20. Pastor, O., Molina, J.: Model-Driven Architecture in Practice. Springer, Valencia (2007).
21. 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).
22. Post, G., Kagan, A. User requirements for OO CASE tools., Information and software technology, Vol. 43, N° 8, (2001), 509-517.
23. Rational Rose: <http://www-01.ibm.com/software/awdtools/developer/rose/>
24. Rational Rose: <http://www-01.ibm.com/software/awdtools/developer/rose/>
25. Russell, J. A. A circumplex model of affect. Journal of personality and Social Psychology, 39, 1161-1178, (1980).
26. 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).
27. Selic, B.: The Pragmatics of Model-Driven Development. IEEE software 20 (2003) 19-25
28. Senapathi, M.: A Framework for the Evaluation of CASE Tool Learnability in Educational Environments. Journal of Information Technology Education 4 (2005), 61-84.
29. Singh, Y., Sood, M.: Model Driven Architecture: A Perspective. Advance Computing Conference, 2009. IACC 2009. IEEE International (2009) 1644-1652.
30. 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.
31. Wieringa R.J., Requirements Engineering: frameworks for understanding, Wiley, (1996).
32. Wohlin, C., Runeson, P., Höst, M. Experimentation in Software Engineering: An Introduction, Springer, (1999).