Improvement of EduBPMN Transformation Rules from an Empirical Validation

Eduardo Díaz[1] and Jose Ignacio Panach[2]

[1] Universidad Peruana de Ciencias Aplicadas, Prolongación Primavera 2390, Lima 15023 – Perú
[2] Escola Tècnica Superior d'Enginyeria, Departament d'Informàtica Universitat de València, Avenida de la Universidad, s/n, 46100, Burjassot, València, Spain.
pcsijord@upc.edu.pe, joigpana@uv.es

Abstract. EduBPMN method allows the generation of graphical components from a BPMN model complemented with the UML class diagram. This article proposes the improvement of five transformation rules of the EduBPMN method from an experiment developed in 2019, with the improved transformation rules an experiment was developed that was executed by 31 subjects where the results of two metrics were obtained, (i) the correctness of the rules, where the subjects had to map BPMN to graphic components intuitively through an experimental problem, had a positive result (87.50%), (ii) the satisfaction of the generalization of the rules, had a positive result in its Perceived Ease of Use (93%), Perceived Usefulness (95%), and Intention to Use (96%). This article provides positive results on the five new improved rules of the EduBPMN method, which is used to map BPMN to graphical components.

Keywords: BPMN, Transformation rules, Graphic components, Experiment.

1 Introduction

Currently, the generation of graphical components from a conceptual model has taken prominence based on the Model Driven Software Development (MDD) paradigm [1]. There are different methods that work in this paradigm such as the EduBPMN method [2] [3] [4] [5] [6] [7], MoCaDiX [28], OO-METHOD [8] and others. It must be taken into account that MDD improves the productivity and quality of the development process, applying an approach that uses models at different levels of abstraction and transformations between said models. The EduBPMN method allows generating graphical components from a Business Process Model and Notation (BPMN) model complemented with a UML class diagram. The BPMN model allows understanding the business procedures of organizations [9]. BPMN is a widely used model to elicit simple and complex business process requirements, furthermore, it is complemented by the UML class diagram [10] to use the attributes with their data types. EduBPMN contains a set of transformation rules that involves a set of graphic components, where for each graphic component a stereotype was assigned to avoid ambiguities in the automatic generation of a graphic component, the method was based on the study of BPMN patterns (sequence pattern, exclusive decision pattern, synchronization pattern, implicit decision pattern, and union pattern and synchronized structure) and on the analysis of Bizagi projects [11] that correspond to a business, health, administrative and academic context. EduBPMN can perform the extension of a BPMN model with stereotypes, these models are developed in the Visual Paradigm v 15.0 application [12] where it can be exported in an XML file, through a web compiler, so that it can automatically generate graphical components.

The contribution of this work is to provide the improvement of five transformation rules of the EduBPMN method, these transformation rules were improved from the validation of transformation rules of an experiment in 2019, where in this experiment nine transformation rules of the EduBPMN were validated with students from the University of Valencia (Spain). With the five improved EduBPMN rules, a new experiment was carried out based on two metrics: (i) correctness of the transformation rules and (ii) satisfaction of the generalization of transformation rules to map BPMN to graphical components. The subjects of this experiment are students of the Software Engineering career of the Peruvian University of Applied Sciences (Peru) where they know the design of graphic components, UML class diagram, and they were trained on the elements of BPMN so that they can develop the experiment. For the correctness of the transformation rules, the subjects subjectively had to draw graphic components from a BPMN complemented with a UML class diagram. The results were compared with the graphic components of the transformation rules proposed in this article, where positive results were obtained with 87.50% correctness. For the satisfaction of the generalization of the rules, the subjects had to fill out a questionnaire based on a Moody's framework [13], based on the work of Lindland's [14]. The questionnaire was based on 16 questions, where it was measured in terms of Perceived Ease of Use (FUP) with 6 questions, Perceived Utility (UP) with 8 questions, and Intention to Use (UI) with 2 questions. The questionnaire is based on questions where the existence of whether it would be useful is evaluated and if there is an intention to use transformation rules in a general way to map BPMN to graphical components, the results were positive with 93% in FUP, 95% in UP, and 96% in UI.

This paper is structured as follows: Section 2 reviews the literature related to the generation of graphical components from BPMN. Section 3 shows EduBPMN. Section 4 presents how the rules of EduBPMN were improved. Section 5 defines the planning and definition of the experiment with the new graphic components for the transformation rules. Section 6 shows the results of the experiment. Section 7 shows the discussions of the work. Section 8 shows the threats to validity where they might affect the experiment. Section 9 presents the conclusions and future work.

2 State of the Art

In this section we review work related to the generation of graphical component design alternatives using a Directed Literature Review (TLR), an in-depth, informative, non-systematic literature review aimed at retaining only significant references to minimize selection bias. The search string in this work was carried out in the Scopus digital library (see https://www.scopus.com/home.uri): ("BPMN" AND "user interface" OR "GUIs" OR "extension" AND “experiment”). The inclusion criteria are: (1) generation of graphical components from a BPMN model, (2) extensions of BPMN models. The exclusion criteria are: (1) models other than the BPMN model, (2) approaches that do not generate design alternatives for graphical components from a BPMN model. The first search shows 110 scientific articles. After applying inclusion and exclusion criteria, the sample of 9 articles has been considered. The articles accepted in the search are shown below:

2.1 Generation of graphical user interfaces from a BPMN model

Bouchelligua et al. [15] defined an approach with a design methodology supported by a set of transformations based on Model Driven Engineering (MDE) [16]. These transformations allow you to derive design alternatives for graphical workflow components, and the interaction of the BPMN model as the task model, and other models. Torres et al. [17] propose an extension to the OOWS Web Engineering method for the development of web applications based on business processes such as BPMN, this extension contains the existence of manual tasks in a B2B, using the executable specification in WS-BPEL. Brambilla et al. [18] describe a BPMN model extended with information on task assignment, policies, activity semantics, and written data flows. The proposal is based on WebRatio, a model-driven web application that allows you to edit BPMN models and automatically transform them into running JEE applications. Sousa et al. [19] proposed an approach to unifying a business process with graphical components, (1) defining associations between business processes and graphical component, and (2) presenting a tool for model transformation that addresses traceability. LeiHan et al. [20] defined an approach for the derivation of graphical components from BPMN models. This is based on a role-enriched business process model developed with task descriptions and associated data, thereby extending the BPMN model. A set of control flow and data flow patterns are identified for the derivation of the graphical components. A complete set of restrictions and recommendations is specified to support the generation and update of the graphical components.

As a conclusion on works considered in this sub-section deal with a set of transformations to generate graphical components from BPMN [17] [19] [21], while others integrate the specifications in the BPMN models [18] [20] [22]. On the contrary, this paper proposes the use of transformation rules that allow the use of a single model to generate graphical components from BPMN.

2.2 Extended BPMN model

In the work of Rodríguez et al. [23] extended BPMN to incorporate security requirements into business process diagrams in accordance with a Model Driven Architecture (MDA). The extension allows the business analyst to express the security requirements from his own perspective. Stroppi et al. [24] presented an extension of the BPMN model, using the extension mechanisms provided by the BPMN 2.0 metamodel. They focused on 3 main aspects of the resource perspective [25], resource structure, authorization, and work distribution, thus improving the communication of resource perspective requirements between analysts and technical developers. Abouzid et al. [26] defined a set of BPMN extensions that represent some crucial manufacturing domain concepts for business process improvement. BPMN extensions allow you to incorporate information into the process model, from a manufacturing point of view, it makes the process more complete. Intrigila et al. [27] propose a lightweight BPMN extension that specifically addresses data properties in terms of constraints, preconditions, and postconditions that business process activities must satisfy. The model allows software analysts and developers to provide information to easily assign updates to the software implementation.

To summarize related works considered in this sub-section, we can state that some use a design framework with extensions of the BPMN [23] [24] [26] [27], to capture relevant information. Therefore, the use of BPMN extension mechanisms is remarkable, taking into account that in our proposal it is for the automatic generation of graphical components.

3 EduBPMN Method

This section shows the EduBPMN method [2] [3] [4] [5] [6] [7] which consists of 15 transformation rules, each rule contains a set of graphical components, where each graphical component is represented by a stereotype from a BPMN model complemented with the UML class diagram in order to use the attributes. The transformation rules were extracted from the analysis of 14 Bizagi BPMN projects [11]. EduBPMN was based on 5 BPMN patterns (sequence pattern, exclusive decision pattern, synchronization pattern, implicit decision pattern, and synchronization structured join pattern) [29]. EduBPMN allows you to develop an extended BPMN model in the Visual Paradigm v. 15.0 [12], the stereotypes were configured so that they can be added to the extended BPMN model. This modeler allows exporting to an XML format file where a web compiler developed in the PHP and HTML5 programming languages can automatically generate graphical components in HTML5. Next, an example of the R0 transformation rule that corresponds to the UML class diagram will be shown, and an example of the R2 transformation rule that corresponds to the BPMN model with the sequence pattern: (i) the R0 transformation rule is used for three data attributes (text string, integers and booleans) of the UML class diagram, each attribute is represented with a graphic component depending on its data type: (1) Text box, for any text string; (2) List box or Combo box for any enum with simple option, (3) Radio button or Check box for any boolean value. For each graphic component a stereotype was assigned to avoid ambiguities in the generation of graphic components. (ii) the R2 transformation rule is used when the sequence pattern appears, this pattern indicates that when there are sequential user-type tasks they have to be developed in an orderly manner one after the other. Figure 1 shows the R2 transformation rule, where from two user-type tasks three graphic component design alternatives can be generated, such as: Wizard (Navigation Assistant) where each user-type task is a form, Tabbed dialog box (Dialog box with tabs) where a form contains tabs, one tab for each user-type task, or Group box (group of boxes) where each user-type task is a group. It must be taken into account that transformation rule 2 is complemented by transformation rule R0 in order to be able to use the attributes of the UML class diagram and convert them into graphical components such as text box, combo box, list box, etc.

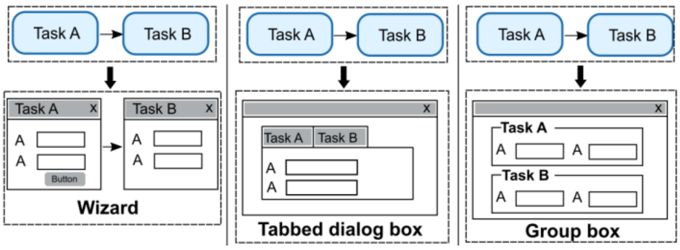

Fig. 1. R2 transformation rule example.

4 Definition of Improved transformation rules

This section shows the transformation rules that were improved in the EduBPMN method. In order to improve these rules, the results of the transformation rules that obtained the worst results in an experiment in 2019 have been considered. For this experiment, nine transformation rules of the EduBPMN method were evaluated with students from the University of Valencia (Spain) [4]. These worse results are due to the fact that the students used other graphical components that were not similar to the graphical components of the transformation rules of the EduBPMN method (these transformation rules are R2 and R3 (sequence pattern), R7 (implicit decision pattern) and R8 (generic transformation rule for any pattern)). To improve the transformation rules for this article, we will use the graphical components that the students voluntarily preferred to draw in the validated experiment.

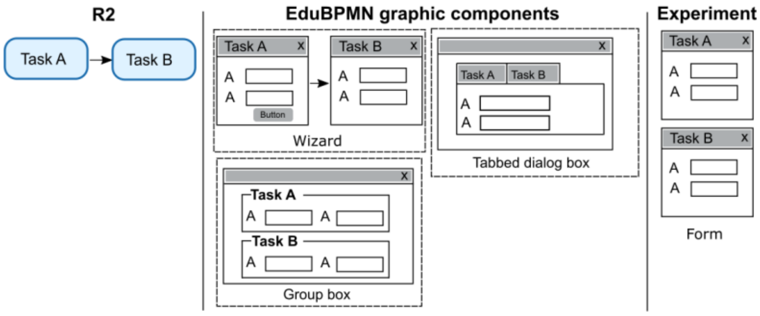

Fig. 2. Graphic components of the R2 transformation rule of the EduBPMN method and Experiment of the year 2019.

Figure 2 shows the R2 transformation rule used with the sequential pattern, where BPMN User type tasks are displayed sequentially. For each user type task (Task A and Task B) they allow mapping a Wizard (navigation assistant), where each user type task is a form, Tabbed dialog box (dialog box with tabs) where each user type task is a tab, or Group box, where each user type task is a group. The students, when analyzing the sequence of two user-type tasks in the experiment, preferred to use simple Forms, this must be because the students considered that using forms is more intuitive. Figure 3 shows the R3 transformation rule, Task B is a Service task. A result has to be displayed that is executed by an automatic operation. Task B allows you to map a Report (report) or a Datagrid (data grid). The students of the experiment preferred to use a Message box, but the students when analyzing a service type task, they prefer to show the results in Message box.

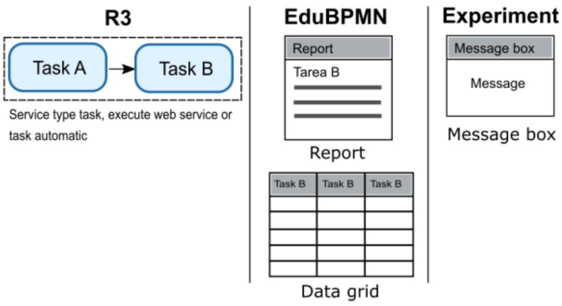

Fig. 3. Graphic components of the R3 transformation rule of the EduBPMN method and Experiment of the year 2019.

Figure 4 shows the R7 transformation rule that is used in the simple type event, this event allows to map a Hyperlink or Menu bar, but most of the students when analyzing the simple type event in the experiment preferred to use a Form.

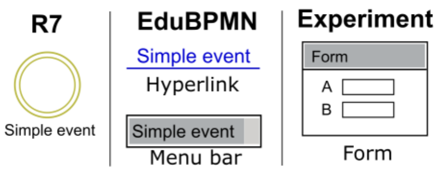

Fig. 4. Graphic components of the R7 transformation rule of the EduBPMN method and Experiment of the year 2019

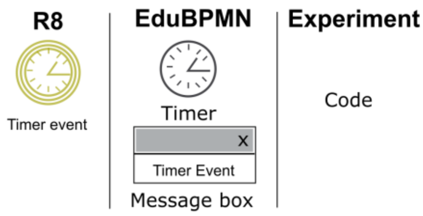

Fig. 5. Graphic components of the R8 transformation rule of the EduBPMN method and Experiment of the year 2019

Figure 5 shows the R8 transformation rule that uses a timer event, which allows mapping a Timer (clock) or Message box (message box). The students, when analyzing the timer event in the experiment, preferred not to use any graphic component, and only use code (textual description), which for them is very common to represent time in code. In addition, for the R0 transformation rule that consists of three attributes of the UML class diagram, more graphic components were assigned for five attributes according to the data type, these data types are: (Date, Char, Real, Media, URL). This was based on the analysis of Bizagi projects that contain UML class diagrams and we verified which graphical components map from the data attributes. Table 1 shows the incorporation of more attributes mapped to graphic components. It has been considered that a stereotype is assigned to each graphic component to avoid ambiguity in the extension of the BPMN model:

Table 1. R0 transformation rule with Data type and Graphics components.

Rules of EduBPMN Method	Date type	Graphics Components
R0	String	Text box
	Boolean	Check box
	Int	Combo box List box Acummulator Slider
	Date	Text box
	Char	Text box
	Real	Text box
	Media	Push button, linked to the media manager.
	URL	Link

Table 2. Summary of improved transformation rules.

Transformation rules	Graphics components	Stereotypes
R2	Form	<< form >>
R3	Message box	<< message >>
R7	Form	<< form >>
R8	Code	-
R0	Text box	<< text >>
	Check box	<< check >>
	List box	<< list >>
	Acummulator	<< acummulator >>
	Slider	<< slider >>
	Push button	<< media >>
	Link	<< link >>

Table 2 shows a summary of the improved rules of EduBPMN with its graphic components. Each graphic component was assigned a stereotype by the researchers, taking into account that in this document a new experiment will be carried out with these transformation rules. For the R2 and R7 transformation rules, the << form >> stereotype is used to generate a form, for the R3 transformation rule the << message >> stereotype is used to generate a Message box, only for the R8 transformation rule the stereotype will not be used since the code will have to be done manually, for the R0 rule the following stereotypes will be used: (i) << text >> stereotype to generate a Text box, (ii) << stereotype check >> to generate a Check box, (iii) stereotype << combo >> to generate a Combo box, (iv) stereotype << list >> to generate a List box, (v) stereotype << accummulator >> to generate an Accumulator, (vi) stereotype << slider >> to generate a Slider, (vii) stereotype << media >> to generate a Push button, (viii) stereotype << link >> to generate a links.

5 Planning and Definition of the Experiment

This section describes the experiment to be able to validate EduBPMN rules that were modified by the 2019 experiment for the R2, R3, R7 and R8, in addition to the R0 rule. This section is structured as follows: first it shows the research questions with their hypotheses, then the experimental design that is detailed with its results.

5.1 Research Questions and Hypotheses

This section shows the research questions with the prefix name (PI).
PI1: What is the correctness of the transformation rules regarding how subjects design graphical components from a BPMN model supplemented with a UML class diagram? Correctness is defined as the degree to which a system or component is free from faults in its specification, design and implementation according to the IEEE [30]. To address this property, we want to analyze the hypothesis H1: The rules used by the subjects are similar to the proposed rules.
PI2: What is the satisfaction of the transformation rules by the subjects to map BPMN to graphical components? Satisfaction is defined as satisfaction and positive attitudes towards the use of a product [31]. We measure satisfaction in terms of how comfortable developers feel while building a system. To address this research question, we want to analyze hypothesis H2: Subjects perceive as useful the use of any transformation rule to map BPMN to graphical components.

5.2 Experiment Method

Response variables and their metrics. The experiment uses two response variables: one variable to evaluate the correctness of the transformation rules of the EduBPMN method (PI1), and another variable to measure the satisfaction of the subjects regarding the mapping of BPMN to graphical components (PI2). The correctness of the transformation rules (PI1) is measured as the percentage of the transformation rule mapping that the subjects would use of our rules without knowing it. For example, if a subject maps the rules R2, R3 of the provided model, while our proposal is to use R2, R4, this means that they agree on a rule that is proposed, therefore, the correctness for the subjects is: 1 / 2 x 100% = 50%, the value close to 100% means that the subjects apply the same rules that we propose.

Rule correctness = Number of rules used by a subject of the reference set x 100% / Total number of rules used by a subject

The satisfaction of the transformation rules (PI2) is measured as the numerical sum of the assigned values of the sixteen definitions of the Moody's framework [13] based on the work of Lindland's [14] which is a widely used and validated framework. Each definition contains a response group based on a Likert scale: 1=Totally disagree, 2=Fairly disagree, 3=Neutral, 4=Fairly agree, 5=Totally agree. The result of the sum of the values is measured in the following ranking: 1 – 16 = Totally disagree, 17 – 32 = Fairly disagree, 33 – 48 = Neutral, 49 – 64 = Fairly agree, 65 – 80 = Totally agree. For this variable, we do not measure the satisfaction of the subjects with respect to our proposed rules, but what is the satisfaction on the part of the subjects if, in general, the concept of using transformation rules to map from BPMN to graphical components existed, this is important to know if the use of transformation rules is necessary. For example, a subject answers 10 questions with Totally agree = 5, and 6 with Fairly agree = 4, the total result is: (10 x 5) + (6 x 4) = 74 (Totally agree).

Experiment problems. The experiment has an experimental problem, this problem is small to avoid fatigue of the subjects and limit the experiment with a duration of one hour. The experimental problem is described below. The BPMN model is structured in four sections, where each section intends to generate a graphic component, where the subject does not know these proposed graphic components.

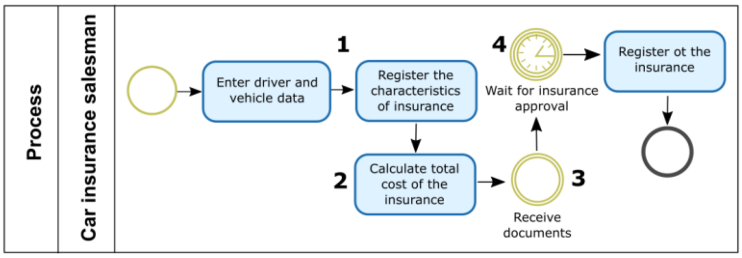

Fig. 6. BPMN model of the experiment problem

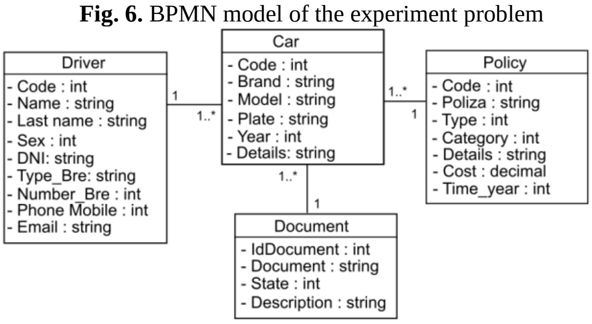

Fig. 7. UML class diagram of the Experimental Problem

Figure 6 shows the four sections: Section 1, are the user type tasks “Enter driver and vehicle data” and “Register the characteristics of the insurance”, Section 2, is the service type task “Calculate total cost of the insurance”, Section 3 is the simple type event "Receive documents", Section 4 is the timer type event “Wait for insurance approval”. Figure 7 shows the UML class diagram of the experimental problem, the classes are Driver, Vehicle, Policy and Document, each class has its attributes and data types.

Experiment procedure. The procedure for the experiment was structured in a one hour session.

Introduction to BPMN*:** Before the subjects carried out the experiment, they were given a document about the elements of the BPMN model two weeks in advance. In addition, before the experiment, an explanation was given to the subjects about this document with a duration of 10 minutes. ***Fill in a test: *Before developing the experiment*: the subjects filled out a test about questions from the BPMN model. This serves so that only subjects with knowledge of the BPMN model can develop the experiment. The test consists of eight questions about the BPMN model with four answer alternatives, and one possible correct answer. Each correct question is worth one point, and each wrong question is worth 0 points. We consider that subjects with a score greater than 6 points could carry out the experiment. All 31 subjects exceeded a score of 6 points, therefore, all subjects were considered fit for the experiment. ***Fill out a demographic questionnaire*:** Each subject recorded their data such as mail, age, sex and signed a consent form to carry out the experiment. ***Solve the experimental problem:*** The subjects developed the experimental problem in a 40-minute session, they mapped graphical components from a BPMN model complemented with a UML class diagram, they did not receive any guidance on how they should map from BPMN to graphical components. ***Fill out the post-questionnaire:*** Finishing the experimental problem, the subjects had to fill out an online questionnaire about the satisfaction of the transformation rules in general, the questionnaire lasts 10 minutes.

Subjects. The subjects for the rules experiment are considered end users, it was based on subjects who are not professionals but university students who know the field of Human Computer Interaction (HCI). In this experiment, the objective is to evaluate subjects so that they can map from BPMN to graphical components, the subjects were students, because experiments with real designers are expensive to involve. Furthermore, using visual editors would also be expensive and would have a different visual appearance with respect to the objective of our experiment. The sample of students was from the Software Engineering Degree of the Peruvian University of Applied Sciences. There were 31 subjects (25 men and 6 women, 28 with ages in the range of 17 – 20, 3 with ages in the range of 21 – 24, M = 19, SD = 1.39) with low knowledge in BPMN models, but with high knowledge in graphic component design. All participated voluntarily, they were instructed to develop the experiment and draw the graphical components with paper and pencil from a BPMN model and a UML class diagram. Table 3 shows the results of the subjects about the knowledge of the BPMN model, UML Class Diagram and Graphic Components, where we can say that the subjects before the experiment had basic knowledge about the BPMN model, for which they were trained. It must be taken into account that they did have medium and high knowledge of the UML class diagram and the design of the graphic components.

Table 3. Knowledge of BPMN, UML Class diagram, and Graphical Components

Knowledge of	Nothing	Low	Half	High
BPMN model	0	20	9	2
UML Class diagram	0	13	17	1
Design of graphics components	0	2	13	16

6 Results

The metrics were calculated through a Microsoft Office Excel 2020 sheet in an anonymous format so that the subjects could not be identified.

6.1 Correctness of transformation rules

The subjects drew graphical components from a BPMN model and a UML class diagram, where they were compared with the transformation rules of the EduBPMN method.

Table 4. Correctness percentage of transformation rules

Experiment Problem	Correctness percentage
Section 1	100%
Section 2	50 %
Section 3	100 %
Section 4	100 %
Average	87.50 %

Table 4 shows the percentage of correctness of each of the four sections of the experimental problem, the sections of the experimental problem that had the best results were sections 1, 3 and 4, with a value of 100% correctness. This suggests that the graphical components that the subjects mapped are the same graphical components that we propose in the improved transformation rules of EduBPMN. For Section 1, most of the subjects used a Form similar to the one proposed by the R2 that is complemented with the fields of the attributes of the UML class diagram using the R0. Section 3, most of the subjects use a Form, being a graphic component similar to the one we propose that is complemented by R0. For Section 4, the subjects did not use any graphic component but used code (textual description). The rule that obtained the worst results was section 2 with a value of 50% correctness. This suggests that for the subjects, this section was not very intuitive to generate a graphic component similar to the one we propose with our rules. Subjects prefer to use a Form instead of a Message box, as proposed in our proposal. This suggests that the result of the correctness of the transformation rules is positive with 87.50%. Therefore, most of the subjects used the same improved rules that we have proposed.

Table 5 shows the frequency of the transformation rules that the subjects used in each section of the experimental problem and the proposal of the transformation rules of EduBPMN of the document. For Section 1 they agree with the two proposed transformation rules R0 and R2, for Section 2 they only agree with one transformation rule R0, the subjects used Another Rule (OR), for Section 3 they agree with the two transformation rules R0 and R7, for Section 4 the subjects prefer to use code where they agree with the transformation rule R8.

Table 5. Correctness percentage of transformation rules

Section	Rules used by subjects	Proposed rules EduBPMN Method
Section 1	R0 (31), R2 (31)	R0 – R2
Section 2	OR (31), R0 (31), R3(0)	R0 – R3
Section 3	R0 (31), R7 (31)	R0 – R7
Section 4	R8 (31)	R8

6.2 Satisfaction of transformation rules

Satisfaction is measured in terms of Perceived Ease of Use (FUP), Perceived Utility (UP) and Intention to Use (UI), on a 5-point Likert scale (the higher the score on the scale, the greater the satisfaction). It is measured through a questionnaire structured as follows (6 questions for FUP, 8 questions for UP, 2 questions for UI).

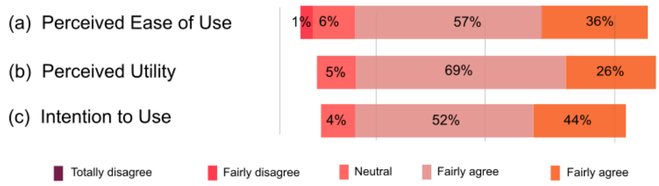

Fig. 8. Distribution of the results of the questionnaire questions.

Figure 8 shows a divergent stacked bar of the answers provided by the subjects to the transformation rules questionnaire, regarding the Perceived Ease of Use (FUP) approximately 36% totally agree and 57% quite agree (93% total). Therefore, most of the subjects consider that the existence of transformation rules is important where they can be useful and easy to use to map from BPMN to graphical components. Regarding Perceived Utility (UP), approximately 26% fully agree and 69% quite agree (95% total). Therefore, most of the subjects find the use of transformation rules to map BPMN to graphical components useful. Regarding the Intention of Use (UI), approximately 44% fully agree and 52% quite agree (96% in total). Therefore, most of the subjects consider that they would intend to use transformation rules to map BPMN to graphical components. It must be taken into account that in FUP (6% neutral), UP (5% neutral) and UI (4% neutral), only a minimum percentage consider that the use of transformation rules would be indifferent. We conclude that the subjects perceive as useful the use of any transformation rule to map BPMN to graphical components.

7 Discussion

This section reviews the results of the experiment, for each variable the results are discussed. The results of the Correctness of the transformation rules show that the subjects use the same rules that are proposed in this article. The transformation rules that yielded the best results are the R2, R7 and R8. For the R2, most of the subjects consider that from a set of user-type tasks in a sequential manner it must be transformed into simple forms, with the fields that are extracted from the UML class diagram, unlike using Wizard, Tabbed dialog box, or Group bx as proposed above, this suggests that the subjects with sider that it is more common and intuitive to use a form for the rule. For the R7, most of the subjects prefer that from a simple event type it be transformed into a form because this allows data to be recorded, this suggests that the subjects do not intend to use Hyperlink or Menu bar as previously proposed, but rather a form with its data fields. For the R8, most of the subjects prefer that from a timer event the representation must be in code using time intervals so that it can perform an action, taking into account that the use of time intervals as a graphic component is not frequent except in the code. The transformation rule that obtained the worst results was the R3, this because it is ambiguous and interpretation is needed to represent it in a specific graphic component. For this R3, most of the subjects prefer that from a service type task it is transformed into a form, because this allows to display a result that is processed in the service type task, this suggests that the subjects do not intend to use a Message box as proposed in this article, so it is not very intuitive. The results with the Satisfaction of the generalization of the rules show that the subjects consider that in terms of Perceived Ease of Use (FUP), the existence of transformation rules to map BPMN to graphic components with easy-to-use and subject-friendly characteristics would be very important, the majority chose quite agree and totally agree. In terms of Perceived Utility (UP) it shows that the majority of the subjects consider that the transformation rules would improve the development of graphic components, being useful for the designer so that he can choose and modify design alternatives. In terms of Intent to Use (UI) it shows that the subjects would intend to use transformation rules, only a small part of the subjects would be hesitant to use.

8 Threats to validity

This section shows the validation of the threats that could affect the development of the subjects in the experiment. Threats are described according to Wohlin's classification [32]. For each group of threats they will be described according to the classification of the group of threats within four types:

Validity at conclusion. This type of threat refers to the ability to draw the correct conclusion about treatment and outcome relationships. The experiment can suffer from the following threats of this type: Random heterogeneity subjects, which means that there is always heterogeneity in a study group. To minimize this threat, we have recruited subjects who are students of the Software Engineering career who have the same profile where they took courses on Human-Computer Interaction and graphic components. The subjects had to register a demographic questionnaire where we can know that they have similar profiles. **Internal validity.** This type of threats analyzes the influences that can affect the causality factor. The experiment may suffer from the following threats of this type: History, which means that differences may arise when treatments are applied at different times. To minimize this threat, the experiment was carried out in one session lasting one hour. Another threat that can appear is Subjects' Experience, which means that the subjects' experience is not enough to perform the experiment. To minimize this threat, subjects have a year and a half practice programming graphic components. Regarding the BPMN model, the subjects were trained with a guide document and training before starting the experiment. **Construct validity.** This type of threat refers to generalizing the result of the experiment to the concept or theory behind the experiment. The experiment can suffer from the following threats of this type: Hypothesis Guessing, which means that when people participate in the experiment, they can try to figure out the purpose of the experiment and act to improve its results. To minimize this threat, the researchers did not mention any data about the experiment's research questions. **External Validity.** This type of threat refers to conditions that limit our ability to generalize the results of our experiments to industrial practice. The experiment can suffer the following threats of this type: Interaction of the environment and the treatment, this means that the effect of not having the experimental environment or material representative of industrial practice. The experiment is run in an academic setting and the results can only be generalized in such a setting. Another threat that can appear is the Interaction of the environment and the treatment, which means that the effect of having a population of subjects that is not representative of the population that we want to generalize. With results obtained from the subjects, it is not possible to generalize and guarantee that the results are valid for other experiments with subjects with a different profile.

9 Conclusions and future work

This article presents four improved transformation rules for EduBPMN that allows the generation of graphical components design alternatives from a BPMN model complemented with the UML class diagram. The transformation rules that were improved are the following: the transformation rules R2, R3, R7, R8 were improved from a 2019 experiment with students from the University of Valencia (Spain). The graphical components that most of the students have preferred in that experiment were used. The graphical components of the R0 transformation rule were extended based on Bizagi projects that have a UML class diagram. The new data attributes were identified to see what graphical components they generate. Using the five improved rules of EduBPMN, this article ran an experiment to test them. This experiment is based on two metrics: (i) correctness of the transformation rules and (ii) satisfaction of the generalization of the transformation rules. The result of the correctness of the transformation rules shows that most of the subjects have drawn graphical components from the BPMN complemented with the UML class diagram, where these graphical components are similar to the graphical components of the improved transformation rules proposed in this article. In addition, the satisfaction result shows positive results where the subjects consider that the existence would be useful and would intend to use transformation rules to map from BPMN to graphical components in a general way. The experiment suffers from the following limitations: (i) to develop experiments with more complex projects of companies with several processes, (ii) the other transformation rules of EduBPMN were improved from the results of the 2019 experiment. As future work, the following has been considered: (i) develop more experiments with various subjects in order to obtain a family of experiments and analyze their data; (ii) carry out experiments with more complex BPMN projects.

Acknowledgements

This work has been developed with the help of the Generalitat Valenciana through the GENI project (CIAICO/2022/229) and the PDC2021-121243-I00 project financed by the Spanish Ministry of Science and Innovation.

References

1. S. J. Mellor, T. Clark, and T. Futagami, "Model-driven development: guest editors' introduction," IEEE software, vol. 20, pp. 14-18, (2003).
2. 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, pp. 1-12 (RCIS), (2018).
3. 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).
4. E. Díaz, J. I. Panach, S. Rueda, and J. Vanderdonckt, "An empirical study of rules for mapping BPMN models to graphical user interfaces," Multimedia Tools and Applications, vol. 80, pp. 9813-9848, (2021).
5. E. Díaz, J. I. Panach, S. Rueda, and D. Distante, "A family of experiments to generate graphical user interfaces from BPMN models with stereotypes," Journal of Systems and Software, vol. 173, p. 110883, (2021).
6. E. Díaz and J. I. Panach, "New Transformation Rules for the EduBPMN Method to Generate Graphical User Interfaces from BPMN," 2022 Third International Conference on Information Systems and Software Technologies (ICI2ST), Quito, Ecuador, 2022, pp. 164.
7. J. E. Díaz Suárez, "EduBPMN: Un método Basado en Reglas de Transformación para Generar Interfaces Gráficas de Usuario a partir de Modelos de Procesos de Negocio (BPMN)," (2020).
8. Ó. Pastor, E. Insfrán, V. Pelechano, J. Romero, and J. Merseguer, "OO-Method: An OO Software Production Environment Combining Conventional and Formal Methods," Lecture Notes Computer Science. 9th Conference Advanced Information Systems Engineering (CAiSE'97), A. Olive and J. A. Pastor, Eds., ed Barcelona, Spain: Springer-Verlag, (1997).
9. BPMN. (2013). Business Process Modeling Notation. Available: http://www.bpmn.org
10. D. Berardi, D. Calvanese, and G. De Giacomo, "Reasoning on UML class diagrams," Artificial intelligence, vol. 168, pp. 70-118, (2005).
11. Bizagi. (2002). Bizagi. Available: https://www.bizagi.com/es

12. V. Paradigm, "Visual Paradigm," (2019).
13. 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).
14. O. I. Lindland, G. Sindre, and A. Solvberg, "Understanding quality in conceptual modeling," IEEE Software, vol. 11, pp. 42-49, (1994).
15. 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, pp. 143-163 LNBIP, ed, (2010).
16. J.-M. Favre, J. Estublier, and M. Blay-Fornarino, L'ingénierie dirigée par les modèles: au-delà du MDA: Hermes-Lavoisier, (2006).
17. V. Torres and V. Pelechano, "Building business process driven web applications," , pp. 322-337, in International Conference on Business Process Management (2006).
18. B. H. Lam, V. T. H. Nguyen, C. H. Phan, and T. T. T. Truong, "An approach for application generation based on BPMN," in 2020 12th International Conference on Knowledge and Systems Engineering, pp. 115-119 (KSE), (2020).
19. 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).
20. 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).
21. 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, pp. 415-429 (2010).
22. E. F. Cruz and A. M. R. da Cruz, "Deriving Integrated Software Design Models from BPMN Business Process Models," in ICSOFT pp. 605-616, (2018).
23. 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).
24. 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).
25. M. Zur Muehlen, "Organizational management in workflow applications–issues and perspectives," Information Technology and Management, vol. 5, pp. 271-291 (2004).
26. I. Abouzid and R. Saidi, "Proposal of BPMN extensions for modelling manufacturing processes," in 2019 5th International Conference on Optimization and Applications (ICOA) , pp. 1-6 (2019).
27. B. Intrigila, G. Della Penna, and A. D'Ambrogio, "A Lightweight BPMN Extension for Business Process-Oriented Requirements Engineering," Computers, vol. 10, p. 171 (2021).
28. J. Vanderdonckt and T.-D. Nguyen, "MoCaDiX: Designing cross-device user interfaces of an information system based on its class diagram," Proceedings of the ACM on human-computer interaction, vol. 3, pp. 1-40, (2019).
29. B. BPMN. (2017). Business model patterns. Available: http: //resources.bizagi.com/docs/Workflow_Patterns_using_BizAgi_Process_Modeler_Esp.pdf
30. "ISO 9241-11: Guidance on Usability," ed, (1998).
31. IEEE, "Systems and software engineering -- Vocabulary," ISO/IEC/IEEE 24765:2010(E), Ed., ed, pp. 1-418 (2010).
32. C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, Experimentation in Software Engineering: An Introduction: Springer, (2012)