A Family of Experiments to generate Graphical User Interfaces from BPMN Models with Stereotypes

Eduardo Díaz[1], José Ignacio Panach[1], Silvia Rueda[1], Damiano Distante[2]

[1] Escola Tècnica Superior d'Enginyeria, Departament d'Informàtica, Universitat de València, Avenida de la Universidad s/n, 46100 Burjassot, València, España

[2] University of Rome Unitelma Sapienza, Viale Regina Elena 295, 00161 Rome, Italy

[1]diazsua@alumni.uv.es, [1]joigpana@uv.es, [1]silvia.rueda@uv.es, [2]damiano.distante@unitelmasapienza.it

Abstract: **Context:** A significant gap separates Business Process Model and Notation (BPMN) models representing processes from the design of Graphical User Interfaces (GUIs). **Objective:** This paper reports on a family of experiments to validate a method to automatically generate GUIs from BPMN models using stereotypes complemented with UML class primitives, and transformation rules. **Method:** We conducted two replications (23 and 31 subjects respectively) in which we compared two methods to generate GUIs from BPMN models; one automatic (using Stereotyped BPMN models) and one manual (using Non-stereotyped BPMN models). The study focuses on comparing effort, accuracy, and satisfaction (in terms of perceived ease of use (PEOU), perceived usefulness (PU), and intention to use (ITU)). **Results:** Results yield significant differences for Effort, Accuracy, and ITU. Effort is lower for the Non-stereotyped method, while accuracy and ITU are higher for the Stereotyped one. If we consider only experimental units whose BPMN models show an accuracy over 75% compared to those of the experimenters' solution, the difference in accuracy for the designed GUIs is even more significant; in contrast, differences for effort and ITU are reduced. **Conclusions:** The use of the Stereotyped method reduces the possibility of errors in the process of designing GUIs.

Keywords: Business Process Model and Notation models, Graphical User Interfaces design, Stereotypes, Empirical study.

1 Introduction

The Business Process Model and Notation (BPMN) is a standard developed by the Object Management Group (OMG) that provides businesses with the capability to represent and understand their internal business procedures using a graphical notation and to communicate these procedures in a standard manner [1]. Furthermore, the graphical notation facilitates the understanding of the performance of collaborations and business transactions between organizations [2]. A BPMN model does not represent the functional behavior of the system. The basic conceptual primitives in BPMN are *events*, *gateway*, *pool*, *lane*, *flows,* and *tasks* [1]. There are two types of tasks mainly used [3]: *user tasks* (carried out by user with the help of a system or software) and *service tasks* (carried out by a system without human intervention, for example, web services or an automated application). In all the existing modeling tools, the design of the graphical user interfaces (GUIs) of the system that will implement or support the business process is put in the background, leaving this task to the analyst. This analyst must make an effort to define interfaces according to the BPMN model. The interfaces are usually derived from BPMN models manually, without any kind of procedure, only relying on the analyst experience. This means that the effort made in building the BPMN model is not useful when designing the interfaces at the end. In addition, normally, analysts that build the BPMN models are not the same designers who define the GUIs, generating a gap between what is described in the BPMN models and what the interface really implements. The main problem is that the analyst invests more time in understanding the business process than in designing the GUIs.

In order to cover the gap between the BPMN models and the GUIs, in a previous work of ours, we proposed a set of rules to automatically generate code from BPMN models [4]. The approach is based on the extension of the BPMN notation with stereotypes and the use of UML class diagrams to represent data persistency. The stereotypes specify which rule must be applied for each BPMN primitive to generate a widget in the GUI. Stereotypes were extracted after analyzing 12 previously existing Bizagi [5] projects (Bizagi has a BPMN modeler and also a BPMN model repository with the implementation of interfaces), where we compared the BPMN models with their real implementation. Through this comparison, we abstracted a set of rules to automate the generation of the GUI: each design alternative in the GUI involved a stereotype in the BPMN model to specify such alternative. The rules generalization was classified according to five BPMN patterns widely used: sequence, exclusive decision, synchronization, implicit decision, and structure union synchronization. We have already implemented a template based on Visual Paradigm v. 15.0 [6] with stereotypes which generates code in PHP and HTML5 [7]. This tool does not generate code to manage data persistency or functional behavior, but it only generates code for non-functional interfaces.

The code generation approach used in our proposal [4] is aligned with the Model-Driven Development (MDD) [8] [9] paradigm, which advocates focusing the effort of the developer on models and delegating the code generation to automatic transformations from such models. According to the literature, some of the main advantages of MDD that we want to leverage in our proposal [4] are:

1. Code generation for different platforms [10] [11].
2. Reduced development effort [12] [13] [14] [15].
3. Better models' accuracy [16] [17].
4. Higher developers' satisfaction [18].

In the present paper, we aim to assess the above advantages when adopting our MDD proposal.

Following the MDD paradigm, currently, several modeling tools allow drawing BPMN diagrams and generating GUIs from them. The list of such tools includes Bizagi [19], Auraportal [20], BonitaSoft [21], E-citiz Studio [22] and WebRatio [23]. All these tools share the peculiarity for which the generation of GUIs requires an interaction model that abstractly represents the GUIs. Our proposal to generate GUIs from BPMN models differs from all these tools in the fact that it does not depend on any interaction model. We propose extending the BPMN models through stereotypes that provide enough expressiveness to generate any widget.

The main contribution of this paper is the design and analysis of a family of experiments that we conducted to assess the *effort*, *accuracy*, and *satisfaction* (measured in terms of Perceived Ease to Use (PEOU), Perceived Usefulness (PU), and Intention to Use (ITU) [24]) when using our method to generate GUIs from stereotyped BPMN models. The design is within-subjects, where we recruited 23 and 31 subjects in two replications, respectively. All the subjects are undergraduate computer science students of the academic years 2018/2019 and 2019/2020 of the University of Valencia (Spain) (one course per replication). The control group was a non-stereotyped method where the subjects had to build standard BPMN models from requirements, and manually draw non-functional GUI prototypes from BPMN models. The experiment was conducted through two experimental problems in each replication to ensure the independence of results from the used problem. To compare the two methods (Stereotyped and Non-Stereotyped), we analyzed the whole software design process, from requirements elicitation to GUIs design (using BPMN models as a connection between requirements and GUIs). The process starts watching a video with slides and a voice-over describing the requirements of the problem. For both treatments, the subjects had to build the BPMN models and to generate the GUIs. The differences between treatments rely on the use of stereotypes and on whether the GUI generation is automatic.

We have aggregated the data of each replication through a moderator variable named Course (the course where the replication was conducted). The results of the experiment yield that there are significant differences between the Non-Stereotyped method and the Stereotyped method for effort, accuracy, and ITU. In particular, the Non-Stereotyped method requires less effort to build the GUIs, while accuracy and ITU are higher in the Stereotyped method. In order to assess the accuracy of the GUIs, we repeat the analysis considering experimental units with a high accuracy in their BPMN models. It resulted that for subjects whose accuracy in the BPMN models was over 75% compared to the experimenters' solution (i.e., whose BPMN models overlap with the experimenters' solution for 75% and more), differences for effort, and ITU are reduced, while differences in accuracy are more significant. If we repeat the analysis considering only subjects which reached a 100% accuracy in their BPMN models, differences for accuracy and ITU disappear. This result highlights one of the key characteristics of the MDD paradigm: the model is the code. All experimental units whose BPMN models were 100% accurate reached 100% of accuracy in the GUIs. Considering that subjects involved in the experiment were undergraduate computer science students with no or very little experience in conceptual modeling, we can also conclude that the Stereotyped method yields better accuracy even when adopted by subjects not expert in conceptual modeling.

The rest of the paper is structured as follows. Section 2 analyzes the design of other families of experiments in the area of GUIs generation from models. Section 3 describes how to generate GUIs from BPMN models. Section 4 defines the design of the experiment. Section 5 discusses threats to validity, as occurred in the experiment conducted in Section 4. Section 6 shows the statistical results after analyzing the data extracted from the experiment. Section 7 discusses the interpretation of the results. Finally, Section 8 presents some relevant conclusions and future works.

2 Related Works

In this section, we review works related to the generation of GUIs from BPMN models, the extension of the BPMN standard notation, and the generation of other models (apart from GUIs) from BPMN models. We conducted a Targeted Literature Review (TLR), a non-systematic, in-depth, and informative literature review aimed at keeping only the significant references to maximize rigorousness while minimizing selection bias. For this purpose, the semantic question about the generation of GUIs from BPMN models is translated into the following syntactical query used as a search string on the Scopus digital library (See https://www.scopus.com/home.uri):

"BPMN" AND ("user interface" OR "model" OR "class" OR "use case" OR "experiment" OR "extension"). References resulting from this search were classified into five categories, which are further discussed in next subsections.

2.1. Graphical User Interfaces Generation from BPMN Models

Since BPMN only represents aspects of business processes, a BPMN model needs to be complemented with additional information in order to be mapped to GUIs. This information is quite heterogeneous, depending on each proposal. Marco Brambilla et al. [25] proposed a process design methodology supported by a tool suite named Social BPMN. The methodology is composed of several parts: business process models to work with social features, extensions of BPMN for capturing social requirements, a gallery of social BPMN design patterns that represent reusable solutions to recurrent process socialization requirements, and a model-to-model and mode-to-code transformation technology that automatically produces web code. In another work of Brambilla et al. [26], business processes are described through a BPMN model extended with information about task assignment, escalation, policies, activity semantics, and typed data flows. They present WebRatio BPMN, a model-driven web application development tool that allows editing BPMN models and automatically transforming them into running JEE applications. Han et al. [27] defined an approach for the derivation of user interfaces from BPMN models. The approach is based on a role-enriched business process

model developed with tasks' descriptions and associated data. The model is specified using an extended version of BPMN. A set of control flow and data flow patterns are identified for the GUI derivation. A comprehensive set of constraints and recommendations are specified for supporting the GUI generation and update. Sousa et al. [28] proposed an approach to correlate a business process to user interfaces by (1) defining associations between business processes and user interface models, and (2) presenting a tool for model transformation that addresses traceability. Yongchareon et al. [29] proposed a model-based automatic user interface generation framework with algorithms to derive user interfaces from process models. A User Interface Flow (UIF) model (also called storyboard) reflects the logic of business processes and intuitively represents what information is required during the process. UIF models include two abstract aspects: behavioral aspect (navigational control flow relations between UIs) and informational aspect (related/required data for each UI). Yongchareon et al. [30] defined a framework for deriving UIF models to help visualize artifact centric processes and support the semi-automatic creation of user interfaces. The UIF model is created by considering the relationships between business process, user interfaces, and user roles in an artifact centric process model. Algorithms are also developed to derive UIF models from an artifact centric process model. Bouchelligua et al. [31] defined an approach that shows a set of model transformations according to a Model-Driven Engineering (MDE) method. AfFirst transformation is used to derive plastic UIs from a workflow; a second one lies in the use of BPMN for modeling interaction (Task Model, Abstract User Interface and Concrete User Interface).

To summarize related works considered in this sub-section, we can state that they use data models [26] [28], patterns [25] [27] [32], model-based [31] [29] to semi-automatically generate GUIs. One of the main concerns with the existing works is that the developer has to spend more effort in drawing several models to generate GUIs. On the contrary, we propose extending the BPMN model through stereotypes to generate user interfaces without the need of learning other models.

2.2. BPMN Extensions

We have found a number of works that propose BPMN extensions. Abouzid et al. [33] defined a set of BPMN extensions which represent some crucial concepts of the manufacturing domain to improve the business process. The BPMN extensions allow putting much more information into the process model, which, from a manufacturing point of view, makes the process more complete to define the process to be used in business process improvement approaches. Decker et al. [34] showed that BPMN fails to capture advanced choreography scenarios. To overcome this limitation, the authors proposed a BPMN extension that broaden the applicability of BPMN; the proposal was validated using Service Interaction Patterns, which describes a set of recurrent choreography scenarios [35]. Rodriguez et al. [36] extended BPMN to incorporate security requirements into business process diagrams from the perspective of the business analyst according to a Model Driven Architecture (MDA). With this extension, the business analyst will be able to express security requirements from their own perspective. Stroppi et al. [37] presented a BPMN extension developed by using the extension mechanisms provided by the BPMN 2.0 metamodel. They focused on three main aspects of the resource perspective [38]: resource structure, authorization, and work distribution. The aim is to improve the communication of the resource perspective requirements between analysts and technical developers. Braun et al. [39] presented an extension that focuses on domain analysis, requirements, and concepts of BPMN. This paper is aligned with the paradigm of design-oriented information systems research [40] and the derived extension model can also be transformed into a Business Process Execution Language (BPEL) [41] model in order to support process model execution. Zarour et al. [42] proposed a BPMN extension (called BPOMN) that allows specifying the requirements of business process activities in terms of security, compliance, cost, and performance. BPOMN was developed on the basis of two representations provided by the BPMN extension mechanism: the MOF meta-model and the XML Schema. These two representations were supplemented by new graphical elements that were integrated into a modeling tool.

As a conclusion on works considered in this sub-section, we can state that there are proposals that extend BPMN in terms of manufacture domain [33], choreography scenarios [34], security [36] [42], resources [37], and e-health [39]. So, we highlight that the idea of extending BPMN models is widely used, as we do in our proposal.

2.3. Generation of Other Models from BPMN Models

The group of works summarized in this sub-section deal with the generation of other models from BPMN ones. Cruz et al. [43] proposed a model-driven approach to support the construction of a UML use case model, an integrated domain model, and a user interface model from a set of business process models. The proposed approach produces a complete use case model including the identification of actors, uses cases and the corresponding descriptions, relations among use cases, and between these uses cases and the structural domain classes. The approach for deriving use case models is new, although some of the rules are based on a previous approach proposed in [44]. Brdjanin et al. [45] conducted an experiment with database professionals (135 subjects) to validate an automatic generation of a UML class diagram from a BPMN model. This experiment confirms that the automatically generated model has a higher percentage of correctness and completeness compared to a manual design from scratch. Khlif et al. [46] proposed a requirements engineering method that helps software analysts to build an Information System analysis model, which is aligned to a given BPMN model. They elaborated a set of transformation rules to generate an aligned UML analysis model (a UML use case diagram with the documentation of each use case, a system sequence diagram, and a class diagram) from an annotated BPMN model.

As a conclusion on works considered in this sub-section, we can state that there are works to generate different UML models from BPMN ones: uses cases [43] [46], class diagrams [45] [46], and sequence diagrams [46]. In our approach, we have tried to simplify the process to generate GUIs from BPMN models by combining BPMN models with UML class diagrams using stereotypes. In our approach, we do not use model to model transformations but just model to code transformations.

2.4. Graphical User Interface Generation from Other Models Different from BPMN

Next, we describe related works that deal with the generation of GUIs from models different from BPMN. In this group we have Radeke et al. [47], who presented a framework that describes how model-based approaches can be extended with patterns. The implementation of the framework is based on the User interface pattern Extensible Markup Language (UsiXML) [48], which allows describing patterns for user interfaces development. Garcia et al. [49] presented a metamodel for designing the various user interfaces of a workflow information system, which are advocated to automate business processes. The workflow model defines what processes and tasks need to be fulfilled and their possible ordering. We can see the workflow model as a framework for creating a task model suitable for designing user interfaces.

As a conclusion on works summarized in this sub-section, we can state that there exist several approaches that generate GUIs from models different from BPMN ones. The main issue with all these works is that, to generate interfaces, the analyst must learn the particularities of other models in addition to BPMN. On the contrary, with our proposal we aim to generate user interfaces only with BPMN models and a UML class diagram, no additional models to represent interaction characteristics are required.

2.5. Measures of Effort, Accuracy, and Satisfaction in Other Empirical Experiments with MDD

Next, we describe experiments that measure effort, accuracy, and satisfaction in the system generation through the MDD paradigm [8] [9]. One of these works is the approach of Panach et al. [50], which consists in an experiment on a small set of classes using subjects to compare the quality, effort, productivity, and satisfaction of MDD versus a traditional method. Krogmann et al. [13] performed an experiment with students in two projects. One project consisted of a team of 11 subjects who developed a system through a traditional method; while in the other project one subject developed another system using MDD. The only studied variable is effort. They found that development effort for a system with MDD is 11% lower than for a traditional method. Notice importantly that time is not measured, it is estimated by developers. Mellegard et al. [51] performed a case study to compare whether developers expend more effort on modeling in MDD than in a traditional method. They interviewed three project managers and measured effort spent on the development of artefacts. Results show that effort spent on modeling is similar using MDD than using a traditional method. Since MDD automatically generates code, the authors deduce that MDD-compliant methods should always be more efficient than traditional methods. Heijstek et al. [52], conducted a case study to evaluate the effort saved by using MDD in the industry. One developer, one lead developer, two project leaders and one estimation and measurement officer were interviewed. The authors studied a project to develop a system for a large financial institution. The study focuses on model size, model complexity, model quality, and effort to build the models. Results show that, for the project under study: (1) large and complex models built with MDD change more often but do not necessarily contain more defects than smaller models; (2) MDD achieves better results with regard to effort, quality, and development complexity. The research also reports on the subjective opinions of development team members about benefits of MDD: increase in productivity, consistent implementation, and improvement of the overall quality.

As a conclusion on the group of works considered in this sub-section, we can state that there are several works based on experiments that deal with the same variables that we use in our design. Table 1 reports the full list of related works we considered in this section, classifying them into four categories by their Scope and summarizing their Goal, Results, and Limitations. In the first category, "Graphical User Interfaces generation from BPMN model", existing approaches to generate GUIs are based on specific models to represent interaction that are complemented with BPMN models. So, analysts must learn these interaction models that are specific of each approach since there is not a standard. In our approach, we opt for using only standard models such as BPMN and UML class diagrams, which are widely used. This way, analysts do not need to learn new models, just stereotypes. In the second category, "BPMN Extensions", there are works that use extensions of BPMN to capture requirements of business processes. These works are aligned with our contribution, since we aim to extend BPMN through stereotypes. The main difference between our contribution and previous works is that none of the previous works can generate GUIs automatically, since those extensions have different goals and generate other parts of the system (functionality and persistency). In the third category, "Generation of other models from BPMN models", we find works that transform BPMN models into other models, such as UML use cases and class diagrams, among others. Our approach is based on model to code transformations, there are not model to model transformations in order to generate GUIs as straight forward as possible. In the fourth category, "Graphical User Interface Generation from Other Models Different from BPMN" we see experiments that validate the proposals through the variables effort, accuracy, and satisfaction. These variables are widely used in the empirical software engineering field, so we use them as variables to validate our approach.

Table 1. Summary of related works.

Author	Scope	Goal	Results	Limitations
Brambilla et al [25]	Graphical User Interfaces Generation from BPMN models	A methodology to design processes for Social BPMN	Extended BPMN model with social features	Proposed a new model that requires learning
Brambilla et al [26]		An MDD method that generates applications from BPMN models	An extended BPMN model	Proposed a new model that requires learning
Han et al. [27]		An approach for the derivation of GUIS from BPMN models	The model is specified using an extended version of BPMN	Proposed an extension that requires learning
Sousa et al. [28]		An approach to correlate a business process to GUIs	Associations between process and user interfaces	Only for organizations driven by processes
Yongchareon et al. [29]		A model-based method to generate GUIs from processes	A User Interface Flow (UIF) model to reflect processes	Proposed tool is semi-automatic
Yonchareon et al [30]		A framework for deriving UIF models for the semi-automatic creation of user interfaces	The UIF model is with business process, user interfaces, and user roles	There is a direct relationship with tasks artefacts
Boucheligua et al [31]		An approach that shows a set of model transformations	Transformations to use BPMN for modeling interactions	Proposes to learn a new task model
Abouzid et al [33]	BPMN Extensions	A set of BPMN extensions to improve business process in manufacturing	An extension of BPMN models for manufacturing	This proposal is only for manufacturing models
Decker et al [34]		BPMN extensions that broaden the applicability of BPMN.	The BPMN extensions use interaction patterns	The study is based on few interactions patterns
Rodríguez et al [36]		Integrate security requirements through process modeling.	Support to security requirements from their own perspective	The extensions are only for security requirements
Stroppi et al [37]		An extension of the BPMN metamodel to support resource perspective requirements	The extension allows defining resource structure requirements	Proposes to learn two new models
Braun et al [39]		An extension that focuses on domain analysis	An extension model transformed to BPEL	The extensions are only for medical requirements
Zarour et al [42]		A BPMN extension to specify security, compliance, cost, and performance	Four new graphic forms are added as artefacts	Define new artefacts only for capturing non functional requirements
Cruz et al [43]	Generation of other models from BPMN models	A model-driven approach to support the construction of UML use cases	Construction of use cases and the corresponding descriptions	The study is based on few rules of transformation
BRdjanin et al [45]		Validate an automatic generation of a UML class diagram from a BPMN model	The automatically generated model has a high percentage of correctness and completeness	Proposed tool is semi-automatic
Khlif et al [46]		A requirements engineering method	Transformation rules to generate UML use cases, sequence, and class diagrams	It does not have an experimental evaluation to evaluate the rules
Radeke et al [47]	GUI generation from models different from BPMN	A framework that describes how model-based can be extended with patterns	The framework is based on user interfaces pattern extensible markup language (UsiXML)	The study is based on few patterns
Garcia et al [49]		A systematic way to design user interfaces for workflows	A metamodel for designing user interfaces of a workflow.	A new task model from workflow to learn
Panach et al [50]	Measures of Effort, Accuracy, and Satisfaction in other empirical experiments with MDD	Experiment to compare quality, effort, productivity and satisfaction	MDD vs traditional method, MDD had better results	The study was conducted with few subjects
Krogmann et al [13]		A case study to analyze effort in code-centric and model-driven	MDD had better results in effort than with a traditional method.	The study was developed with few subjects
Mellegard et al [51]		A case study to compare the use of MDD and a traditional method	More effort in MDD than in traditional method	This study is based on automobile process
Heijstek et al [52]		A study to evaluate effort, quality and productivity using MDD	MDD had better results in effort, quality and productivity	The results are not generalizable

3 GUIs from BPMN models

This section describes the two methods to develop GUIs from BPMN models that we compare in our family of experiments. The first method uses standard BPMN models and manual generation of GUIs from them. The second uses the approach presented in [4], which is based on BPMN models extended with stereotypes and transformation rules to automatically generate GUIs code from BPMN models. We refer to the first and second method as to the Non-stereotyped and Stereotyped method, respectively. In the following of this section, we describe in detail each of the two methods

3.1 Non-Stereotyped Method

The non-stereotyped method consists in generating GUIs from standard BPMN models manually, without any rule or guide, just intuitively. Next, we describe the steps of this method:

- Starting from a list of requirements, the analyst has to draw the BPMN model with the requirements the system has to support. This model aims to represent how the company works, highlighting the company behavior.
- Taking as input the BPMN model drawn in the previous step and the list of requirements, the analyst has to design the GUIs of the system. Note that BPMN only represents processes, so the analyst has to complement the BPMN model with the list of requirements in order to know how to design the user interface. In our proposal, we focus on non-functional GUIs, so, how to include functionality in the system from BPMN models is out of the scope of our work.

At the end of the method, we have the final GUIs. Note that, above all, using this method has only sense in systems with many processes, whose complexity requires the use of a BPMN model [3]. In other cases, the generation of GUIs can be done directly from requirements without building the BPMN model.

3.2 Stereotyped Method

The stereotyped method consists in generating GUIs from BPMN models that have been enhanced with stereotypes. These stereotypes aim to specify GUIs characteristics unambiguously. Some of the stereotypes are based on a UML class diagram representing the information dealt with in each task of the BPMN model. The stereotyped method aims to use the same model to represent both process and interaction, instead of using an additional specific model to represent GUIs. Note that we do not aim to define a new flexible model to generate GUIs; the target is profiting from the effort to build the BPMN model to generate interfaces automatically from it, following the MDD paradigm. Usually, the person that builds BPMN models is not the same person that designs the GUI. This could lead to a gap between what is modeled in the BPMN models and what it is designed in the GUI, which the stereotyped method aims to solve. Next, we describe the steps of the stereotyped method. More details can be found in [4] and [53].

- Starting from a list of requirements, the analyst has to draw the BPMN model with the processes the system has to support. This step is the same as the first step of the non-stereotyped method.
- The analyst includes stereotypes in the BPMN model to define how each process will be displayed in a GUI. A plugin for Visual Paradigm v. 15.0 supporting all the proposed stereotypes can be used for this purpose [6].
- Once the extended BPMN model is finished, it is saved in a XML file. This file is the input for a model to code transformer that generates HTML5 GUIs by interpreting the enhanced BPMN model (see http://hci.dsic.upv.es/bpmn/). Note that this generation is automatic, according to the characteristics specified in the model.

At the end of the method, we have the final GUIs in HTML5, such a way they can be included in a software development project. In the same way as in the non-stereotyped method, our proposal only generates non-functional GUIs, and it is beneficial in complex systems with many processes. Next, we summarize the different stereotypes that compose our proposal and the associated concrete transformation rule to generate GUIs from them.

3.2.1 Overview of the Transformation Rules Used in the Experiment and Stereotypes

This section shows a summary of the transformation rules proposed to build GUIs from BPMN models through the stereotyped method. More details can be found in [4] and [53]. These rules were abstracted from 12 existing BIZAGI projects [5] related to various domains of human activities such as administrative, sales, management, and education software. The identification of the transformation rules was based on the classification of BPMN primitives into five BPMN patterns [54]: sequence, exclusive decision, synchronization, implicit decision, and structure union synchronization. We abstracted 14 transformation rules from the 12 BIZAGI projects. The process to define the rules and their justification is out of the scope of this paper and can be found in [4] [53].

From the 14 rules, we focus our experiment on 5 rules that are the most widely used in BPMN projects [54]: **R0** and **R1** (which are generic rules that apply to any BPMN pattern), **R2** (which belongs to the sequence pattern), **R6** (which belongs to the exclusive

decision pattern), and **R9** (which belongs to the synchronization pattern). The rest of rules focus on other patterns, such as implicit decision or union synchronization, that would have required a higher level of experience in the subjects and whose use in BPMN models is less frequent. So, in order to schedule the experiment in a reasonable time (2 hours), we focused the experiment on the most frequently used rules.

These rules have several alternatives to generate the GUIs, so we need an unambiguous semantic to specify which alternative will be used. For this aim, we extended the BPMN model with new stereotypes that allow to specify what alternative will be applied in each case. These stereotypes are published in [4] [53]. The extended BPMN model is complemented with a UML class diagram, since part of the interface depends on the persistency model. A mapping is required between BPMN tasks and UML class diagrams. This mapping is represented through the use of BPMN packages. Next, we describe a summary of the 5 rules used in the experiment to generate code from a stereotyped BPMN model in collaboration with a UML class diagram.

R0: From the attributes of a UML class diagram we can extract the input fields of a to be generated form. In particular: (1) a class attribute with a stereotype << Text >>, an input field accepting any type of textual string (Text box) has to be generated; (2) the stereotypes << Combo >> and << List >> attached to a class attribute represent an input field whose value must be chosen from a closed list (a Combo box and a List box, respectively). It is not possible to generate the items that compose these widgets since this information is not represented in the class diagram; (3) to generate an input field that accepts a boolean, the class attribute from which the input field has to be generated is annotated with the stereotype << Radio >> and << Check >> (for a Radio button and a Check box, respectively).

R1: User type tasks of a BPMN model are usually transformed into forms in GUIs. The input fields of the form are extracted from the attributes of the UML classes that are involved in the user tasks: each attribute generates an input field. After it is applied, R1 is complemented with R0. The proposed stereotype to generate a form from a package including a user task is << Form >>.

R2: User type tasks with a dependency with other tasks in the same lane are mapped onto a GUI providing the end user with some guidance to carry out these tasks, with three alternatives: (1) a Wizard, when the navigation through the different forms of each user task is ordered in a sequence with the possibility to go forward and backward; we annotate a package with the stereotype << Wizard >> to indicate that a Wizard has to be generated from that package; (2) a Tabbed dialog box where each tab contains the form corresponding to each user task, when there is no particular order between them; the stereotype used to generate a Tabbed dialog box from a package is << Tab >>; (3) a Group box, when a limited number of user tasks can be grouped in the same form; the stereotype we use to indicate that a Group box has to be generated from a package is << Group >>.

R6: The text in a BPMN exclusive decision gateway is mapped onto two alternatives: (1) a Radio button, when each alternative of the gateway represents an option (package with stereotype << Radiobutton >>); (2) a push button, when each alternative of the gateway represents an action (package with stereotype << Button >>).

R9: If after the parallel gateway there are user type tasks that are in the same lane and there is a dependency between them, interfaces to guide the user through a process of several steps are generated. The alternatives to represent the different steps are exactly the same as the alternatives explained in Rule 2: Wizard, Tabbed dialog box and Group box.

Note that the stereotype << Form >> can only be used if the package contains a single user type task. The stereotypes << Radiobutton >> and << Button >> can only be used if the package contains a gateway that belongs to exclusive decision pattern. The gateway is depicted by an empty rhombus. The stereotypes << Wizard >>, << Group >>, and << Tab >> are used in two cases: (1) when the package contains a set of user type tasks; (2) when the package contains a parallel gateway that belongs to the synchronization pattern (the gateway is depicted by a rhombus with a cross). Table 2 summarizes the 5 rules described above, indicating, for each rule, the used stereotypes, the modeling primitives to which the stereotypes are applied, and the generated GUI widgets.

Table 2 Rules, stereotypes, primitives, and GUI widgets.

Rules	Stereotypes	Primitives	GUI Widgets
R0	<< Text >>	Attributes of UML class	Text box
	<< Combo >>		Combo box
	<< List >>		List box
	<< Radio >>		Radio button
	<< Check >>		Check box
R1	<< Form >>	BPMN package (when it contains a user type task)	Form
R2, R9	<< Wizard >>	BPMN package (when it contains a set of user type tasks), or BPMN package (when it contains a parallel gateway)	Wizard
	<< Group >>		Group box
	<< Tab >>		Tabbed dialog box
R6	<< Radiobutton >>	BPMN package (when it contains exclusive decision gateway)	Radio button
	<< Button >>		Push button

3.2.2 Illustrative Example of Working with Stereotypes

This section shows an illustrative example of how to use the proposed stereotypes. We will use the Travel Request project from Bizagi [5]. The process starts when the employee boss records an employee's travel request. The boss specifies information regarding transportation and hotel reservations, and reviews the information so that the request can be approved or rejected. The boss records the reservations and the travel disbursement for the employee who is finally notified of it. Stereotypes << Text >>, << Combo >>, << List >>, << Radio >>, and << Check >> are applied only on the UML class diagram primitive named attribute. Stereotypes << Form >>, << Wizard >>, << Group >>, << Tab >>, << Radiobutton >>, and << Button >> are applied only on the component package in such a way that they affect all the primitives inside the package. Fig. 1 shows the example of BPMN model extended with our stereotypes and complemented with UML classes built for the Travel Request process described above. We built next model:

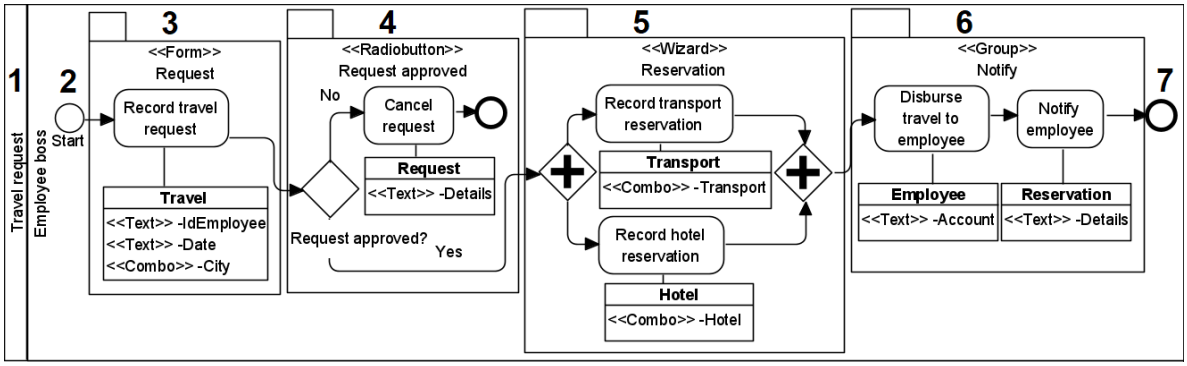


Fig. 1 Illustrative example of a BPMN model extended with stereotypes and UML classes.

Fig. 2 shows the generation of GUIs from the extended BPMN model of Fig. 1. Next, we describe each section of the figure in a sequential order. (1) We see a form that is generated from the stereotype << Form >> assigned to the BPMN package "Request" in Fig. 1. In the form, we see three input fields that are generated from the three UML class attributes associated to the "Record travel request" user task in Fig. 1. In particular, we have a text box for attributes "IdEmployee" and "Date", and a combo box for attribute "City". (2) We see a form radio button with two options (Yes/No) that is generated from the stereotype << Radiobutton >> that was assigned to the package "Request approved" in Fig. 1. We see one form "Cancel request" with the text box input field that is generated from the class attribute "Details" in Fig. 1. (3) We see a wizard that is generated from the stereotype << Wizard >> that was assigned to the package "Reservation" in Fig. 1. The fields of this wizard are generated from the class attributes associated to the tasks "Record transport reservation" and "Record hotel reservation" in Fig. 1. (4) We see a group box that is generated from the stereotype << Group >> that was assigned to the package "Notify" in Fig. 1. The fields of this group box are generated from the class attributes associated to the tasks "Disburse travel to employee" and "Notify employee" in Fig. 1.

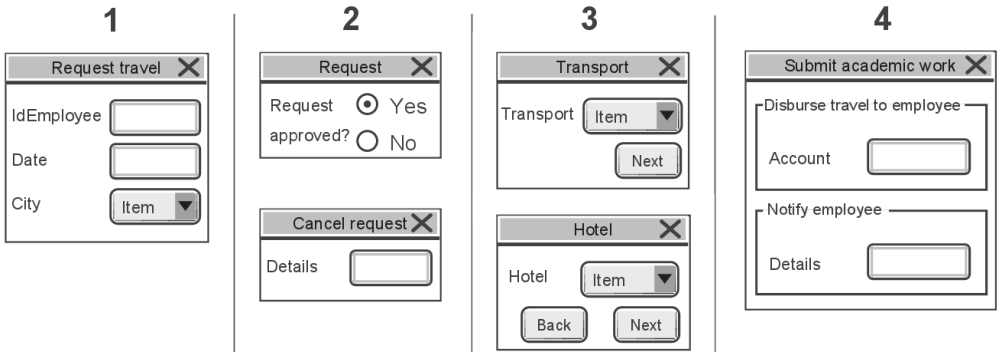


Fig. 2 Illustrative example of generating GUIs from stereotyped BPMN models.

4 Experimental Definition and Planning

The goal of the family of experiments consists in the comparison of the Non-stereotyped method versus the Stereotyped method to generate GUIs from BPMN models. The focus is on designing GUIs for a business process. The experiment is conducted from the perspective of researchers and practitioners interested in the research of generating GUIs from business process models.

The experiment locates in the context of MDD: we aim to focus the analyst's effort on modeling and rely on automatic model-to-code transformations for implementation. Our approach starts from the idea that analysts have to build BPMN models to identify the processes of the system to be developed. The target of the Stereotyped method is to profit from the BPMN models to automatically

generate interfaces too. Without the Stereotyped method, the analyst has to draw the GUI manually, apart from the BPMN models. This is why we compare the Stereotyped method versus the Non-stereotyped one in our family of experiments. Note that both methods start from a BPMN model whose initial goal is to identify the system processes.

4.1. Research Questions and Hypothesis Formulation

Next, we describe the research questions we have formulated for our study. These questions are extracted from previous works that have dealt with the advantages of a MDD method versus a traditional one, such as [12], [16], [18]. Through our family of experiments, we aim to check such advantages:

RQ1: Is subjects' effort affected by the method used to generate GUIs from BPMN models? Effort is defined as "the number of labour units required to complete a schedule activity or work breakdown structure component and is usually expressed in person-hours, person-days, or person-weeks" [24]. We measure effort as the time in minutes taken to build GUIs starting from the provided requirements. It includes the time spent in drawing the BPMN models from which GUIs are derived. To address this research question, we test the null hypothesis *H_{01}: The effort to generate GUIs using the non-stereotyped method is similar to the effort using the stereotyped method.*

RQ2: Is GUI accuracy affected by the method used to generate GUIs from BPMN models? Accuracy is defined in ISO 25000 as "the degree to which data has attributes that correctly represent the true value of the intended attribute of a concept or event in a specific context of use" [55]. We have defined sections of each experimental problem that cover each generation of GUIs. We measure accuracy as the percentage of sections of the generated GUIs which comply with the requirements. To address this research question, we test the null hypothesis *H_{02}: The accuracy of GUIs generated using the non-stereotyped method is similar to the accuracy using the stereotyped method.*

RQ3: Is subject satisfaction affected by the method used to generate GUIs from BPMN models? Satisfaction is defined as "the contentedness and positive attitudes towards the use of a product" [24]. We measure satisfaction as to how at ease developers are as they develop a system. To address this research question, we test the null hypothesis *H_{03}: The subject satisfaction to generate GUIs using the non-stereotyped method is similar to the satisfaction using the stereotyped method.*

4.2. Factors and Treatments

We now define factors and treatments to operationalize the cause of our experiment construct. Factors are independent variables with an effect on the response variables we want to understand [56]. Treatments are alternatives (levels) of a factor. In our experiment we have one factor: method to generate GUIs from BPMN models. The control treatment is the non-stereotyped method while the treatment we want to compare is the stereotyped method.

Regarding the **control treatment**, subjects had to use the non-stereotyped method to generate GUIs. This treatment starts watching a video that describes a set of requirements to consider in the system. The video consists of 4 slides with images and a few notes about the requirements. The details of the requirements were described in a voice-over of one of the experimenters throughout the slides. Considering these requirements, subjects have to build a BPMN model in Visual Paradigm 15.0. Once the BPMN model is finished, subjects have to draw the GUIs manually in a paper. These GUIs have to represent the processes expressed in the BPMN model. In this treatment, rules applied to transform BPMN models into GUIs are subjective and depend exclusively on subject's decision. Experimenters do not provide any guide to know how to transform BPMN primitives into GUI widgets.

Regarding the **treatment level**, subjects had to use the stereotyped method to generate interfaces. The beginning of this treatment is the same as the control treatment; there is a short video with slides and a voice-over describing the system requirements. After watching the video, subjects have to build the BPMN models with the stereotypes through Visual Paradigm v. 15.0. In this case, Visual Paradigm has been extended with a plugin that supports the use of all the stereotypes. Once the model is finished, subjects have to export the model to an XML file. This XML file is the input for a model to code transformer that automatically generates the GUI in HTML5. In this treatment, subjects only have to focus their efforts on building the BPMN model extended with stereotypes, while rules to generate GUI widgets are automatically applied without subject intervention.

As blocking variable, we have the **problem.** We have blocked this variable since we are not interested in studying which problem yields the best value. We studied two different problems to avoid the threat of learnability between treatments and to better justify the generalizability of results independently of a specific problem. As moderator variable we have the **course.** Moderator variable is considered as a fixed effect to aggregate the data of both replications. We have two courses: 2018/2019 and 2019/2020.

4.3. Response Variables and Metrics

Response variables are the effects studied in the experiment caused by the manipulation of factors. Our experiment has a within-subjects design (all participants are exposed to every treatment), with three response variables.

RQ1 requires a variable to measure effort. Since effort is the time to develop a system per developer [24], we measured it as the time in minutes taken by each subject since he started watching the video with the requirements until they generate GUIs (including building the BPMN model). Each subject has to write on her/his own in a paper the time when she/he starts the problem (watching the video) and the time she/he finishes (the GUI has been drawn).

RQ2 requires a variable to measure accuracy. Accuracy is measured as the percentage of sections of the generated interface which are compliant with the requirements. This metric is calculated comparing GUIs generated by the subjects versus a GUI generated by the experimenters (GUI solution). The GUI solution was designed by the experimenters working in collaboration, ensuring that it supports all the requirements. The experimenters first built the BPMN model to express the requirements in processes notation. Next, applying the transformation rules explained in Section 3.2.1, they generated the GUI solution. The GUI solution is divided into several sections, each section being the portion of the GUI that has been generated with a specific transformation rule. This division is done by the experimenters and allows to maintain the traceability between requirements and the interface generation, since our goal is to analyze the whole software design process, from requirements elicitation to GUI design, not just the final GUIs. Accuracy is calculated as the percentage of sections of the subjects' GUIs that matches with the sections of the GUI solution. This way we are measuring the accuracy obtained in the final result (GUIs) but also considering the requirements that are behind the GUIs. Note that the BPMN model has been built from the requirements and the GUIs have been designed form the BPMN model. The result of comparing a section of the GUI of a subject with the corresponding section of the GUI solution can be 1 or 0. 1 means that both sections have a combination of widgets that could be generated through the same transformation rule. 0 means the contrary.

Note that each rule may have several alternatives of GUI widgets (see Table 1) and all of them should be considered for the comparison since all of them involve a good value for accuracy according to our proposal. For example, in a section of the GUI solution we can have a wizard, while subjects, for the same section, can have a tabbed dialog box. Even though the two sections would be different, the transformation rule behind them is the same (R2). So, the accuracy for this part of the GUI would be 1. The accuracy of the whole system is the average of accuracy obtained in all its sections.

Eq. 1 shows the formula used to calculate the accuracy. For example, if the GUI generated by the subject has three sections and there are only 2 sections that agree with alternatives of our transformation rules, accuracy is: 2/3 x 100% = 66%, this means that the generated GUI by the subject is compliant with the elicitation requirements in 66%.

$$Accuracy = \frac{(Number\ of\ sections\ successfully\ completed)\ x\ 100\%}{Number\ of\ sections\ of\ the\ experimental\ problem} \quad (1)$$

RQ3 requires a response variable to measure satisfaction (the positive attitude towards the use of the development method [24]) both in Stereotyped and in Non-Stereotyped. Satisfaction is measured on-line using a 5-point Likert scale questionnaire based on the framework developed by Moody [57]. Moody defined a framework (based on the work of Lindland [58]) in order to evaluate satisfaction in terms of Perceived Ease to Use (PEOU), Perceived Usefulness (PU) and Intention to Use (ITU). This framework has been previously validated and is widely used [57]. The possible answers for each statement in the questionnaire of PEOU, PU and ITU are: Totally disagree, Fairly disagree, Neutral, Fairly agree and Totally agree. We provided a numerical value to each statement from 1 (Totally disagree) to 5 (Totally agree) [59]. We defined six questions to measure PEOU; the metric was calculated adding the numerical values of the answers and classifying into a rank of five possible values: Rank 1-6: Totally disagree, Rank 7-12: Fairly disagree, Rank 13-18: Neutral, Rank 19-24: Fairly agree, Rank 25-30: Totally agree. For example, if a subject answers 5 questions with Totally agree and 1 questions with Neutral in PU, the result of this metric will be 28 (Totally agree). We defined eight questions to measure PU; the metric was calculated adding the numerical values of the answers that each subject filled in through the eight questions, the result of this addition is classified into a rank with the five possible options: Rank 1-8: Totally disagree, Rank 9-16: Fairly disagree, Rank 17-24: Neutral, Rank 25-32: Fairly agree, Rank 33-40: Totally agree. We defined two questions to measure ITU, the metric was calculated adding numerical values of the answers and classifying the result into a rank of two possible values: Rank 1-2: Totally disagree, Rank 3-4: Fairly disagree, Rank 5-6: Neutral, Rank 7-8: Fairly agree, Rank 9-10: Totally agree. The questionnaire used for the Non-stereotyped method is shown in Appendix 1 and the one for the Stereotyped method is shown in Appendix 2. Table 3 shows a summary of the research questions, hypotheses, response variables, and metrics used to test the null hypotheses.

Table 3 Summary of RQs, hypothesis, response variables and metrics.

RQs	Hypotheses	Response Variables	Metrics
RQ1	*H01*	Effort	Time
RQ2	*H02*	Accuracy	Percentage of sections in the GUI that matches with requirements
RQ3	*H03*	Satisfaction	PU, PEOU, and ITU

4.4. Experimental Problems

The experiment has two experimental problems of similar complexity. These problems are small to avoid the fatigue of the subjects and limit the experiment duration to 2 hours. Even though this simplicity, the problems contain the 5 rules described in Section 3.2.1. Next, we describe the problems:

Experimental Problem 1: A system to submit an academic work. This problem aims to design the GUI of a system for submitting academic works to teachers. The process starts with Section 1, where the student logs in to enter the virtual classroom through user and password. Next, in Section 2, the student goes to submit an academic work, where he/she records the following data: title of the work, file to submit, and comments. Finally, in Section 3, the teacher reviews the academic works submitted. If the academic work is approved, the teacher will record the mark for the work. If the academic work is failed, the teacher records the corrections to improve the work.

Experimental Problem 2: A system to buy an on-line product. This problem aims to design a GUI for recording the online purchase of a product. The process starts with Section 1, where the user enters the following data: Email, First name, Surname, Product (only one), and Quantity. Next, in Section 2, the user has to perform three tasks in any order: (1) enter shipping address providing the following data: Address, City, Country, Postal code, and Mobile phone; (2) choose shipping option among three: free one week, free three days, and now. (3) provide data of the credit card: Card type (among 3 options), Card number, and CVV. The amount to pay is finally shown.

Next, we show the experimenters' solution for the GUI of each experimental problem. Fig. 3 shows the BPMN model obtained with the non-stereotyped method for the experimental problem 1 with the three sections that compose it. In the *lane* Student, Section 1 shows the *user type task* - User login. Next, Section 2 shows the *user task* Submit an academic work. Finally, in the *lane* Teacher, Section 3 shows a *gateway* of exclusive decision pattern Review academic work, with two options Approved and Failed, and two *user tasks*. Each section in Fig. 3 should generate a GUI similar to the GUI in Fig. 5 (experimenters' solution).

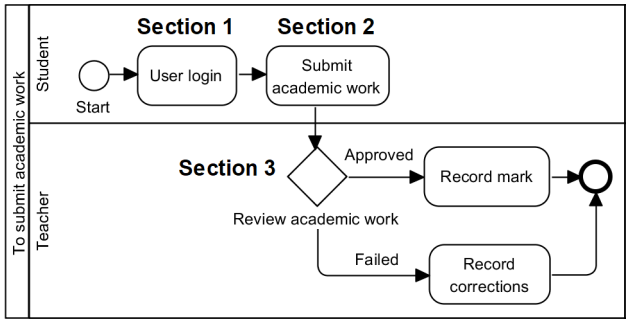


Fig. 3 BPMN model for experimental problem 1 obtained with the non-stereotyped method.

Fig. 4 shows the same BPMN model of Fig. 3 but including stereotypes (used in the Stereotyped method). In the *lane* Student, Section 1 shows the stereotype << Form >> to generate a form with the attributes of the class User. These attributes will be displayed in a text box since they are defined with the stereotype << Text >>. Next, Section 2 shows the stereotype << Form >>, which means that we need a form to fill in the attributes of the class Work, whose attributes are annotated with the stereotype << Text >>. For Section 1 and Section 2, subjects can use the stereotypes << Wizard >> to generate a wizard or << Group >> to generate group box or << Tab >> to generate tabbed dialog box in the two user tasks. In the *lane* Teacher, Section 3 shows the stereotype << Radiobutton >> to generate two radio buttons (Approved and Failed). If the work is Approved, the flow continues through the *user task* (Record mark) to generate a form with attributes of the class Work with stereotype << Text >>. If it Fails, the flow continues through a *user task* (Record corrections) to generate a form with attributes of the class Corrections, whose attributes have stereotype << Text >>. We consider that in Section 3 subjects can use the stereotype << Button >> to generate push buttons according to rule R6. Each section of Fig. 4 should generate a GUI similar to the GUI in Fig. 5 (experimenters' solution).

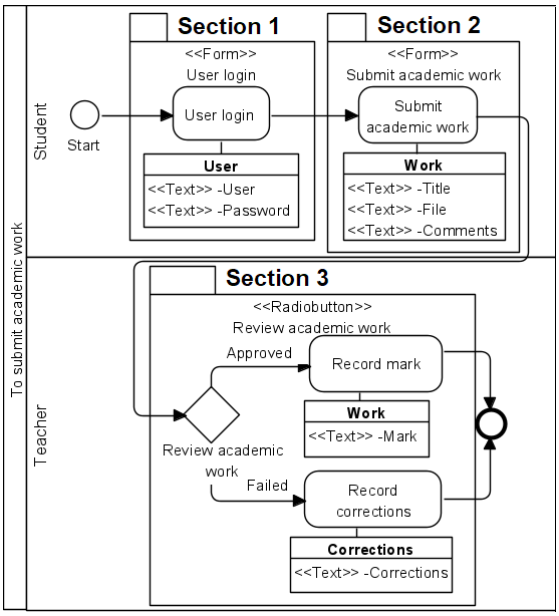


Fig. 4 BPMN model for experimental problem 1 obtained with the Stereotyped method.

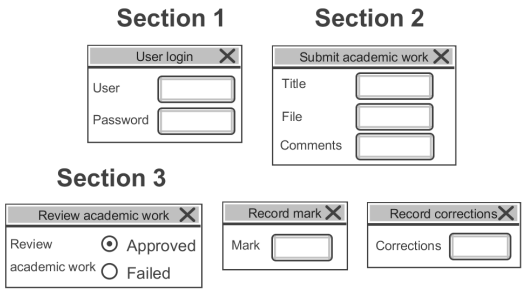


Fig. 5 Experimenters' solution for the GUIs of experimental problem 1.

Fig. 6 shows the BPMN model obtained with the non-stereotyped method for the experimental problem 2. The model includes two sections. In the *Lane* User, Section 1 shows a *user task* (Enter customer and product data). Finally, Section 2 shows a *parallel gateway* of synchronization pattern, with three *user tasks* (Record shipping address, Choose shipping option, and Choose method of payment). Each section of the model in Fig. 6 should generate a GUI similar to the GUI in Fig. 8 (experimenters' solution).

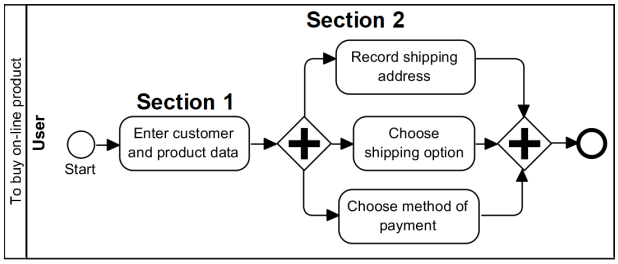


Fig. 6 BPMN model for experimental problem 2 obtained with the non-stereotyped method.

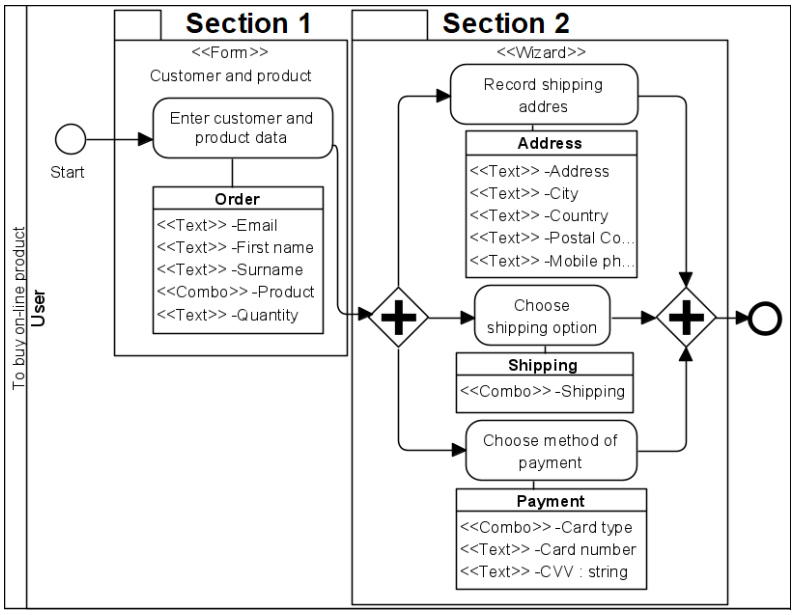


Fig. 7 BPMN model for experimental problem 2 obtained with the Stereotyped method.

Fig. 7 shows the BPMN model obtained for the experimental problem 2 using the stereotyped method. In the *lane* User, Section 1 shows the stereotype << Form >> to generate a form with attributes of the class Order with the stereotypes << Text >> to generate text boxes, and a stereotype << Combo >> to generate a combo box for selecting one of the possible product options. Next, Section 2 shows the stereotype << Wizard >> to generate a wizard; this wizard includes three forms, each corresponding to one of the three user tasks included in package Section 2 of Fig. 7. The three forms are: (i) a form with attributes of the class Address with stereotypes << Text >>; (ii) a form with attributes of the class Shipping in which the stereotype << Combo >> is used to generate a combo box for selecting one of the possible shipping options; (iii) a form with attributes of the class Payment with stereotype << Combo >> and << Text >>. We consider that in Section 2 it is valid that subjects use the stereotypes << Group >> to generate group box or << Tab >> to generate a tabbed dialog box according to rule R9. Each section of Fig. 7 should generate a GUI similar to the GUI in Fig. 8 (experimenters' solution).

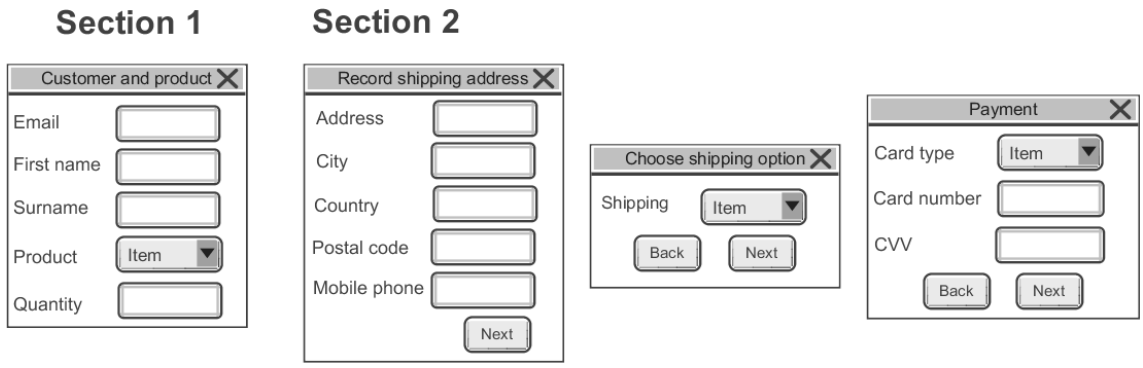


Fig. 8 Experimenters' solution GUIs of the experimental problem 2.

Table 4 shows a summary of the GUIs that compose the experimenters' solution and the stereotypes used in the BPMN model for the stereotyped method in both experimental problems. For each problem we summarize the method, sections, stereotypes, and GUI widgets of the experimenters' solution. Note that the GUIs associated to a given experimental problem are the same independently of the treatment, since differences between treatments focus on how GUIs are designed but final GUIs must be the same. Note also that we have only stereotypes and rules when the method is Stereotyped, otherwise GUIs are generated without rules. We have addressed the 5 rules of Table 2 in both experimental problems. The experimental problem 1 uses rules R0, R1, R2 and R6, while the experimental problem 2 uses rules R0, R1, and R9.

Table 4 Summary of experimenters' solution GUIs for the two experimental problems.

Experimental problem	Method	Section	Stereotypes		GUI widgets	Rules	
1	Non-stereotyped	1	-		Form	-	
		2	-		Form	-	
		3	-		Radio button or Push button, Text box	-	
	Stereotyped	1	<<Form>>	<<Wizard>> or <<Group>> or <<Tab>>,	Form	R1	R2
		2	<<Form>>		Form	R1	
		3	<<Radiobutton>> or <<Button>>, <<Text>>		Radio button or Push button, Text box	R6, R0	
2	Non-stereotyped	1	-		Form	-	
		2	-		Wizard or Group box or Tabbed dialog box, Text box, Combo box	-	
	Stereotyped	1	<<Form>>		Form	R1	
		2	<<Wizard>> or <<Group>> or <<Tab>>, <<Text>>, <<Combo>>		Wizard or Group box or Tabbed dialog box, Text box, Combo box	R9, R0	

4.5. Experimental Procedure

Before starting the experiment, we used a learning activity to train the subjects and ensure that their knowledge was enough to participate in the experiment:

- **Introduction to the Non-stereotyped and the Stereotyped method.** Subjects had to follow a tutorial about the Non-stereotyped and the Stereotyped method. The tutorial consists of a document illustrating the primitives of BPMN and our stereotypes, as well as a short video explaining how to draw the BPMN models (both stereotyped and non-stereotyped) in Visual Paradigm v. 15.0. The tutorial was delivered to subjects a week before the experiment, so that they could learn and train before the experiment. As a result of the training, subjects had to build a small BPMN model both with the non-stereotyped method and with the stereotyped one. The elaboration of these training models was a prerequisite to becoming a subject. Only subjects whose training models were done properly were recruited as subjects for the experiment.
- **Filling a training test.** Apart from the training models previously explained, the subject's knowledge of BPMN was evaluated through a test before conducting the experiment. The test consisted of 10 questions (5 questions on the non-stereotyped method and 5 questions on the stereotyped one). Each question had 4 alternatives, which were shown by means of pictures. There was only

one possible correct answer, and each correct answer was computed as one point, so possible points were between 0 (no correct answers) and 10 (all answers are correct). We assumed that subjects that got more than 5 points were capable of participating in the experiment. In case we had subjects with a score between zero and five, these were removed from the experiment. Table 5shows the marks of the 59 subjects that were evaluated (28 subjects in the course 2018/2019 and 31 subjects in the course 2019/2020). Note that only 5 were removed from the course 2018/2019. Therefore, in the experiment participated 54 subjects.

Table 5 Results of the entrance test.

Scores	# subjects for course 2018/2019	# subjects for course 2019/2020
<=5	5 (removed)	0
6	0	0
7	1	3
8	2	7
9	4	9
10	16	12
TOTAL	**23**	**31**

The procedure of the experiment included three steps and lasted two hours:

1. **Filling the demographic questionnaire.** Subjects filled in a demographic questionnaire (fields such as email, first name, last name, gender, age, and knowledge of BPMN model, UML class diagram, and GUIs were included) before running the experiment to identify their background. Each subject signed a consent form.
2. **Solving the experimental problems.** Subjects had to solve both experimental problems. Subjects were instructed to behave naturally, build BPMN models and generate the GUIs according to the treatment. We applied a different treatment to each problem per subject. The process of drawing GUIs lasted sixty minutes in total, maximum thirty minutes for each experimental problem.
3. **Filling the post-test questionnaire.** After finishing each experimental problem, subjects filled in an on-line questionnaire about satisfaction in terms of PEOU, PU, and ITU based on the treatment used with such a problem. Filling this questionnaire spent 10 minutes per problem.

After conducting the experiment, one of the experimenters analyzed the data. He compared the sections of the GUI proposed by each subject versus the experimenters' solution (GUI solution) to calculate the accuracy. The experimenter also extracted the effort per treatment reading the papers where each subject wrote down her/his times. The results of satisfaction were extracted directly from the questionnaires.

4.6. Experimental Design

This section describes the design of the family of experiments, which is a crossover design [60]. Depending on the combination of treatments and the blocking variable, we have four different profiles for subjects. Each subject is assigned to each profile randomly, ensuring that the sample size in each profile is balanced. We assigned a profile to a subject depending on how they were located in the classroom. Next we describe the characteristics of each profile:

- **Profile 1**: Subjects begin to develop experimental problem 1 with the non-stereotyped method. Then, subjects develop experimental problem 2 with the stereotyped method.
- **Profile 2**: Subjects begin to develop experimental problem 2 with the non-stereotyped method. Then, subjects develop experimental problem 1 with the stereotyped method.
- **Profile 3**: Subjects begin to develop experimental problem 1 with the stereotyped method. Then, subjects develop experimental problem 2 with the non-stereotyped method.
- **Profile 4**: Subjects begin to develop experimental problem 2 with the stereotyped method. Then, subjects develop experimental problem 1 with the non-stereotyped method.

Table 6 shows the design used in the experiment considering the different profiles.

Table 6 Design used in the experiment.

	Non-stereotyped	**Stereotyped**
Profile 1	Experimental Problem 1	Experimental Problem 2
Profile 2	Experimental Problem 2	Experimental Problem 1
	Stereotyped	**Non-stereotyped**
Profile 3	Experimental Problem 1	Experimental Problem 2
Profile 4	Experimental Problem 2	Experimental Problem 1

4.7. Experimental Subjects

Subjects are undergraduate students in computer science who are considered as representative end users. The total number of subjects involved in the experiment is 54, 23 in a first replication (18 male vs 5 female, 16 aged in the range 17-20, 3 in 21-25, 3 in 26-30, and 1 in 31-35, M= 21.13, SD = 3.92) and 31 in a second (29 male vs 2 female, 8 aged in the range 17-20, 21 in 21-25, and 2 in 26-30, M= 22.97, SD = 2.53). All of them were recruited from the course of Software Engineering in the computer science degree of the University of Valencia (Spain) in the academic years 2018/2019 and 2019/2020. Note that both groups have knowledge in interface programming and conceptual modeling (see Table 7). Two of the experimenters of this article were teachers of the course. Most subjects had not a high knowledge of BPMN prior to taking the course, even though a large number of subjects had a medium knowledge of UML class diagram and GUIs. In order to ensure that all the subjects had enough knowledge to participate in the experiment, subjects had to design a small BPMN model as homework before the experiment and to submit it to experimenters, who evaluated the submission. Moreover, before the experiment, subjects had to pass a test about the non-stereotyped method and the stereotyped one. They had to study BPMN at home through a short BPMN report of 1 page prepared by the experimenters that summarized primitives used in the experiment. Only subjects that submitted the small BPMN models and who passed the exam were recruited as subjects. All of them participated voluntarily in the experiment and we applied both methods to all of them.

Table 7 Knowledge of BPMN models, UML class diagram and GUIs of course 2018/2019 and course 2019/2020.

Knowledge of	**Course 2018/2019**				**Course 2019/2020**			
	None	**Low**	**Medium**	**High**	**None**	**Low**	**Medium**	**High**
BPMN models	13%	48%	30%	9%	39%	58%	3%	0%
UML class diagram	0%	30%	57%	13%	10%	22%	65%	3%
Graphical user interfaces	0%	17%	52%	31%	6%	42%	52%	0%

4.8. Data Analysis

The statistical analysis of data collected with the experiment was done applying the mixed model statistical test [61]. The blocking variable, Problem, is introduced as a covariate in the mixed model test. Moreover, we used Course as a moderator variable that is introduced to aggregate both replications to conduct the analysis. The assumption for applying the mixed model is normality of residuals. The normality of residuals can be tested with Saphiro-Wilk test applied to the residuals automatically calculated during the application of the mixed model test [62]. The p-value of the Mixed Model shows whether or not there is a significant difference between treatments of each factor. If the p-value is less than 0.05, we assume that there are significant differences between treatments. For those variables with significant differences among treatments, we use the non-parametric test called Cohen's d [63] to calculate the effect size; this calculus yields the magnitude of such difference. Cohen's d is defined as the difference between two means divided by a standard deviation of the data. According to Cohen [63], the meaning of the effect size is as follows: values greater than 0.8 indicate a large effect; values from 0.79 to 0.5 indicate a moderate effect; values from 0.49 to 0.2 indicate a small effect. We use this technique for the three response variables: Effort, Accuracy, and Satisfaction. Table 8 shows a summary of the statistical methods and ranks.

Table 8 Summary of statitical methods and ranks

Concepts	Rank	The meaning is:
Mixed model	P-value is less than 0.05	There are significant differences between treatments
Effect size	Cohen's > 0.8	Large effect
	Cohen's between 0.79 and 0.5	Moderate effect
	Cohen's between 0.49 and 0.2	Small effect

The power of any statistical test is defined as the probability of rejecting a false null hypothesis. Statistical power is inversely related to beta or the probability of making a Type II error. In short, power = 1 – β. Power in software engineering experiments tends to be low. Dyba et al. [64] report values of 0.39 for medium power and 0.63 for large power. Low values of power mean that non-significant results may involve accepting null hypotheses when they are false. According to G*Power [65], for a moderate effect size (0.5) in a within-subjects design, we need a sample size over 16 subjects to get good values of power (0.96). In our experiment, we have 23 subjects in course 2018/2019 and 31 subjects in course 2019/2020, so we have enough power in each replication. However, adding both replications, we have 54 subjects, which implies a high statistical power in the aggregation. This means that we have enough sample size to reject null hypotheses in case there are possible differences between treatments, both in each replication, independently and especially in the aggregation of the replications.

5 Results

This section reports the quantitative results of our experiment in order to address the research questions (Raw data can be seen in Appendix 3). All analyses have been performed using IBM SPSS v. 20. First, we analyze the results for course 2018/2019 and course 2019/2020 independently and then the aggregation of both courses. Table 9shows the p-values of each response variable for the method

to generate GUIs from BPMN models using stereotypes compared to the non-stereotyped method. P-values show significant differences in the course 2019/2020 for accuracy with a p-value of 0.018; these differences are moderate, according to the effect size of 0.65. PU has a p-value of 0.017 with differences that are small according to the effect size of 0.46. ITU has a p-value of 0.002 with differences that are moderate according to the effect size of 0.52. Fig. 9 shows the box-and-whisker plots for the significant variables of the course 2019/2020. Note that the line between both boxes means the average in each box. We can appreciate that the first quartile is worse for the Non-stereotyped method in the three variables Accuracy, PU, and ITU. Medians between treatments are similar in the three plots, which means that existing significant differences are not very large. The Stereotyped treatment yields better results compared to the Non-stereotyped one.

Table 9 P-values for the Stereotyped method compared to the Non-stereotyped one.

	Effort	**Accuracy**	**Satisfaction**		
			PEOU	**PU**	**ITU**
Course 2018/2019	0.096	0.476	0.056	0.818	0.586
Course 2019/2020	0.057	**0.018** effect size: 0.65	0.687	**0.017** effect size: 0.46	**0.002** effect size: 0.52

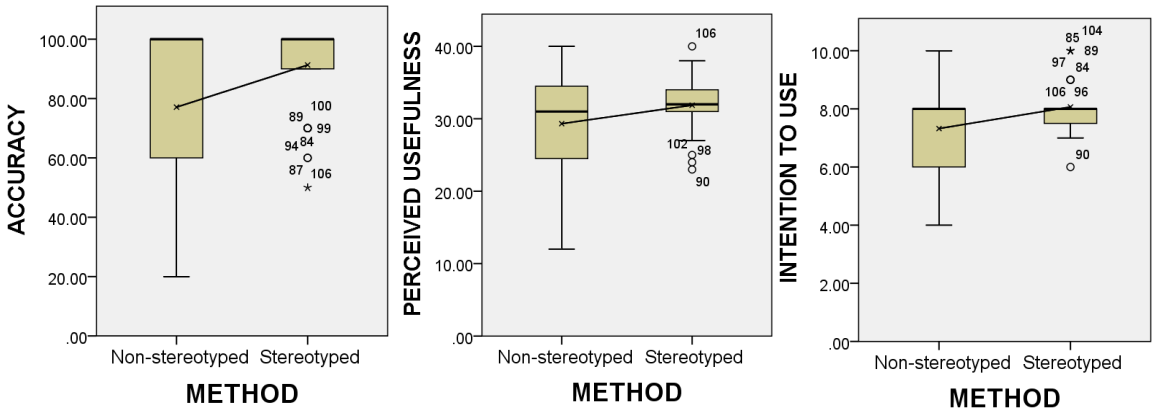
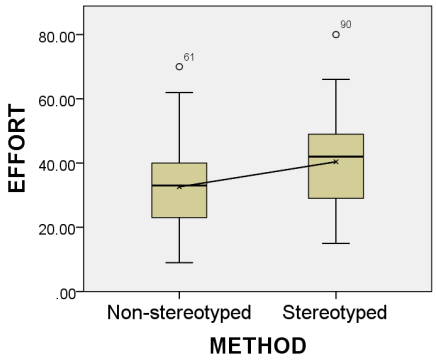


Fig. 9 (a) Box-and-whisker plot for accuracy (b) Box-and-whisker plot for PU (c) Box-and-whisker plot for ITU.

We conclude that we have only some differences in one replication (course 2019/2020) for Accuracy, PU and ITU, and that these differences are not so large. So, in order to improve the statistical power and to avoid accepting the null hypothesis when they must be rejected, we opt for aggregating the data of both replications through the moderator variable "Course". Next, we analyze in detail each response variable after the aggregation of both courses (2018/2019 and 2019/2020).

5.1. Effort

Effort is measured as the time in minutes taken by each subject to develop GUIs (the less time in minutes spent in the development, the best effort). Fig. 10 shows the box-and-whisker plot comparing effort (in minutes). The first quartile and medians are similar, but there is a small difference with the third quartile, where the non-stereotyped yields better results. The median for effort using the non-stereotyped method is lower than for the Stereotyped method, which means that the time working with the Stereotyped method is higher than with the Non-stereotyped one. The aim of our experimental investigation is to identify whether there are differences or not in the effort working with the Non-stereotyped method or with the Stereotyped one (*H01*). The p-value for the method to generate GUIs from BPMN models is 0.013, which means that there are significant differences between the two methods (Non-stereotyped and Stereotyped), these differences are moderate according to the effect size of 0.575. The interaction Method*Problem is not significant with a p-value of 0.378, which means that the combination of problem and method does not influence effort. Note that we are not interested in differences between problems, just in the interaction Problem*Method to check that the type of problem is not affecting the results of the method. We also identify significant differences in the moderator variable Course (p-value =0), these differences are moderate according to the effect size of 0.558. Course 2018/2019 yields better results than Course 2019/2020 comparing averages. We conclude that we can reject *H01*, since there are significant differences between effort using the Non-stereotyped method and the Stereotyped one. The effort with a Non-stereotyped method is less than with the Stereotyped one.



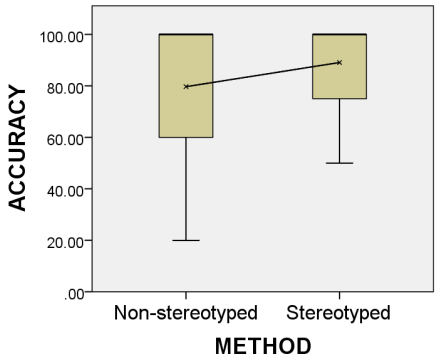
Source	P-value	Effect size
Method	**0.013**	0.575
Course	**0.000**	0.558
Problem*Method	0.378	-
Course*Method	0.836	-

Fig. 10 (a) Box-and-whisker plot for effort; (b) Output of mixed model test for effort.

5.2. Accuracy

Accuracy was measured as the percentage of sections of the generated interface compliant with the requirements (the higher percentage, the best accuracy). Fig. 11 shows the box-and-whisker plot with the percentage of accuracy per treatment. The first quartile is different between treatments. The median and third quartile for accuracy using the Non-stereotyped method is the same as for the Stereotyped method, which means that both methods generate similar GUIs compliant with requirements. Moreover, we can see that results obtained with the Non-stereotyped method are more scattered than that obtained with the Stereotyped one.

Applying the mixed model to look for significant differences between the Non-stereotyped method and the Stereotyped one, we obtain a p-value of 0.028, which means significant differences. These differences are small according to the effect size of 0.462. So, we can state that the Stereotyped method yields better accuracy. The interaction Method*Problem and the moderator variable Course do not present significant differences. We conclude that we can reject H_{02}, as there are significant differences for the accuracy of the Non-stereotyped method and the Stereotyped method, obtaining the Stereotyped one the best value.



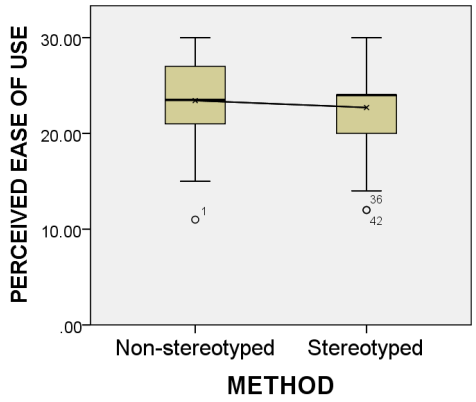
Source	P-value	Effect size
Method	**0.028**	0.462
Course	0.814	-
Problem*Method	0.371	-
Course*Method	0.194	-

Fig. 11 (a) Box-and-whisker plot for accuracy; (b) Output of mixed model test for accuracy.

5.3. Satisfaction

Satisfaction is measured in terms of PEOU, PU, and ITU on a 5-point Likert scale (the higher mark in the scale the better satisfaction). Since each metric is measured through several questions (6 for PEOU, 8 for PU, and 2 for ITU), we have aggregated questions by adding the responses per metric. Fig. 12 shows the box-and-whisker plot comparing PEOU (sum of answers of a 5-point Likert scale).

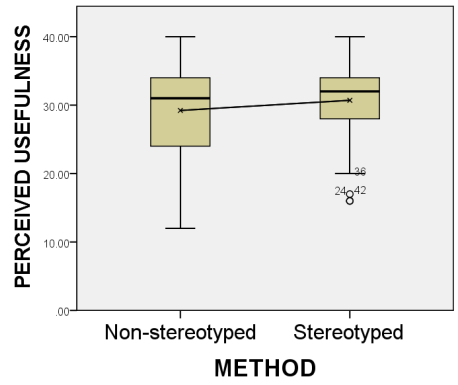
The median is the same between both treatments, but there is a difference in the first and third quartile, where the Non-stereotyped method yields the best satisfaction. The p-value for the method to generate GUIs from BPMN models is 0.152, indicating that there are not significant differences between the Non-stereotyped method and the Stereotyped one. The interaction Method*Problem and the moderator variable Course are not significant.



Source	P-value	Effect size
Method	0.152	-
Course	0.325	-
Problem*Method	0.912	-
Course*Method	0.058	-

Fig. 12 (a) Box-and-whisker plot for PEOU; (b) Output of mixed model test for PEOU.

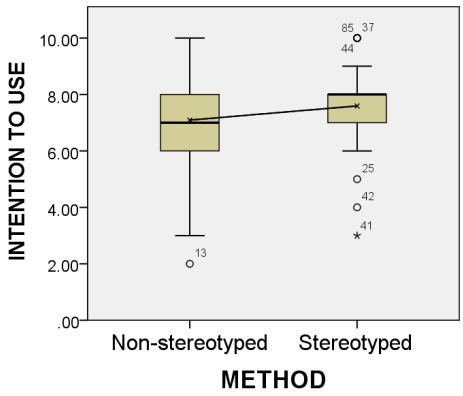
Fig. 13 shows the box-and-whisker plot comparing PU (sum of answers of PU in a 5-point Likert scale). The first quartile is different, while the median and third quartile are similar. The median for PU using the Stereotyped method is slightly higher than using the Non-stereotyped one, which means a better PU for the Stereotyped method. The p-value for the method to generate GUIs from BPMN models is 0.052, so there are not significant differences between both methods. The interaction Method*Problem and the moderator variable Course do not yield significant differences. Note that for the Course 2019/2020 replication, we yield significant differences that disappear in the aggregation of the family. This could be because in the family we have data more scattered (Fig. 13 (a)) regarding the replication alone (Fig. 9(b))



Source	P-value	Effect size
Method	0.052	-
Course	0.383	-
Problem*Method	0.620	-
Course*Method	0.087	-

Fig. 13 (a) Box-and-whisker plot for PU; (b) Output of mixed model test for PU.

Fig. 14 shows the box-and-whisker plot for ITU (sum of answers of ITU in a 5-point Likert scale). The medians and first quartiles are different; the Stereotyped method yields better results. We do not appreciate differences in the third quartile. This means that most subjects have the intention to use the Stereotyped method. The p-value for the method to generate GUIs from BPMN models is 0.026, indicating that there are significant differences between both methods; these differences are small according to the effect size of 0.289. The interaction Method*Problem and the moderator variable Course do not yield significant differences.



Source	P-value	Effect size
Method	**0.026**	0.289
Course	0.068	-
Problem*Method	0.325	-
Course*Method	0.159	-

Fig. 14 (a) Box-and-whisker plot for ITU (b) Output of mixed model test for ITU.

We conclude that H_{03} is rejected only in terms of intention to use. The method Stereotyped is perceived as easier to use as the Non-stereotyped one. Note that, the difficulty of working with stereotypes is large but with training the difficulty tends to mitigate.

6 Discussion

This section reviews all the results of the experiment. We discuss the results for each response variable. The results show that the **effort** to work with the Stereotyped method is significantly higher than the effort to work with the Non-stereotyped one, even though the effect size is moderate. This result makes sense since working with too many conceptual primitives (i.e., primitives of BPMN and our stereotypes and UML classes) requires more time than working with a smaller set of conceptual primitives (just BPMN primitives). Note importantly that the Stereotyped method requires that all the primitives are specified with no errors and with all details; otherwise, automatic transformation rules from BPMN to GUIs cannot be applied. This high precision modeling needs a larger time than the time to draw a simple BPMN model which does not carry all the information required to derive GUIs from it (as in the Non-stereotyped method).

To extract conclusions from these results, we must also consider that subjects were not experts at working with the Stereotyped method. They had made one small modeling task as training previous to the experimental task, and a test, but not a full software system. Note that subjects are already used to working with conceptual models like the simple BPMN but not with extensions based on stereotypes; the 19% of the subjects have a medium and high knowledge in BPMN models, the 68% have a medium and high knowledge in UML class diagram, and 65% have a medium and high knowledge in GUIs. These differences between treatments in the background of the subjects may justify the Non-stereotyped method requiring less effort than the Stereotyped one. Maybe, if we had recruited subjects with a wide experience in the use of the BPMN stereotypes, results for effort would have been better for the Stereotyped method, since subjects can generate GUIs automatically only modeling with BPMN.

The results show that **accuracy** using the Stereotyped method is significantly better than using the Non-stereotyped one. This result means that even though subjects require more effort to build Stereotyped models, the accuracy is better than with a Non-stereotyped one. This relates the conclusions of effort and accuracy: a higher effort involves a better accuracy. Note importantly that just with a short training, subjects were capable of building GUIs more accurate through the Stereotyped method. If we replicate the experiment with experts in the use of stereotypes, maybe the effect between treatments could be larger.

Regarding **satisfaction**, only ITU presents significant differences between methods. This means that even though the Stereotyped method requires more effort, the subjects would like to use this method in the future instead of the Non-stereotyped one. It is curious that subjects did not appreciate differences in the perceived usefulness of both methods, they consider useful both of them. The Non-stereotyped method is considered easy to use, maybe because this is the method that all subjects knew previously.

Accuracy was measured comparing the experimenters' solution of GUIs that supports all the requirements versus the GUIs generated by each subject. This means that we are focusing our validation only on the final GUIs, ignoring the accuracy in building the BPMN models. It is probable that subjects working with the Non-stereotyped method focus just on drawing good GUIs, reducing the quality of their BPMN models. This would lead to obtain less effort in the development process (the BPMN models would be done quickly) and more accuracy in the GUI (even though accuracy in the BPMN model would be poor). Accurate BPMN models are important as they carry information on the system behavior, not just its GUIs. Note that the Stereotyped method relies on building BPMN models that accurately and precisely model all the system requirements. So, the BPMN is essential in the use of the stereotyped method. In order to take into account the accuracy of the BPMN models, we are going to repeat the data analysis only with subjects whose BPMN models were specified properly in both methods. For this aim, we have compared the experimenters' solution of the BPMN models versus the BPMN built by each subject. As a result, we have removed subjects whose accuracy in the BPMN models were under a threshold. First, we are going to analyze the data of subjects whose accuracy of the BPMN models used in both treatments were over 75%. Second, we repeat the data analysis only with subjects whose accuracy of the BPMN models were 100%.

Considering only subjects whose accuracy in building BPMN models is over 75%, we reduce the sample size from 54 subjects to 30 subjects (see Table 10 for a summary of results). After this filtering, the analysis of effort yields a p-value of the method of 0.033 and an effect size of 0.551. So, there are significant differences with a moderate effect but there is a reduction of the effect regarding the sample size before filtering (0.575). This means that differences in effort between treatments are reduced when the accuracy in the BPMN models is similar. The interaction Method*Problem has a p-value of 0.235, so there are not significant differences. Course has a p-value of 0.013 indicating that there are significant differences, these differences are moderate according to the effect size of 0.616. Course 2018/2019 yields better results. We see in Fig. 15(a) how differences between medians are considerably reduced regarding Fig. 10, where no filter on the accuracy of BPMN models was applied. Accuracy in the GUIs yields a p-value for the method of 0.031 and an effect size of 0.792, larger than the obtained before the filtering (0.462). This means that differences for accuracy in GUIs when BPMN models are accurate are more evident, obtaining the Stereotyped method a better result. For the interaction Method*Problem and Course we have a p-value of 0.62 and 0.992 respectively, so there are not significant differences. We see in Fig. 15(b) that after the filtering, the accuracy of the GUIs of all the subjects using the Stereotyped method is close to 100%.

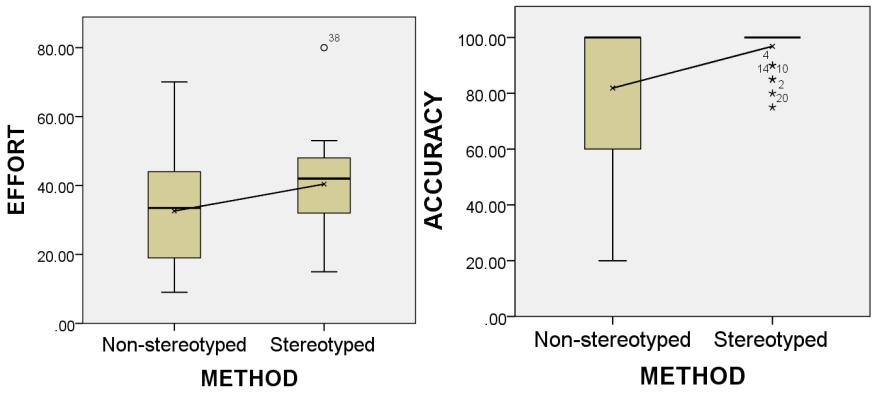


Fig. 15 (a) Box-and-whisker plot for Effort filtering subjects by 75% of BPMN model accuracy; (b) Box-and-whisker plot for Accuracy filtering subjects by 75% of BPMN model accuracy

The p-value of method in PEOU is 0.082, so there are not significant differences between the Non-stereotyped method and the Stereotyped one. The interaction Method*Problem and the Course yield a p-value of 0.779 and 0.161 respectively, which means that there are not significant differences. We see in Fig. 16(a) that there not many differences between using all the subjects or applying the filter of 75% in the BPMN model accuracy. The p-value of method in PU is 0.077, for Method*Problem is 0.828 and for Course is 0.064. So, we do not find any significant difference between treatments. In Fig. 16(b) we see that differences between analyzing all the subjects (Fig.13) or filtering per 75% of accuracy in the BPMN model has no important differences. The p-value of method in ITU is 0.003, showing that the Stereotyped method is significantly better than the Non-stereotyped one. The effect size is 0.404, which shows small effect but larger than before filtering (0.289). The interaction Method*Problem has a p-value of 0.790, showing no significant differences. Course has a p-value of 0.004 with an effect size of 1.03 (large), so, the course 2019/2020 yields significantly better results for ITU.

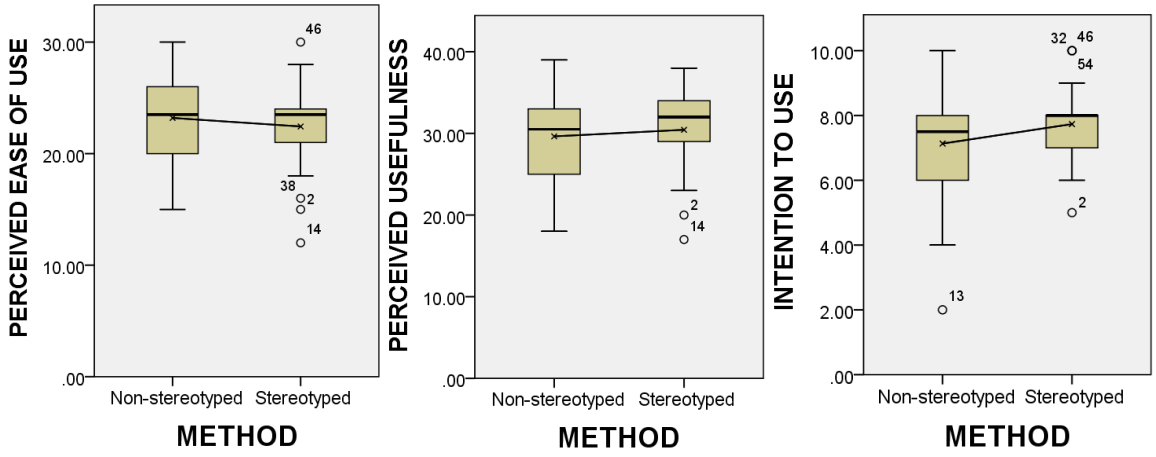


Fig. 16 (a) Box-and-whisker plot for PEOU filtering subjects by 75% of BPMN model accuracy (b) Box-and-whisker plot for PU filtering subjects by 75% of BPMN model accuracy (c) Box-and-whisker plot for ITU filtering subjects by 75% of BPMN model accuracy

As a conclusion, after filtering the subjects whose BPMN model accuracy is over 75% in both treatments, we state that differences in effort are reduced, while differences in accuracy of the GUIs and ITU are more evident. This means that when BPMN models are done properly with both methods, the time spent in the development is similar, but the stereotyped method involves more accuracy in the generated GUIs. The increase in ITU after the filtering may be because subjects perceive the high accuracy on their own.

After the filtering of 75% in the accuracy of the BPMN model, we have reduced the statistical power, but we would like to analyze if these trends remain when BPMN models are even more accurate. For this aim, next we apply a new filter, considering only subjects whose accuracy in the BPMN models of both treatments is 100%. This way, we reduce the sample size from 54 subjects to 23 subjects (see Table 10 for a summary of results). After this filtering, differences in effort are significant with a p-value of 0.038 and an effect size of 0.544 (lower than with the filter of 75%, which was 0.551). This result reinforces the idea that when BPMN models are accurate, differences in effort tend to decrease. The interaction Method*Problem has a p-value of 0.21, so there are not significant differences. Course has a p-value of 0.038 and effect size of 0.657, course 2018/2019 yields better results. We see in Fig. 17(a) that differences between treatments are similar as the filtering of 75% of Accuracy in BPMN models (Fig. 15(a)). Regarding accuracy in the GUIs, the p-value for the method is 1.00, this means that there are not significant differences, the interaction Method*Problem and Course are not significant with a p-value of 1.00 in both cases. We see in Fig. 17(b) that the stereotyped method yields better results, this means that when the models are built correctly, we obtain GUIs more compliant with the requirements. Note in Fig. 17(b) that in the case of the Stereotyped method, we have 100% of accuracy in the GUIs. This is because according to the MDD paradigm, the model is the code in the Stereotyped method (GUIs are automatically generated from BPMN models). So, if the BPMN model is perfect (100% of accuracy in the BPMN model), the GUI is also perfect (100% of accuracy in the GUI). This relationship does not appear in the Non-stereotyped

method, since the model is just a documentation, and GUIs generation is done manually, which involves possible mistakes in this transformation that reduces the GUIs accuracy.

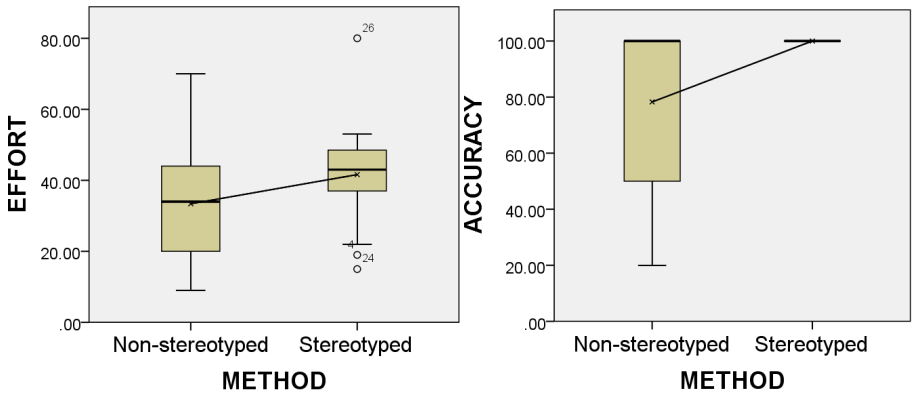


Fig. 17 (a) Box-and-whisker plot for Effort filtering subjects by 100% of BPMN model accuracy (b) Box-and-whisker plot for Accuracy filtering subjects by 100% of BPMN model accuracy

In PEOU, the p-value for the method is 0.161, for the interaction Method*Problem is 0.969, and for Course is 0.986. So, there are not significant differences. Moreover, Fig. 18(a) is similar to Fig. 16(a), so there are not differences between filtering by 75% and by 100%. This pattern is the same for PU, where the p-value for the method is 0.067, for the interaction Method*Problem is 0.947, and for Course is 0.72. The significant differences that we had for ITU in the filtering of 75% of accuracy in the BPMN model disappear if we filter by 100%. The p-value for the method is 0.08, for the interaction Method*Problem is 0.932, and for Course is 0.097.

However, if we compare box-whiskers of both filtering (Fig. 18(c) and Fig. 16(c)), we see that when we filter by 100% of accuracy in the BPMN model, values are more concentrated around the medians. So, subjects intend to use both methods indistinctively.

As a conclusion, we cannot state that if we only analyze perfect BPMN models, differences in both methods are significantly more important for accuracy, even though we have seen this trend through the descriptive data. We have checked that for effort, these differences are reduced, which means that the more accurate BPMN models in both treatments, the least differences in effort. Note also that the filtering involves a reduction of power that may lead us to avoid rejecting null hypotheses due to few subjects.

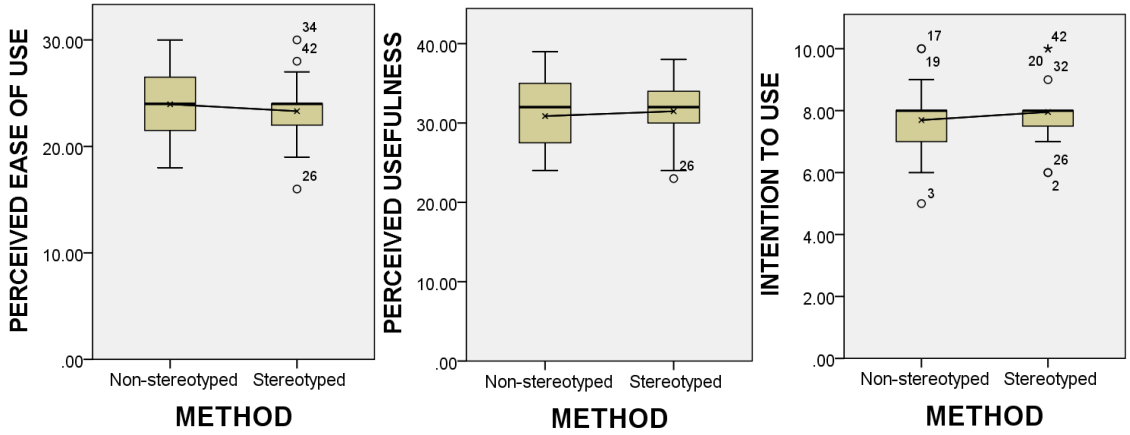


Fig. 18 (a) Box-and-whisker plot for PEOU filtering subjects by 100% of BPMN model accuracy (b) Box-and-whisker plot for PU filtering subjects by 100% of BPMN model accuracy (c) Box-and-whisker plot for ITU filtering subjects by 100% of BPMN model accuracy

Table 10 P-values and effect sizes for the Stereotyped method compared to the Non-stereotyped one filtering subjects by 75% and 100% of BPMN model accuracy.

	Group 75%		**Group 100%**	
	p-value	**Effect size**	**p-value**	**Effect size**
Effort	**0.033**	0.551	**0.038**	0.544
Accuracy	**0.031**	0.792	1.00	-
PEOU	0.082	-	0.161	-
PU	0.077		0.067	-
ITU	**0.003**	0.404	**0.08**	-

7 Threats to validity

This section discusses the threats to validity of our experiment. In the following, we describe the threats according to Wohlin's classification [66]. For each group of threats, we made a distinction between threats that we were unable to address, threats whose effect we managed to minimize, and threats that we solved. We classify the threats into four groups: Conclusion validity, Internal validity, Construct validity, and External validity.

Conclusion validity: This type of threat deals with the ability to draw the correct conclusion about relations between the treatment and outcome. The experiment may suffer the following threats of this type: *Low statistical power*, which means that the number of subjects is not enough to reveal a true pattern in the data. We solved this threat since we have 54 subjects, which implies a high statistical power for our experiment. Another threat that appears is *Subjects of random heterogeneity,* which means that there is always heterogeneity in a study group. The risk appears when the variation due to individual differences is larger than that due to the treatment. In order to solve this threat, we recruited subjects with similar profiles (students are undergraduates who have previously taken Software Engineering courses, with knowledge in interface programming). Moreover, we used demographic questionnaires to detect differences among the profiles of the subjects. Another threat that appears is *Fishing*, which means that the experimenters are looking for specific results. In order to minimize this threat, the experiment was performed without showing the aim to the subjects. Another threat that appears is *Reliability of measures,* which means that the validity of an experiment is highly dependent on the reliability of the measures. In order to minimize this threat, the metrics were calculated by an experimenter. Moreover, effort and satisfaction were automatically calculated (with the entry and end times and through a questionnaire, respectively), which reduces possible errors.

Internal validity: This type of threat deals with influences that can affect the factor concerning causality. The experiment may suffer the following threats of this type: *Experience of the subjects*, which means that the experience of the subjects is not enough to conduct the experiment. Our subjects had two years of experience in coding GUIs. In order to solve the lack of experience in BPMN, we trained them through a tutorial, and they had to pass a test before participating in the experiment. Note that subjects only had to draw GUIs, they did not need a large experience in BPMN modeling. *Instrumentation,* which means that the use of instruments may affect the results. In our case, results depend on experimental problems that are not much complex to limit the time of the whole experiment to 2 hours. In order to minimize this threat, we used two different problems in each replication. *History,* which means that differences may arise when treatments are applied at different times. We solved this threat by conducting the experiment only in one session of two hours. Another threat that may appear is *Maturation,* which means the effect of reacting differently as time passes. In order to solve this threat, the experiment has two experimental problems for each treatment; this is to avoid learning between treatments. Another threat that appears is *Instrumentation,* which means that the artifacts used in the experiment could affect the results. In order to minimize this threat, to measure the effort, we used an online platform for subjects to record the time they began to develop the experimental problem and the time they ended the problem. Satisfaction was measured through online questionnaires that save the data automatically. The experimenter used a spreadsheet as a help to apply the formula of the accuracy. Another threat that appears is *Selection,* which means the effect of natural variation in human performance, depending on how the subjects are selected from a larger group, the selection effects can vary. In order to minimize this threat, we have recruited all the subjects voluntarily. Another threat that appears is *Resentful demoralization*, which means that some treatments can be more motivating than others. In order to minimize this threat, we motivated the subjects with extra marks in the Software Engineering course. Another threat that appears is *Ecological validity* [67], which means that the context of use in which the experiment was conducted may affect the results. In order to minimize this threat, subjects conducted the experiment in a controlled environment and no in a real environment, such as corporate environments where real analysts and designers work collaboratively.

Construct validity: This type of threat concerns generalizing the result of the experiment to the concept or theory behind the experiment. The experiment may suffer the following threats of this type: *Evaluation apprehension*, which means that some people are afraid of being evaluated. In order to minimize this threat, we communicate to the subjects that these experimental problems are exercises that allow learning objectives of the course, without mentioning the term "experiment" or "test". Also, each subject signed a consent form. Another threat that appears is *Hypothesis guessing*, which means that when people take part in an experiment, they might try to figure out the purpose of the experiment. In order to minimize this threat, we do not talk about research questions in the experiment. Another threat that appears is *Interaction of testing and treatment*, which appears when the treatment is part of a test in the course. In order to minimize this threat, subjects were recruited voluntarily, and participants got extra points to pass the course.

External validity: This type of threat concerns conditions that limit our ability to generalize the results of our experiments to industrial practice. The experiment may suffer the following threats of this type: *Interaction of selection and treatment*, which means the effect of having a subject population not representative of the population we want to generalize. Our experiment suffers this threat since we cannot ensure that results can be generalized to subjects with different profiles of our sample. Another threat of this type that appears is *Interaction of setting and treatment,* which means the effect of not having the experimental setting or material representative of industrial practice. In order to minimize this threat, we have used problems whose context is widely known among the subjects. Another threat that appears in our study is *Limit of scope of the experiment:* from 14 rules that compose the proposal to generate GUIs from BPMN models, this experiment focuses on 5 of them (the ones that appear in BPMN models more frequently). Our experiment suffers this threat since the generalization of results is only valid for the 5 rules used in the experiment. Further experiments must be conducted to validate the rest of rules. Note that even though we validated 5 rules, we identify significant differences between both treatments. The use of more rules could even result in a larger effect between treatments.

8 Conclusion and Future Work

This paper presents a family of experiments conducted to assess a method to automatically generate GUIs from BPMN models extended with stereotypes. Each stereotype defines how the process will be displayed in the GUI and represents a unique alternative in the model to GUI transformation rules.

The experiment involved 54 subjects divided into two replications and for each replication two experimental problems were used. Each replication compares the use of standard, Non-stereotyped BPMN models and the manual derivation of GUIs from them versus the use of Stereotyped BPMN models and the automatic generation of GUIs from them. Response variables of the experiment are: effort (measured as the time in minutes taken to build GUIs after watching a video describing requirements), accuracy (measured as the percentage of sections of the generated GUI compliant with the requirements) and satisfaction (measured in terms of PEOU, PU and ITU).

The results of the family of experiments show that there are significant differences in effort, accuracy and ITU between the Non-stereotyped method and the Stereotyped one. The Stereotyped method requires more effort, while it yields better accuracy and ITU. The fact of focusing our analysis of accuracy on the GUI might hide the accuracy of the BPMN model, which is essential for the Stereotyped method and which allows for a higher quality results in the whole system development. If we reduce the sample size to subjects whose BPMN models had a high accuracy regarding the experimenters' solution (over 75%), we find that differences in effort and ITU are reduced while differences in accuracy are larger. When we filter subjects considering only those that had built BPMN models having an accuracy of 100%, we obtain significant differences only for effort, likely due to the lack of statistical power after filtering experimental units. This filtering shows one of the characteristics of the MDD paradigm in the Stereotyped method: the model is the code; there is a perfect correlation between subjects that build a perfect model and subjects that design a perfect GUI.

The experiment suffers from the following limitations: (1) the subjects had watched the stereotyped method tutorial only one week before the experiment, (2) the subjects had no experience with the use of stereotypes before the experiment, (3) we have used experimental problems simple for this experiment due to time limitations. Note that even with the poor background of the subjects in the area of stereotypes and only with a short tutorial to learn stereotypes in BPMN, we concluded significant differences between treatments, where the Stereotyped method yields better results. These results lead to think that recruiting for future experiments experts in stereotypes, these differences could be even higher.

Currently, there is little or no literature tackling with experiments on GUIs from BPMN models, so this paper is a step forward to cover this gap. Other approaches to generate GUIs based on the use of stereotypes exist, so our proposal is aligned with previous existing works. This work aims to automate the software development process, aligned with the MDD paradigm. Note that the list of GUI widgets we consider can be extended with more sophisticated ones. This way the method can be extended for more enhanced GUIs

As future work, we plan to replicate this experiment changing some elements of the design. First, we plan to use more stereotypes and more rules: there are other elements of the GUIs that could not be evaluated due to time limitations in the experiments; the current work focuses on the most relevant GUIs widgets. Second, we plan to recruit subjects with a wide experience of working with BPMN models or with stereotypes. Third, we plan to propose a recommendation of stereotypes to optimize the usability of the generated GUIs. This way, the developer will be able to follow the recommendations to generate usable interfaces from BPMN models.

Acknowledgements The first author acknowledges support from the Ministry of Education of Peru with the National Scholarship and Educational Loan Program PRONABEC – President of the Republic Scholarship. This project also has the support of Spanish Ministry of Science and Innovation through project DATAME (ref: TIN2016-80811-P). We would like to thank the subjects for conducting the experiments.

References

[1] BPMN. (2013). *Business Process Modeling Notation*. Available: http://www.bpmn.org

[2] M. Dumas, M. La Rosa, J. Mendling, and H. A. Reijers, *Fundamentals of business process management* vol. 1: Springer, 2013.

[3] S. A. White, *BPMN Modeling and reference guide*, First Edition ed., 2008.

[4] E. Diaz, J. I. Panach, S. Rueda, and O. Pastor, "Towards a method to generate GUI prototypes from BPMN," in *2018 12th International Conference on Research Challenges in Information Science (RCIS)*, 2018, pp. 1-12.

[5] Bizagi. (2017). *Examples of BPMN Projects*. Available: https://www.bizagi.com/es/comunidad/process-xchange

[6] V. Paradigm. (2004). *Visual Paradigm, https://www.visual-paradigm.com/*. Available: https://www.visual-paradigm.com/

[7] J. D. Gauchat, *El gran libro de HTML5, CSS3 y Javascript*: Marcombo, 2012.

[8] D. W. Embley, S. Liddle, and Ó. Pastor, "Conceptual-Model Programming: A Manifesto," in *Handbook of Conceptual Modeling*, ed: Springer, 2011, pp. 3-16.

[9] S. J. Mellor, A. N. Clark, and T. Futagami. (2003) Guest Editors' Introduction: Model-Driven Development. *IEEE Software*. 14-18.

[10] Y. Singh and M. Sood, "Model Driven Architecture: A Perspective," in *Advance Computing Conference, 2009. IACC 2009. IEEE International*, 2009, pp. 1644-1652.

[11] M. J. Rutherford and A. L. Wolf, "A case for test-code generation in model-driven systems," in *2nd international conference on Generative programming and component engineering*, Erfurt, Germany, 2003, pp. 377-396.
[12] P. Papotti, A. Prado, W. Souza, C. Cirilo, and L. Pires, "A Quantitative Analysis of Model-Driven Code Generation through Software Experimentation," in *Advanced Information Systems Engineering*. vol. 7908, C. Salinesi, M. Norrie, and Ó. Pastor, Eds., ed: Springer Berlin Heidelberg, 2013, pp. 321-337.
[13] K. Krogmann and S. Becker, "A case study on model-driven and conventional software development: The palladio editor," in *Software Engineering 2007 - Beiträge zu den workshops*, 2007, pp. 169–176.
[14] W. Heijstek and M. R. Chaudron, "Empirical investigations of model size, complexity and effort in a large scale, distributed model driven development process," in *2009 35th Euromicro Conference on Software Engineering and Advanced Applications*, 2009, pp. 113-120.
[15] P. Baker, S. Loh, and F. Weil, "Model-driven engineering in a large industrial context - motorola case study," in *8th International Conference of Model Driven Engineering Languages and Systems (MoDELS)*, Montego Bay, Jamaica, 2005, pp. 476–491.
[16] Ó. Pastor, S. España, and J. I. Panach, "Learning Pros and Cons of Model-Driven Development in a Practical Teaching Experience," in *Advances in Conceptual Modeling: ER 2016 Workshops, AHA, MoBiD, MORE-BI, MReBA, QMMQ, SCME, and WM2SP, Gifu, Japan, November 14–17, 2016, Proceedings*, S. Link and J. C. Trujillo, Eds., ed Cham: Springer International Publishing, 2016, pp. 218-227.
[17] E. E. Thu and N. Nwe, "Model driven development of mobile applications using drools knowledge-based rule," in *2017 IEEE 15th International Conference on Software Engineering Research, Management and Applications (SERA)*, 2017, pp. 179-185.
[18] Y. Martínez, C. Cachero, and S. Meliá, "MDD vs. traditional software development: A practitioner's subjective perspective," *Information and Software Technology,* vol. 55, pp. 189-200, 2013.
[19] Bizagi. (2002). *Bizagi*. Available: https://www.bizagi.com/es
[20] IBM. (2017). *Auraportal*. Available: https://www.auraportal.com/es/
[21] BonitaSoft. (2019). *BonitaSoft*. Available: https://www.bonitasoft.com/
[22] E-citiz. *E-citiz Studio*. Available: https://www.e-citiz.com/bpm
[23] R. Acerbis, A. Bongio, M. Brambilla, and S. Butti, "WebRatio 5: An Eclipse-Based CASE Tool for Engineering Web Applications," *Lecture Notes in Computer Science,* vol. 4607, pp. 501-505, 2007.
[24] IEEE, "Systems and software engineering -- Vocabulary," ISO/IEC/IEEE 24765:2010(E), Ed., ed, 2010, pp. 1-418,.
[25] M. Brambilla, P. Fraternali, and C. Vaca, "BPMN and design patterns for engineering social BPM solutions," in *International Conference on Business Process Management*, 2011, pp. 219-230.
[26] M. Brambilla, S. Butti, and P. Fraternali, "Webratio bpm: a tool for designing and deploying business processes on the web," in *International Conference on Web Engineering*, 2010, pp. 415-429.
[27] L. Han, W. Zhao, and J. Yang, "An approach towards user interface derivation from business process model," *Communications in Computer and Information Science,* vol. 602, pp. 19-28, 2016.
[28] K. S. Sousa, H. Mendonça, and J. Vanderdonckt, "A Model-Driven Approach to Align Business Processes with User Interfaces," *J. UCS,* vol. 14, pp. 3236-3249, 2008.
[29] S. Yongchareon, C. Liu, X. Zhao, and J. Xu, "An artifact-centric approach to generating web-based business process driven user interfaces," in *International Conference on Web Information Systems Engineering*, 2010, pp. 419-427.
[30] S. Yongchareon, C. Liu, X. Zhao, J. Yu, K. Ngamakeur, and J. Xu, "Deriving user interface flow models for artifact-centric business processes," *Computers in Industry,* vol. 96, pp. 66-85, 2018/04/01/ 2018.
[31] W. Bouchelligua, A. Mahfoudhi, N. Mezhoudi, O. Daassi, and M. Abed, "User interfaces modelling of workflow information systems," in *Lecture Notes in Business Information Processing* vol. 63 LNBIP, ed, 2010, pp. 143-163.
[32] J. Gonzalez-Huerta, A. Boubaker, and H. Mili, "A business process re-engineering approach to transform BPMN models to software artifacts," *Lecture Notes in Business Information Processing,* vol. 289, pp. 170-184, 2017.
[33] I. Abouzid and R. Saidi, "Proposal of BPMN extensions for modelling manufacturing processes," in *2019 International Conference on Optimization and Applications, ICOA 2019*, 2019.
[34] G. Decker and F. Puhlmann, "Extending BPMN for modeling complex choreographies," in *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, 2007, pp. 24-40.
[35] A. Barros, M. Dumas, and A. H. Ter Hofstede, "Service interaction patterns," in *International Conference on Business Process Management*, 2005, pp. 302-318.

[36] A. Rodríguez, E. Fernández-Medina, and M. Piattini, "A BPMN extension for the modeling of security requirements in business processes," *IEICE transactions on information and systems,* vol. 90, pp. 745-752, 2007.
[37] L. J. R. Stroppi, O. Chiotti, and P. D. Villarreal, "A BPMN 2.0 extension to define the resource perspective of business process models," in *XIV Congreso Iberoamericano en Software Engineering*, 2011.
[38] M. Zur Muehlen, "Organizational management in workflow applications–issues and perspectives," *Information Technology and Management,* vol. 5, pp. 271-291, 2004.
[39] R. Braun and H. Schlieter, "Requirements-based development of bpmn extensions: The case of clinical pathways," in *2014 IEEE 1st International Workshop on the Interrelations between Requirements Engineering and Business Process Management (REBPM)*, 2014, pp. 39-44.
[40] R. H. Von Alan, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS quarterly,* vol. 28, pp. 75-105, 2004.
[41] M. B. Juric, B. Mathew, and P. G. Sarang, *Business process execution language for web services: an architect and developer's guide to orchestrating web services using BPEL4WS*: Packt Publishing Ltd, 2006.
[42] K. Zarour, D. Benmerzoug, N. Guermouche, and K. Drira, "A BPMN extension for business process outsourcing to the cloud," in *World Conference on Information Systems and Technologies*, 2019, pp. 833-843.
[43] E. F. Cruz and A. M. R. da Cruz, "Deriving Integrated Software Design Models from BPMN Business Process Models," in *ICSOFT*, 2018, pp. 605-616.
[44] E. F. Cruz, R. J. Machado, and M. Y. Santos, "From business process models to use case models: A systematic approach," in *Enterprise engineering working conference*, 2014, pp. 167-181.
[45] D. Brdjanin, G. Banjac, D. Banjac, and S. Maric, "An experiment in model-driven conceptual database design," *Software & Systems Modeling,* vol. 18, pp. 1859-1883, 2019.
[46] W. Khlif, N. E. Ben Ayed, and H. Ben-Abdallah, "From a BPMN model to an aligned UML analysis model," in *ICSOFT 2018 - Proceedings of the 13th International Conference on Software Technologies*, 2019, pp. 623-631.
[47] F. Radeke and P. Forbrig, "Patterns in task-based modeling of user interfaces," in *International Workshop on Task Models and Diagrams for User Interface Design*, 2007, pp. 184-197.
[48] Q. Limbourg, J. Vanderdonckt, B. Michotte, L. Bouillon, M. Florins, and D. Trevisan, "Usixml: A user interface description language for context-sensitive user interfaces," in *Proceedings of the ACM AVI'2004 Workshop" Developing User Interfaces with XML: Advances on User Interface Description Languages"(Gallipoli*, 2004, pp. 55-62.
[49] J. G. García, C. Lemaigre, J. M. González-Calleros, and J. Vanderdonckt, "Model-driven approach to design user interfaces for workflow information systems," *J. UCS,* vol. 14, pp. 3160-3173, 2008.
[50] J. I. Panach, S. España, Ó. Dieste, Ó. Pastor, and N. Juristo, "In search of evidence for model-driven development claims: An experiment on quality, effort, productivity and satisfaction," *Information and Software Technology,* vol. 62, pp. 164-186, 2015.
[51] N. Mellegård and M. Staron, "Distribution of Effort among Software Development Artefacts: An Initial Case Study.," in *Proc. of 11th International Workshop BPMDS 2010, held at CAiSE 2010*, Hammamet, Tunisia, 2010, pp. 234-246.
[52] W. Heijstek and M. R. V. Chaudron, "Empirical Investigations of Model Size, Complexity and Effort in a Large Scale, Distributed Model Driven Development Process," in *Software Engineering and Advanced Applications, 2009. SEAA '09. 35th Euromicro Conference on*, 2009, pp. 113-120.
[53] E. Diaz, J. I. Panach, S. Rueda, and O. Pastor, "Generación de Interfaces de Usuario a partir de Modelos BPMN con Estereotipos," presented at the Jornada de la Sociedad de Ingeniería de Software y Tecnologías de Desarrollo de Software (SISTEDES), 2018.
[54] B. BPMN. (2017). *Business model patterns*. Available: http://resources.bizagi.com/docs/Workflow_Patterns_using_BizAgi_Process_Modeler_Esp.pdf
[55] Iso/iec, "ISO/IEC 25000 - Software engineering - Software product Quality Requirements and Evaluation (SQuaRE) - Guide to SQuaRE," 2005.
[56] N. Juristo and A. Moreno, *Basics of Software Engineering Experimentation*: Springer, 2001.
[57] D. L. Moody, "The method evaluation model: a theoretical model for validating information systems design methods," presented at the European Conference on Information Systems (ECIS 03), Naples, Italy 2003.
[58] O. I. Lindland, G. Sindre, and A. Solvberg, "Understanding quality in conceptual modeling," *IEEE Software,* vol. 11, pp. 42-49, 1994.

[59] S. Jamieson, "Likert scales: how to (ab)use them," *Medical Education,* vol. 38, pp. 1217-1218, 2004.
[60] Byron Wm. Brown, Jr., "The Crossover Experiment for Clinical Trials," *Biometrics,* vol. 36, pp. 69-79, 1980.
[61] B. T. West, K. B. Welch, and A. T. Galecki, *Linear mixed models: a practical guide using statistical software*: CRC Press, 2014.
[62] L. S. Meyers, *Applied multivariate research : design and interpretation / Lawrence S. Meyers, Glenn Gamst, A.J. Guarino*. Thousand Oaks: SAGE Publications, 2006.
[63] L. Cohen, *Statistical power analysis for the behavioral sciences*, 2nd. Edition ed.: Lawrence Earlbaum Associates, 1988.
[64] T. Dybå, V. B. Kampenes, and D. I. K. Sjøberg, "A systematic review of statistical power in software engineering experiments," *Information and Software Technology,* vol. 48, pp. 745-755, 2006.
[65] F. Faul, E. Erdfelder, A.-G. Lang, and A. Buchner, "G*Power 3: A flexible statistical power analysis program for the social, behavioral, and biomedical sciences," *Behavior Research Methods,* vol. 39, pp. 175-191, 2007/05/01 2007.
[66] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in Software Engineering: An Introduction*: Springer, 2012.
[67] S. Kieffer, "ECOVAL: Ecological validity of cues and representative design in user experience evaluations," *AIS Transactions on Human-Computer Interaction,* vol. 9, pp. 149-172, 2017.