Multimed Tools App manuscript No.
(will be inserted by the editor)

An Empirical Study of Rules for Mapping BPMN Models to Graphical User Interfaces

Eduardo Díaz · José Ignacio Panach · Silvia Rueda · Jean Vanderdonckt

Received: 2019-08-26 / Accepted: date

Abstract **Context:** Generation of Graphical User Interfaces (GUIs) from Business Process Model Notation (BPMN) models is a step manually performed by an analyst in any information system development. By analyzing twelve BPMN projects and comparing them with their associated GUIs, a set of rules for mapping BPMN models expressed in terms of BPMN patterns to GUIs has been identified. **Objective:** This paper provides three main contributions: an empirical validation of these mapping rules; a classification of their alternatives to identify which ones support usability; and the validation of this classification. **Method:** We conducted an experiment to study whether 43 participants apply the same rules as previously identified with response variables as: correctness of the rules, their completeness, perceived usefulness and intention to use. To select the mapping rules alternatives that effectively impact usability, we classified them according to empirically validated usability guidelines found in the literature. We also validated them with participants and response variables as: correctness of the guidelines, their perceived usefulness and intention to use. **Results:** Correctness of the mapping rules was assessed positively (74%), as well as

Eduardo Díaz
Escola Tècnica Superior d'Enginyeria, Departament d'Informàtica, Universitat de València, Avenida de la Universidad, s/n, 46100 València, Spain
E-mail: diazsua@alumni.uv.es

José Ignacio Panach
Escola Tècnica Superior d'Enginyeria, Departament d'Informàtica, Universitat de València, Avenida de la Universidad, s/n, 46100 València, Spain
E-mail: joigpana@uv.es

Silvia Rueda
Escola Tècnica Superior d'Enginyeria, Departament d'Informàtica, Universitat de València, Avenida de la Universidad, s/n, 46100 València, Spain
E-mail: silvia.rueda@uv.es

Jean Vanderdonckt
LouRIM Institute, Université catholique de Louvain, Place des Doyens, 1, B-1348 Louvain-la-Neuve, Belgium
E-mail: jean.vanderdonckt@uclouvain.be

their completeness (76%), perceived usefulness (95%) and intention to use (68%). Correctness of the usability recommendations for using these mapping rules was also assesses positively (57%), as well as their perceived usefulness (86%) and intention to use (72%). **Conclusion:** This paper provides analysts with effective and efficient guidance on how to apply them consistently and to feed a software for semi-automatic transformation of BPMN models into their corresponding GUIs.

Keywords BPMN patterns · Business Process Model and Notation · Experimental study · Graphical User Interfaces · Mapping rules · Usability Engineering

1 Introduction

Business Process Model Notation (BPMN) provides businesses with the capability of understanding their internal business procedures in a graphical notation and gives to organizations the ability to communicate these procedures in a standard [42]. This model is widely used to elicit requirements in large systems with many processes. Examples of the wide use of BPMN models are: the definition of a BPMN-based framework that supports concepts of security requirements [49]; definition of mapping users/organizations requirements with BPMN model [17]; generation of a BPMN model from textual requirements [38]; definition of a mapping of corporate communications based on BPMN models [47], and other works for eliciting requirements of complex systems as [13, 8, 29, 36]. There are other models to elicit requirements such as UML Activity Diagrams, UML Use Case Diagrams or workflows, among others. This paper focuses on BPMN since we target to deal with large and complex systems, and according to the BPMN definition, this models is the most suitable for that goal. The conceptual primitives brought by BPMN models are: *event*, *gateway*, *lane*, *flow* and *tasks* [42]. Two types of tasks are mainly used [40]: *user task*, carried out by a person or user with the help of a system and *service task*, carried out by a system without human intervention, for example web services or an automated application.

Currently, Graphical User Interfaces (GUIs) are generated from Business Process Model Notation (BPMN) models manually. The analyst applies mapping rules from model to GUI subjectively in the information systems development. Consequently, the quality of the resulting GUIs entirely depends on the analyst's experience who is responsible for maintaining usability, consistency across models, etc. Moreover, a BPMN model expresses by definition some dynamic aspects of processes, thus containing little or no information for ensuring this mapping and leaving the analyst with little or no guidance on how to ensure this mapping. Other stakeholders are also concerned as analysts who built the BPMN models are not necessarily the same people as designers who design the corresponding GUI or developers who program them. To bridge the gap between the expressiveness of BPMN models and their mapping to GUIs, this paper suggests a set of rules for mapping BPMN models, together with pointers to a UML Class Diagram, to GUIs of an information system supporting the associated processes.

Fortunately, several BPMN modelers, such as Bizagi [3], AuraPortal [37], BonitaSoft [4], e-Citiz [55] and WebRatio [1], offer some facility to (semi-)automatically generate GUIs from the designed BPMN models. But the transformation logic they rely on is not explicitly represented anywhere nor empirically validated, thus leaving the end user with some uncertainty about their resulting quality. Also there is an Architecture of Integrated Information System (ARIS) [2] which allows mappings from BPMN to Business Process Execution Language (BPEL) [25]. In addition, these BPMN modelers require additional models and/or model fragments to compensate the lack of expressiveness of the initial BPMN models towards their GUIs, thus requiring an additional modelling effort and diving into the risk of model proliferation. The level of details is largely varying from one modeler to another: only a few features in WebRatio until the full specifications of a UML Activity Diagram associated to each BPMN model in e-Citiz. This modeler even enables the designer to select widgets for each data attribute involved in the Activity Diagram.

In this work, we focus on defining mapping rules to generate GUIs from BPMN models extended through stereotypes, that enhance the models with enough expressiveness to create the GUIs automatically. This contribution is different from the other approaches that propose GUIs mapping from BPMN models (as WebRatio or e-Citiz among the others previously mentioned), which are based on complementing BPMN models with other abstract models. Note that our approach focuses on GUIs generation, how to map the behavior expressed in BPMN models is out of topic. There are many previous works that have tackled how to generate code expressing the behaviour of BPMN models, such as [20], but a few that deal with the GUI generation. So this paper is a step forward to cover the gap of existing works that deal with the GUI generation.

We opted for extending BPMN models through stereotypes instead of complementing BPMN with other models such as WebRatio or ARIS do (among others) because the idea of our approach is to profit from the effort spent building BPMN models. Note that BPMN models are usually used to elicit requirements of large systems, where there are many primitives to represent in the BPMN model. With our approach, all the effort spent drawing BPMN models involves the generation of GUIs with no effort if we automate the mapping rules.

In our approach, in order to identify the set of mapping rules from BPMN patterns, we analyzed twelve real-world Bizagi BPMN projects belonging to various domains of human activity such as administration, educational software, management, and sales [16]. The most widely and frequently used BPMN patterns identified were: sequence, exclusive decision, synchronization, and implicit decision. For each such BPMN pattern, we defined a set of rules for establishing the mapping BPMN-GUIs with specific indications for deciding which alternative is decided for selecting widgets of the GUIs. When many alternatives are offered for a mapping rule, the concerned BPMN primitives are extended with stereotypes specifying which mapping alternative is decided. The first contribution brought by this paper concerns the empirical validation of the mapping rules resulting from this identification [16]. We conducted an experiment with 43 participants to assess the correctness [23] of the mapping rules, their completeness [35,65], their perceived usefulness [15] and intention to use [15]. The experiment was conducted through two experimental problems. By starting from a BPMN model along with its textual description and link to its Class diagram, participants had to design GUIs corresponding to this model without knowing our mapping rules. Through correctness and completeness, the validity of our mapping rules is assessed. Through perceived usefulness and intention to use, we analyze the general concept of a mapping rule, since participants were not aware of our mapping rules.

The second contribution concerns the definition of usability recommendations for choosing the best alternative for mapping rules with more than one alternative. Only such rules require stereotypes. Relying on these recommendations provide analysts with guidance on how to apply mapping rules to ensure usability of the resulting GUIs without any prior experience or knowledge in usability engineering. These recommendations have been extracted from previous works in usability engineering, such as [59,21,39,44,62].

The third contribution consists in the validation of these usability recommendations in terms of correctness of the usability recommendations, perceived usefulness and intention to use of the usability recommendations. The results of the experiment suggest that:

1. Regarding the measures of correctness and completeness, the mapping rules used by most participants remain the same as our proposed rules.
2. Regarding their perceived usefulness and intention to use, a high acceptance rate is obtained for using these rules to automatically generate GUIs from BPMN models and for the usability recommendations.
3. Regarding correctness of the usability recommendations, they agree with participants' preferences to support usability.
4. The existence and the validation of these mapping rules are considered as a positive and valuable contribution to provide analysts and other stakeholders with guidance on how to (semi-)automatically generate GUIs from BPMN models.

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

2 Related Work

In this section, we review work related to mapping BPMN models to GUIs by running a Targeted Literature Review (TLR), a non-systematic, in-depth and informative literature review aimed at keeping only the significant references maximizing rigorousness while minimizing selection bias. In addition, our goal is not to compare all work involving some form of mapping or generation of GUI based on BPMN, but instead to identify the most relevant references. For this purpose, the semantic question of mapping BPMN-GUI is translated into the following syntactical query used as a search string on Scopus digital library[1]:

[1] See `https://www.scopus.com/home.uri`

``graphical user interface'' AND (``BPMN'' OR ``model'' OR ``code'').

References resulting from this search were classified into three categories, which are further discussed in the next subsections.

2.1 Graphical User Interface Generation from BPMN models

Since a BPMN model only expresses some dynamic aspects of business processes and nothing particular about GUIs, it needs to be augmented or complemented with additional information that is useful to map BPMN to GUIs and that should be found elsewhere than in BPMN. This information is quite heterogeneous from one source to another. For example, business processes are described through the BPMN notation [42] extended with information on task assignment, escalation, policies, activity semantics and typed data flows [9]. WebRatio BPM [10] provides various features increasing the productivity and the quality of the resulting application: one click generation of a running prototype of the process from the BPMN model. Instead of adding features, the BPMN model could be itself extended, such as with a set of control flow patterns and data flow patterns [22]. These extensions remain outside the world of Human-Computer Interaction (HCI), which is not the case of some other works. For instance, Model-Driven Engineering (MDE) could be exploited for this purpose by progressively transforming the initial model into a GUI model [57] in four steps structured according to the Cameleon Reference Framework (CRF) [12]: (1) modelling business process modeling in their organizational context; (2) deriving a task model from the BPMN model; (3) refining the task model for interaction; and (4) deriving a GUI user interface model derivation from the task model. Although this method could be mostly automated, it does not offer a wide range of GUI possibilities: only one GUI is generated for each BPMN. Therefore, rules [56] could be added based on usability guidelines to expand the range of potential GUIs to be obtained.

Some of these rules are obviously more frequently triggered than others, thus enabling patterns to merge for transforming fragments of business processes, whose activities are performed by the same user, to GUIs of the process-aware information systems [27]. GUI logic, which includes local interaction and navigation and domain rules can be expressed using the same notation as for process modeling [28]. This approach applies these patterns automatically to derive user interfaces by establishing a bidirectional mapping between process model and their GUIs, which are predefined. Similarly to [57], a task model expressed in CTT can be directly associated to a BPMN [46], which is then used to produce GUIs based on web services. On the contrary, BPMN++ [61] argues that no task model is required, but that BPMN could simply enhanced with more expressiveness captured in a dialog model specified in Diamodl to make the correspondence with the BPEL part.

Instead of willing to progressively enhance BPMN and/or transform it, Usi4Biz [58] aligns business process in BPMN to GUIs expressed in UsiXML [34] in two steps: (1) defining association of business process with GUI model, and (2) presenting a tool for model transformation that addresses traceability. The transformation process that is typically found in MDE could also be released, e.g., by relying only on a model-based approach for generating web GUIs based on Artifact-Centric Process (ACP) modeling approach [68]. In this framework, the artifact centric business process model and the User Interface Flow Model (UIF) are defined with a mechanism to derive the UIF model from the ACP model. The UIF model reflects the logic of business processes and intuitively represents what information is required during the process. Based on this process, a framework [69] is developed for deriving user interface flow models to help visualize artifact centric processes and support semi-automatic creation of GUIs. The GUI model is created by taking into account the relations among business process, GUIs and user roles in an artifact centric process model. Algorithms derive user interface flow model from an artifact centric process model. A process design methodology [10] supported by a tool suite address the extension of business process with social features, the social process design exploits an extension 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 (M2M) and Model-to-Code (M2C) transformation technology that automatically produces a process enactment web. Business processes could be also captured in a business requirements model [72] from which task models are directly derived instead of added. By combining them, GUIs are generated using two other intermediate models for dialog and presentation. The M2M transformations are inspired by user interface design principles and task patterns, which reinforce GUI consistency across applications and transition guidance. There is a model-driven approach [14] to support the construction of a use case model, an integrated domain model, and a user interface model, from a set of BPMN models, comprising all existing information in those models. In [51], a conceptual mapping of supported concepts is presented and technically implemented using a Question/View/Transformation (QVT-O) [43], which uses an operational approach to demonstrate an automated mapping between BPMN and Munster App Modeling Language (MAML) [50]. Consequently, it is possible to simplify the automatic generation of mobile apps by reusing processes specified in BPMN. In [70], a modeling approach is introduced to represent Enterprise Resource Planning (ERP) processes through BPMN concepts. The approach consists of a set of transformation rules that generate functional executable Java web services with Services Oriented Architecture Modeling Language (SoaML).

All the above works follow a forward engineering approach. In contrast, the area of electronic commerce was the participant of such a reverse engineering of GUIs [7] to come up with a first identification of business processes emerging from these GUIs [71]. In this category, we observe the reliance of many different models, such as UML Activity Diagram [55], a data or a domain model [10,14,51,68], a task model [57,46,72] or no task model at all [61], a presentation model [72,70], a dialog model [61,72], a data flow diagram [69], some patterns, web-services [70], etc. The heterogeneity of the models involved in these approaches suggest that no consensus can be established on an effective and efficient way to map GUIs from the initial BPMN models. Some approaches even fall in the trap of model proliferation. All of them require the analyst, the designer, the developer to spend more resources and effort in drawing these additional models which are not directly related to business processes. On the contrary, our method prefers extending the BPMN model through stereotypes to produce GUIs without the need of learning other models.

2.2 Transformation of BPMN Models to WS-BPEL Code

As opposed to the previous category, this one does not rely on any extraneous model to gain more expressiveness for mapping GUIs, but exploit the dynamic aspects throughout BPEL and its BPEL engine. For example, the Business Process Diagram (BPD) [45] generates a BPEL block structure by applying three mappings: a well-structured component onto BPEL straightforwardly (a component is a subset of the BPD that has one entry point and one exit point); an unstructured, but acyclic, component is mapped onto a component control link-based BPEL code; an unstructured component which cannot be translated using control links is mapped to a special section. Similarly, a BPMN model can be turned into a specific language such as WS-BPEL, expressed at the Platform Specific Model (PSM) (is a model of a software or business system that is linked to a specific technological platform) level of MDE [19]. The Object Oriented Web Solutions (OOWS) (extension of an object-oriented software production method) conceptual model [60] adds expressiveness capturing the navigation and presentation requirements of web applications. From a BPMN model, the navigation across web pages and the WS-BPEL executable description that implements the entire process are derived. UniFlexView [67] is a unified framework that consists of a business process model, a process view model, and a comprehensive set of rules that can be used to guarantee the consistency between a derived process view and its original process model defined by WS-BPEL and BPMN.

In this category, we mainly observe techniques typically used for translating graphical, potentially unstructured or informal, contents to textual, more structured and formal languages so as to implement them straightforwardly. Transformations from BPMN to code WS-BPEL through components are emphasized [19,60], convert graphical/unstructured to textual/structured contents [45] and an unified framework [67]. On the contrary, our proposal does not perform component transformations since mapping rules produce code from stereotyped BPMN models.

2.3 Graphical User Interface Generation from other Models than BPMN

Model-based design of user interfaces (MB-UID) is distinguished from *Model-Driven Engineering* (MDE) of GUIs in that the former exploits one or many models to produce GUIs, but not necessarily based on explicit transformations governed by a meta-model among meta-models corresponding to input and output model in the latter. Both sub-categories adhere to the Cameleon Reference Framework (CRF) [12] that structures the GUI development life cycle into four steps similar to MDE: task and domain modelling, abstract user interface modelling, concrete interface modelling, and final user interface production.

For example, the UsiPXML (User interface pattern Extensible Markup Language) [48] framework extends UsiXML [34] with GUI patterns useful from designing the concrete user interface and generate the corresponding final GUI. As in the first category, a task model, together with other models, can be created within the context of already existing workflow models and linked with them [30]. Again based on MDE and CRF, a workflow can be mapped to supporting GUIs as follows [6]: a first transformation derives a plastic GUI from a workflow, a second transformation relies on BPMN for interaction modeling, and finally task model, abstract and concrete GUIs are obtained. This process can be made even wider and more in-depth based on GUI workflow patterns in FlowiXML (to represent the workflow) [**?**]: a workflow model defines what processes and tasks need to fulfill and their possible ordering, the processes are implemented through workflows, which in turn are decomposed into tasks. For each task, workflow GUI patterns are applied which give rise to final GUIs. Although this process is made systematic, there is only one possible GUI for each pattern, thus limiting the range of alternative GUIs for the same pattern. No choice is allowed to the designer. The UML Activity Diagram and OCL constraints (UML AD/OCL) are exploited in a method for modeling business processes [66]. AutoPa [32] similarly generates automatically an executable prototype with GUIs running in Java.

In this category, we observe that an extensive effort in research and development has been produced so that summarizing these approaches is beyond the scope of this paper. A recent comparison can be found in [52]. We again observe another heterogeneity of models, all different from BPMN, sometimes so far away that the concepts manipulated by these models could become even farther, if not stranger, to the analyst: patterns [**?**,48], workflows [6,**?**,30], task model [**?**,48], UML Activity Diagram [32,66]. Consequently, the analyst must learn the particularities of these extraneous models to generate GUIs. On the contrary, our proposal is aimed at producing GUIs only with BPMN and its related Class Diagram, more models to represent interaction characteristics are not required.

3 Method Used for Identifying Mapping Rules

3.1 BPMN Patterns as Input

BPMN is a language where patterns represent frequently used situations that commonly happen in any business process model [42]. These patterns help reusing solutions to face problems suffered in any business process. For this reason, we identify transformation rules from four such patterns and not for individual cases: sequence, exclusive decision, synchronization, and implicit decision. The following subsections detail them for a single user. Hence, the BPMN will hold in one swimlane. For a BPMN involving many users, the corresponding GUIs will be distributed as a suite of individual GUIs produced in the same way. The four patterns considered are:

– **Sequence Pattern**: occurs when completing a task *A* enables continuing with the next task *B* sequentially (see Fig. 1 for an example).

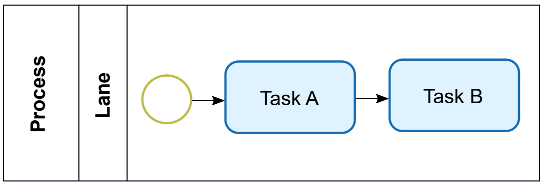

Fig. 1 Sequence pattern.

– **Exclusive decision pattern**: occurs when a single path must be chosen among several ones available depending on a decision or process data. The gateway is depicted by a rhombus (see Fig. 2 where, after the gateway, Task *B*, *C* or *D* can be chosen to continue with the process).

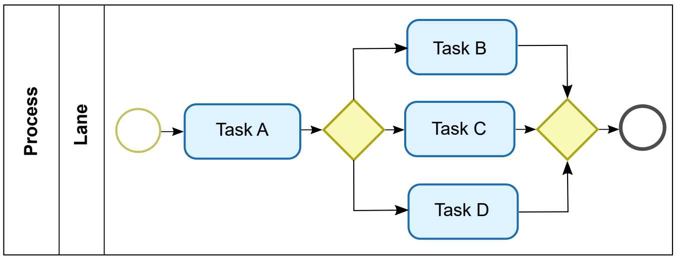

Fig. 2 Exclusive decision pattern.

– **Synchronization pattern**: occurs when two or more branches of the process are merged into one. All incoming branches are completed before continuing the next task. The gateway is depicted by a rhombus with a cross (see Fig. 3 where Tasks *B*, *C* and *D* must be completed after the gateway to continue with the process).

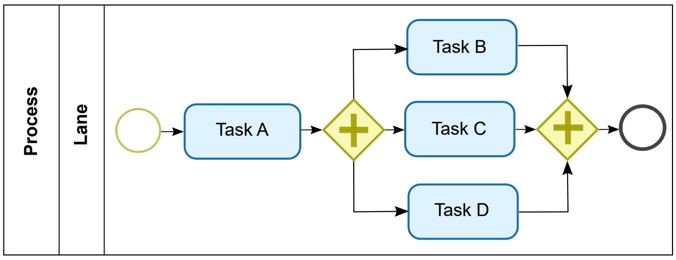

Fig. 3 Synchronization pattern.

– **Implicit Decision Pattern**: occurs when a branch must be selected from a set of several ones depending on the data of the process. When a branch is chosen, the other ones must be disabled. This gateway is depicted by rhombus with circles inside with two events (see `Event 1 (timer type)` and `Event 2 (simple type)` in Fig. 4, where the upper branch continues `Event 1` with Task *B*, and in the lower part, the gate continues `Event 2` which then continues Task *C*.

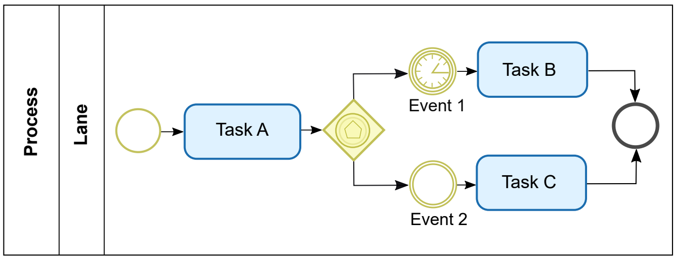

Fig. 4 Implicit decision pattern.

The BPMN model contains information about the system behavior but still lacks important information for the GUI mapping: the data, which are found in the UML Class Diagram. Thus, information about the GUI navigation and its behavior can be extracted from the BPMN model, while the information required in GUI is extracted from the Class Diagram. We argue that the UML Class Diagram has today become the most frequently used model for representing data through classes, their attributes and methods, and the relationships between these classes. Since 2000, the class diagram is the most favoured conceptual model by designers [11]. In an analysis of 121 open UML projects, all of them (100%) use class diagrams and half of them use cases (47%) [31]. One third of these projects only used one model (34%), 17% used two models, and 22% used three models.

3.2 Mapping Rules Definition

This section defines a set of nine rules for mapping the aforementioned BPMN patterns and models onto their corresponding GUIs with alternatives when any, i.e., `R0-R6` [3] and `R7-R8` [16], which are defined as follows:

- `R0`**. Generic rule for any pattern**. Each attribute extracted from the class diagram related to the BPMN model is mapped onto a widget depending on its data type: (1) a text box, for any string; (2) a list box or a combo box for any enumeration with simple choice, (3) a radio button or a check box for any Boolean. When a class A is related to a class B, its B objects are incorporated with three alternatives: (4) a combo box, (5) a data table, or (6) a list box.
- `R1`**. Generic rule for any pattern**. Any user task is mapped onto a form containing widgets corresponding to extracted attributes. The name of the user task is mapped onto the label of a push button included in the form.
- `R2`**. Sequence pattern**. User tasks with a dependency among them in the same swimlane are mapped onto a GUI providing the end user with some guidance to carry out these tasks, with three alternatives (Fig. 5): (1) a wizard, when the navigation throughout the different forms corresponding to each user task is ordered in a sequence with the possibility to go forward and backward; (2) a tabbed dialog box where each tab contains the form corresponding to each user task, when there is no particular order between them; (3) a group box, when a limited number of user tasks can be grouped in the same form.

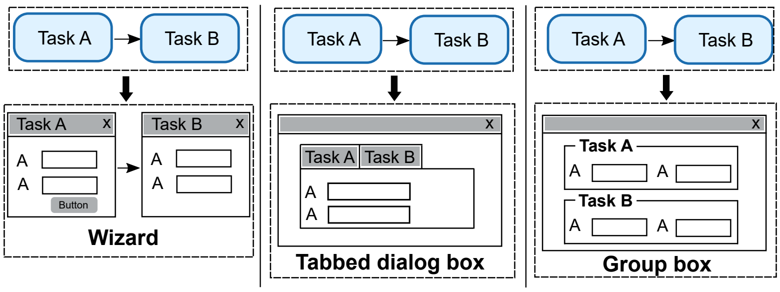

Fig. 5 The three alternatives of R2.

– R3. **Sequence pattern**. A user task *A* having a dependency with a service or automatic task *B* is mapped onto a form, while *B* is mapped onto two alternatives (Fig. 6): (1) a report, when information should be presented in a way that is ready to be visualized or printed depending on the task goal, (2) a data grid, when information should be presented according to its data structure.

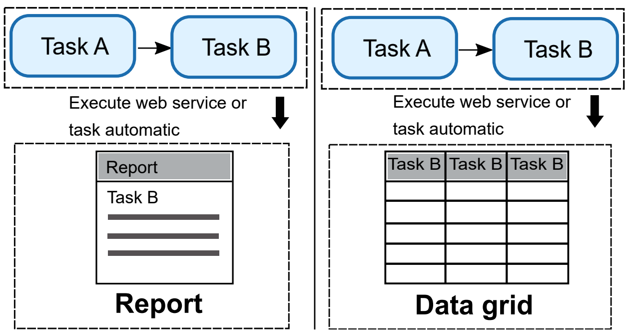

Fig. 6 The two alternatives of R3.

– R4. **Exclusive decision pattern**. The text in the exclusive decision gateway is mapped onto a label phrased as a question to initiate an end user's decision. This rule has no alternatives.

– R5. **Exclusive decision pattern**. The text in the connection objects of the exclusive decision gateway is mapped onto two alternatives (Fig. 7a): (1) a radio button, when each alternative of the gateway represents an option, (2) a push button, when each alternative of the gateway represents an action.

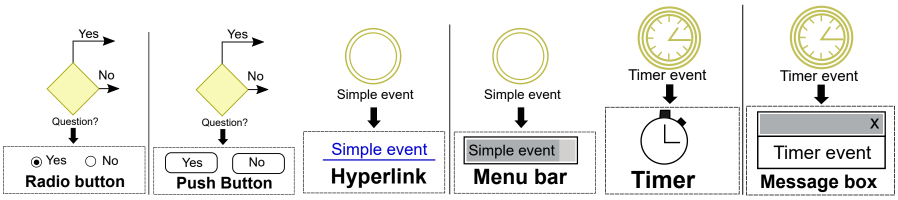

Fig. 7 The two alternatives of R5 (a), R7 (b), and R8 (c).

- R6. **Synchronization pattern**. User tasks with a dependency among them after the synchronization gateway in the same swimlane are mapped onto a GUI providing the end user with some guidance to carry out these tasks, with three alternatives similar to R2: (1) a wizard, (2) a tabbed dialog box, (3) a group box.
- R7. **Implicit Decision Pattern**. A simple type event continuing after an event based gateway is mapped onto two alternatives (Fig. 7b): (1) a hyperlink with the name of the event, navigating to a form, (2) a menu bar with the name of the event, navigating to a form. A target form is created with the attributes associated with the event according to R1.
- R8. **Timer**. A timer type event using a time variable continuing after a gateway is mapped onto two alternatives (Fig. 7c): (1) a timer generating a programmed action after a time interval, (2) a message box showing a notice information.

3.3 Usability Recommendations

Mapping rules defined in Section 3.2 offer multiple alternatives, from which the analyst or the designer is free to decide. This decision is specified as a stereotype in the BPMN model [16]. Note that the definition of these stereotypes can be seen in the previous publication ([16]) and is out of scope of the current work. In order to provide some guidance to choose the best option, the various alternatives are classified and sorted according to usability guidelines from usability engineering: Galitz's usability guidelines [18], Johnson's GUI Bloopers [24], Microsoft MSDN [39], SAP Fiori [59], U.S. Web Design [44], Google Material Design [21] and van Welie [62]. Table 1 shows the alternatives for each rule and the circumstances under which each alternative is recommended to support the usability of the resulting GUI. We name them *usability recommendations*, since they are defined in terms of elements of the BPMN models and class diagrams. Usability recommendation related to R0 is related to the attributes of the class diagram, while the recommendations of the remaining rules are related to BPMN models.

Table 1 Usability recommendations.

Rule	Alternatives	Usability recommendations
R0, R1	Text box	The attribute can be a string or a short number in a single line [59]. The attribute can be a password, URL, phone number or email address [59]. The attribute can be any value that the system cannot reasonably predict, such as a user's response to a warning. There is a huge variability in users responses [44].
	Combo box	The attribute can be a long list of elements (minimum 13, maximum 200 entries) [59]. The values in the list of options are secondary information and do not need to be displayed immediately [59].
	List box	The attribute can take values of one or more options [39]. A list large enough to show three to eight items when opened [39]. To choice one or several options from a list [39].
	Check box	The attribute can take values of a group or a list of options that can be selected independently of each other [59]. The attributes can take values of a list of options displayed immediately without user interaction (also in read only cases) [59].
	Radio button	The attribute can take one value of a set of mutually exclusive options [44]. User needs to quickly choose among at least two clearly different options [59].
R2, R6	Wizard	The user needs a sequential navigation assistant [39]. Each task has more than eight attributes of the class diagram [62]. There are three or more steps with dependency in the BPMN model [62].
	Tabbed dialog box	The user must show the information on pages classified separately [21].
	Group box	The user needs to group widgets within a form [39]. The task has less than eight attributes of the class diagram [39].
R3	Report	The task shows information ready for printing [18].
	Data grid	The user needs to display data with multiple properties in rows and columns [18]. An advantage of the data grid is that users can sort and filter the data [24]
R7	Hyperlink	The user needs a navigation between interfaces or another event [44].
R8	Timer	The user needs an event that requires a chronometer [39].
	Message box	The user needs to be interrupted during the execution of an action [59]. The system needs to display error messages, warning messages, success messages, confirmation messages or information messages [59]. The user needs to take a decision [59].

4 Experiment

This section describes the experiment conducted to empirically validate the set of rules for mapping BPMN models to GUIs and the usability recommendations formulated for their alternatives introduced in Section 3. This section is structured as follows: first, the research questions are defined along with their hypotheses; then, the experimental design is detailed; finally, results are reported and discussed.

4.1 Research Questions and Hypotheses

Let us define our *reference set* as the set of the nine mapping rules R0 to R8 along with their usability recommendations. To investigate the empirical validation of this reference set, there is a need to check whether it maps the BPMN models onto the same GUIs as developers would obtain without knowing our reference set. Moreover, we need to assess whether our rules and usability recommendations could reach a consensus with rules used by participants. Thus, the experiment is conducted both from a research perspective interested in identifying a valid set of rules for mapping BPMN models onto GUIs and from a practitioners' perspective interested in receiving relevant and valid information for designing GUIs corresponding to their business processes without requiring any prior experience in usability engineering. The following research questions are formulated:

- **RQ1**: What is the *correctness of the mapping rules* regarding how the participants design GUIs for the same BPMN models without the reference set? The correctness is defined as the degree to which a system or component is free from faults in its specification, design, and implementation, according to IEEE [23]. To address this property, we want to test the hypothesis H_1 = *Mapping rules used by participants are similar to rules of the reference set.*
- **RQ2**: What is the *completeness of the mapping rules* regarding how the participants design GUIs for the same BPMN model without the reference set? The completeness is defined as the degree to which a model specifies all the relevant statements of a domain [65, 35]. To address this property, we want to test the hypothesis H_2 = *Mapping rules of the reference set are all used by participants.*
- **RQ3**: What is the *perceived usefulness of the rules* by the participants for mapping BPMN to GUIs? The perceived usefulness is defined as a person's participantive probability that using a particular system would enhance her or his job performance [15]. To address this property, we want to test the hypothesis H_3 = *The use of any rule of the reference set is perceived as useful by the participants.*
- **RQ4**: What is the *intention to use the rules* by the participants for mapping BPMN to GUIs? The intention to use is defined as the extent to which a person intends to use a particular system [15]. To address this property, we want to test the hypothesis H_4 = *There is intention to use any rule from the reference set by the participants.*
- **RQ5**: What is the *correctness of the usability recommendations* regarding participants' preferences? To address this property, we want to test the hypothesis H_5 = *The participants preferences for alternatives are the same as the usability recommendations of the reference set.*
- **RQ6**: What is the *perceived usefulness of the usability recommendations* by the participants for deciding among alternatives of the same mapping rule? To address this property, we want to test the hypothesis H_6 = *The use of any usability recommendation of the reference set is perceived as useful by the participants.*
- **RQ7**: What is the *intention to use the usability recommendations* by the participants? To address this property, we want to test the hypothesis H_7 = *There is*

intention to use any usability recommendation from the reference set by the participants.

4.2 Method

Our study was a within-participants design with seven response variables.

4.2.1 Response Variables and their Qualitative and Quantitative Measures

This experiment uses seven response variables: four variables to assess the validity of the mapping rules (RQ1-RQ4) and three variables to assess the validity of the usability recommendations (RQ5-RQ7). The correctness of the rules (RQ1) is measured as the percentage of mapping rules that a participant would use from our reference set without knowing it (Eq. 1). For example, if a participant used the mapping rules `R1`, `R2`, and `R3` while our proposal used `R2` and `R4`, the rules correctness for that participant is: $1/3 \times 100\% = 33\%$. A ratio closer to 100% would mean that the participant applied almost the same rules as the ones proposed from our reference set.

$$\text{Rules correctness} = \frac{\text{Number of rules used by a participant from the reference set}}{\text{Overall number of rules used by a participant}} \times 100\% \quad (1)$$

The completeness of the rules (RQ2) is measured as the percentage of all rules that are used by the participant from the reference set without knowing it (Eq. 2). For example, if a participant used the rules `R1`, `R2`, and `R3` while our solution used `R2` and `R4`, the rules completeness for that participant is: $1/2 \times 100\% = 50\%$. A ratio closer to 100% would mean that the rules used from our reference set are almost the same as the ones used by the participant.

$$\text{Rules completeness} = \frac{\text{Number of rules used by a participant from the reference set}}{\text{Overall number of rules of the reference set}} \times 100\% \quad (2)$$

The perceived usefulness of the rules (RQ3) is measured as the sum of numerical values assigned to the eight statements of Moody's framework [41] (based on Lindland's work [35]), a widely used and validated framework. Each statement is captured by a 5-point Likert scale [33] as follows (see Appendix 1): 1=Totally disagree, 2=Fairly disagree, 3=Neutral, 4=Fairly agree and 5=Totally agree. The result of the addition is then classified into a rank with five possible options: 1–8=Totally disagree, 9–16=Fairly disagree, 17–24=Neutral, 25–32=Fairly agree, and 33–40=Totally agree. Since participants were not aware of our reference set, this measure is the perceived usefulness that a possible existence of these rules will emerge in the participant's knowledge. We are not measuring the perceived usefulness of the reference set, but the perceived usefulness of the general concept of using such mapping rules. For example, a participant answering 6 questions with 5=Totally agree and 2 questions with 3=Neutral obtains a total of $(6 \times 5) + (2 \times 3) = 36$ (Totally agree).

The intention to use of the rules (RQ4) is measured as the sum of numerical values assigned to the two statements of Moody's framework [41] to capture intention to use. Similarly to the perceived usefulness, each statement is captured by a 5-point Likert scale [33] as follows (see Appendix 2): 1=Totally disagree, 2=Fairly disagree, 3=Neutral, 4=Fairly agree and 5=Totally agree, but the final rank differs:1–2=Totally disagree, 3–4=Fairly disagree, 5–6=Neutral, 7–8=Fairly agree, and 9–10=Totally agree. The intention to use measure is generic for any mapping rule, not just for those from our reference set, since participants were not aware of our rules.

The correctness of the usability recommendations (RQ5) is measured as the percentage of the participant's preferences for alternatives that agree with our usability recommendations (Eq. 3). For this purpose, a questionnaire of fourteen questions is created (see Appendix 3), each question deals with one usability recommendation for one mapping rule at a time. Each question has four possible responses, among which one and only one matches our usability recommendation. All responses are graphically depicted as GUI widgets, from which the participant must choose the alternative that she/he thinks that supports the usability as much as possible. Next, these preferences are compared with our usability recommendations from the reference set. A ratio closer to 100% would mean that participants' preferences of rules alternative are the same as our usability recommendations.

$$\text{Usability recommendations correctness} = \frac{\text{Number of participants who chose our recommendation}}{\text{Overall number of participants}} \times 100\% \qquad (3)$$

The perceived usefulness of usability recommendations (RQ6) is measured as the sum of numerical values assigned to the eight statements of Moody's framework [41] to capture perceived usefulness (see Appendix 4), similarly to RQ3. Since participants were not aware of our usability recommendations, the perceived usefulness was measured regarding the general idea of using usability recommendations, not particularly those belonging to our reference set.

The intention to use of the usability recommendations (RQ7) is measured as the sum of numerical values assigned to the two statements of Moody's framework [41] to capture intention to use (see Appendix 5). The measure is the same as computed in RQ4 for intention to use of the rules. The questionnaire focuses on the intention to use of any usability recommendation as in RQ6, not just our proposed usability recommendations.

4.2.2 Stimuli

The experiment was conducted with stimuli as two experimental design problems of a similar complexity divided into several steps. This division is required to compute the measures of correctness (RQ1) and completeness (RQ2). These two problems were selected for the following reasons: no particular BPMN pre-requisites, no semantic knowledge of the domain of discourse, familiarity exposure is comparable, and no prior knowledge of the design problem. Both experimental problems include a BPMN model, along with its textual description, and its corresponding UML 2.5 Class Diagram to describe the business processes and the data persistence respectively, thus mitigating the threat of low prior BPMN knowledge. The use of two problems is due to yield results independently of a specific problem, which mitigates the generalization threat.

First Problem: Hiring and Staff Integration. This problem is aimed at designing a GUI for registering new employees in a company. The process starts when (step 1) the human resources verifies if the new employee comes from selection and recruitment, the new employee's data is not recorded. If the new employee does not come from selection and recruitment, the personal resources area saves information (step 2) of the new employee in the system with: `IdEmployee`, `First name`, `Last name`, `Phone`, `Mobile phone`, `Email`, `Gender`, `NIE-International Number`, `Salary`, `Contract starting date`, `Contract ending date`, and `Department`. Next (step 3), the human resources saves the documents and contract in the system with `Family's book`, `DNI`, `Criminal record`, and `Curriculum Vitae`. Next (step 4), the manager saves material from the new employee (`Work table`, `Computer`, `Phone`, `Mobile phone`, and `Identification card`), and (step 5), the administrator of Technology area creates the user and authorizes material for the employee. Fig. 8 reproduces the BPMN model used for this problem, while Fig. 9 shows the Class Diagram.

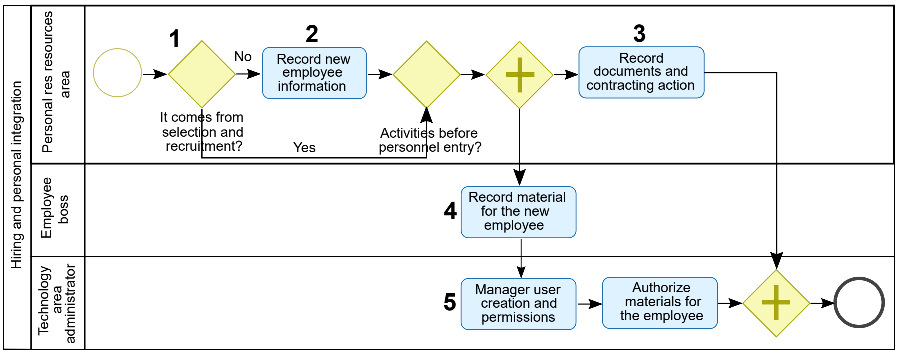

Fig. 8 BPMN model for Problem 1.

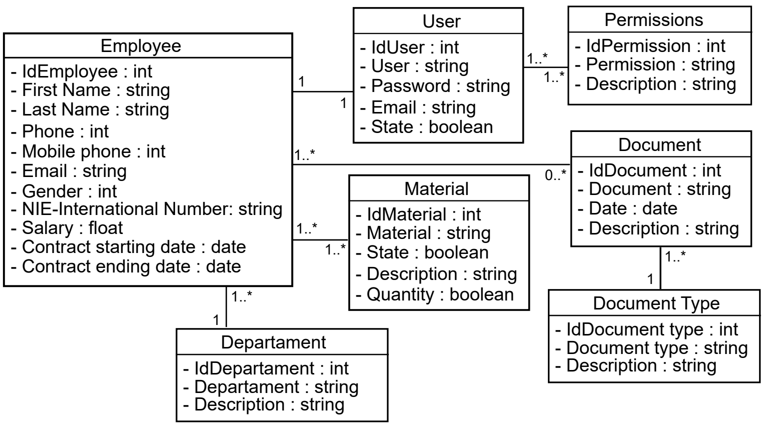

Fig. 9 Class diagram for Problem 1.

Second Problem: Credit Request. This problem is aimed at designing a GUI for registering a credit request issued by a customer. The process starts when (step 1) the manager bank saves a credit request (`IdCredit`, `Credit date`, `State`, `Manager bank`, `Customer`, `Disbursement date`, `Amount`, and `Credit type`). Next (step 2), the manager bank verifies the information of the requested credit. If the result of verification is a rejection, the process finishes; if the result of the verification is an approval, two events could happen: (1) (step 3) The manager receives the documentation, selects the `IdCredit` from a list and provides the following documents: `Payroll`, `NIE copy`, water or electricity bill and other documents. Next (step 4), the bank manager analyzes the received documents. If the credit request is approved (step 5), the manager bank saves the disbursement of the credit; if the credit request is rejected (step 6), the customer is informed of the credit rejection. (2) (step 7) An internal timer (not visible) is activated to start the next step in two days. The manager does a customer follow-up of the presented documents (Step 8). Next (step 9), if the customer presents documents, she/he goes to the event (1); if the customer does not present documents, the process finishes. Fig. 10 reproduces the BPMN model used for this problem, while Fig. 11 shows the Class Diagram.

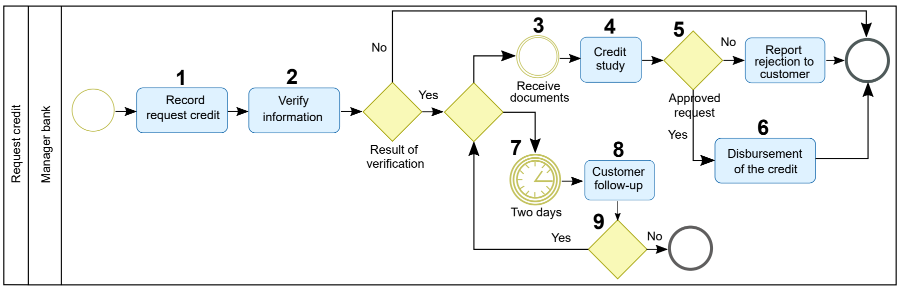

Fig. 10 BPMN model for Problem 2.

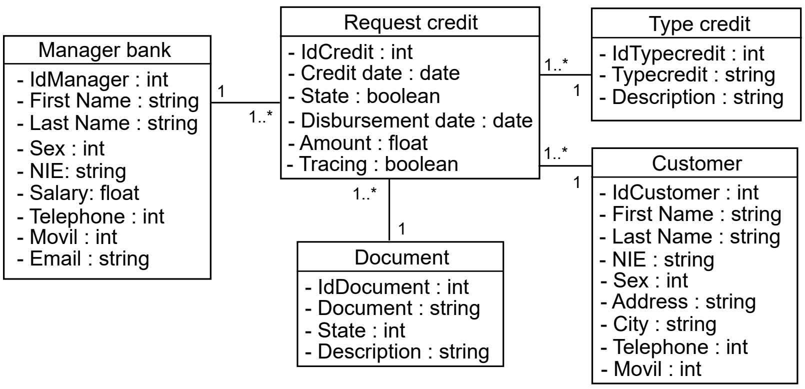

Fig. 11 Class diagram for Problem 2.

4.2.3 Procedure

The procedure for the experiment was structured in one session of two hours:

1. **Introduction to BPMN**. Before attending to the experiment, participants had to read a description of all the BPMN primitives as homework. The description consisted in a document with the basics of BPMN, both syntaxes and semantics. This document was given to the participants two weeks before the experiment. One experimenter taught an introduction to BPMN in 15 minutes before starting the experiment.
2. **Filling a training test**. After the training and the brief introduction to BPMN by the experimenter, participants had to pass a test with questions related to BPMN. The test consisted of 10 questions about BPMN, each question had 4 alternatives with only one possible correct answer. Each correct answer was computed as one point, so possible points were between 0 (no correct answers) and 10 (all answers are correct). We considered that participants that got more than 5 points were capable of participating in the experiment. In case we had participants with a score between zero and five, these were removed from the experiment. Table 2 shows the marks of the 43 participants that were evaluated. All participants passed the test.

Table 2 Results of the entrance test.

Marks	Numbers of participants
6	0
7	5
8	4
9	8
10	26
Total	43

3. **Filling the demographic questionnaire**. Each participant performed the task in a controlled environment. Prior to the task, each participant was welcomed, had the process explained to them, signed a consent form, and filled in a questionnaire to identify their background.
4. **Solving the experimental problems**. The participants had to solve both experimental problems by designing and drawing GUIs taking as input BPMN models and UML Class Diagrams. No particular guidance was provided nor any constraint was imposed. Participants were instructed to behave naturally and to draw GUIs as they preferred. The process of drawing interfaces lasted sixty minutes in total, thirty minutes for each experimental problem.
5. **Filling the Post-test questionnaires**. After the task was completed, participants filled in on-line the questionnaire of perceived usefulness of rules and the questionnaire of intention to use of such rules during ten minutes. Then, participants filled in on-line successively the questionnaire of correctness of the usability rec-

ommendations (twenty minutes), the questionnaire of perceived usefulness of usability recommendations and the questionnaire of intention to use (ten minutes).

One experimenter then analyzed the GUIs drawn of both experimental problems to calculate the correctness and completeness of our proposed rules, the perceived usefulness and intention to use of rules, the correctness of usability recommendations, the perceived usefulness and intention to use them (Subsection 4.2.1).

4.2.4 Participants

Whitefield et al. [63] defined a framework for evaluating user interfaces that is structured according to two dimensions: user and user interface. Each dimension could be a real one or a representation of it. Therefore, the end user could be either represented or real. And so does the user interface. For instance, an analytical user model could replace real users to simulate their behavior; the final user interface, when it is not yet existing or running, could be substituted by a prototype or an interactive mockup. In this way, it is not needed to wait for real users to evaluate a real interface. The recruitment of non-professional users has been used in other works in the HCI field, such as [54], which validates visualization techniques of HCI with subjects with and without knowledge. Other examples are the work [5], which validates ecological interfaces with non-expert subjects, and the work [53], which validates usability through computer science students.

In this experiment, we target *represented end users* evaluating a *represented graphical user interface* for the following reasons: having real designers would introduce new variables to be controlled such as level of expertise, number of years of design experience; real designers are expensive to involve; having real GUIs would require participants to design their results within a visual editor that would inevitably be tied to a particular look and feel imposed by an operating system, such as MS-Windows or MacOS. We preferred to let participants to concentrate on high-level questions raised by the two problems instead of asking them to finely design GUIs with low-level presentation details.

Consequently, a *random sampling* (each member has an equal chance to be selected) was applied on a population of undergraduate students in computer science, who are considered as represented end users. 43 participants (41 male vs. 2 female, 29 aged in the range 15-20, 9 in 21-25, 3 in 26-30, 1 in 31-35, and 1 above 35, M=26.20, SD=6.71) were recruited from the whole undergraduate computer science program of University of Valencia (Spain), with a low BPMN knowledge but with a reasonable knowledge of GUI design (see Table 3). All of them participated voluntarily in the experiment and had the same treatment: they were instructed to naturally design and draw GUIs by paper and pencil from BPMN models and Class Diagrams.

5 Results and Discussion

After all participants completed the task, the questionnaires and the computed measures were entered into a Microsoft Excel 2016 spreadsheet in an anonymous format so the participants could not be identified and data were analyzed using SPSS V20.

Table 3 Knowledge of BPMN models, UML Class Diagram and GUIs.

Knowledge of	None	Low	Medium	High
BPMN models	23	15	4	1
UML Class Diagram	10	17	15	1
Graphical User Interfaces	3	18	18	4

5.1 Correctness of the Mapping Rules

Table 4 shows the percentage of correctness of the five steps of Problem 1 as well as the nine steps of Problem 2 computed according to Eq. 1. Steps 1-4 of Problem 1 and Steps 1, 5, 6, 8, 9 of Problem 2 yield a ratio of 100% of correctness. This suggests that the participants apply the same rules as proposed in the reference set in these cases. The most frequent rules are `R0`, `R1`, `R4`, and `R5`, since they are widely used by the participants. Step 5 in Problem 1 shows a value of 67%, thus meaning that the participants agree on two rules for three proposed ones. Steps 2, 3, 4, and 7 of Problem 2 have no correctness, thus meaning that participants in these cases use mapping rules that are different from our proposed ones. These less frequent rules are `R3`, `R6`, `R7`, and `R8`. While a report or data grid is mapped by `R3`, participants preferred a message box. While a wizard or a group box or a tabbed dialog box are mapped by `R6`, participants preferred a simple form, which is the simplest and most common GUI container in contrast to those more advanced widgets, which are specific to complex tasks. Similarly, while a hyperlink or a menu bar is mapped by `R7`, participants again preferred a form. `R8` maps a timer or a message box, but participants preferred not using any GUI for representing the time. The average percentage for Problem 1 ($M =$ 93.40%) is significantly higher than for Problem 2 (M=55.55%), but still giving a global average of 74.47%, thus supporting H_1.

Table 4 Percentage of Correctness of the Mapping Rules.

Problem 1	% Correctness	Problem 2	% Correctness
Step 1	100%	Step 1	100%
Step 2	100%	Step 2	0%
Step 3	100%	Step 3	0%
Step 4	100%	Step 4	0%
Step 5	67%	Step 5	100%
		Step 6	100%
		Step 7	0%
		Step 8	100%
		Step 9	100%
Average	93.40%	Average	55.55%
		Global average	**74.47%**

Table 5 reports the frequency of mapping rules selected by participants for each step in Problem 1 and our proposed rules from the reference set. OR expresses another rule than from our reference set. `R0` and `R1` are the most frequent in most steps (2, 3, and 4) as they manipulate basic widgets commonly used such as form, text box, combo box, and radio button, which all agree with our proposed rules. In Step 1, most participants used R4 (label) and R5 (radio button), which also concur with our proposal. Only four participants used different rules. In Step 5, R6 (wizard or group box or tab control) and R1 (form) share the same number of participants that use them.

Table 5 Frequency of rules used by participants in Problem 1.

Steps	Rules used by participants (total)	Our proposal rules
Step 1	R4 (36) - R5 (39) - OR (4)	R4 - R5
Step 2	R0 (43) - R1 (43)	R0 - R1
Step 3	R0 (42) - R1 (42) - OR (1)	R0 - R1
Step 4	R0 (43) - R1 (43)	R0 - R1
Step 5	R0 (40) - R6 (20) - R1 (20) - OR (3)	R0 - R6

Table 6 reports the frequency of mapping rules selected by participant for the nine steps of Problem 2 and our proposed rules from the reference set. OR expresses another rule and NWD means that no widget is drawn (the participant draw nothing). In Steps 1, 6 and 8, all participants used R0 and R1, which concur with our proposal. In Steps 2 and 4, most participants used different rules since these involve very specific widgets such as a form, a message box and a label. In Step 7, most participants do not draw any widget, considering merely implementation code for validating the condition of the event timer. A high number of participants preferred R8 (timer or message box), as we proposed. In Steps 2–4, participants used another rule OR, because they tend to rely on other widgets. For example in Step 2, most preferred a message box, in Step 3, a form, and in Step 4, again a message box. All these decisions were participative and suggest that a few rules were not applied in the same way as our proposal because participants feel other options more intuitive.

Table 6 Frequency of rules used by participants in Problem 2.

Steps	Rules used by participants (amount)	Our proposal rules
Step 1	R1(43) - R0(43)	R1 - R0 - R2
Step 2	R1(1) - R3(4) - R4(1) - OR(35) - NWD(2)	R3
Step 3	OR(43)	R7
Step 4	R0(2) - R1(2) - R3(4) - R4(1) - R5(1) - OR(35) - NWD(1)	R3
Step 5	R4(37) - R5(37) - OR(2) - NWD(4)	R4 - R5
Step 6	R0(43) - R1(43)	R1 - R0
Step 7	R8(20) - NWD(21) - OR(2)	R8
Step 8	R1(43) - R0(43)	R1 - R0
Step 9	R4(35) - R5(35) - OR(1) - NWD(7)	R4 - R5

5.2 Completeness of the Mapping Rules

Table 7 shows the percentage of completeness of the five steps of Problem 1 as well as the nine steps of Problem 2 computed according to Eq. 2. All steps of Problem 1 and Steps 5, 6, 8, 9 of Problem 2 yield a ratio of 100%. This suggests that the mapping rules of the reference set were all applied by the participants. The most frequent rules are R0, R1, R4, R5, and R6, since these rules belong to frequently used widgets. However, Step 1 in Problem 2 shows a value of 67%, which means that from three proposed rules, only two agree with the participants' preferences. Steps 2, 3, 4, 7 of Problem 2 are incomplete, thus meaning that they are not considered appropriate by participants in the context of the step as they opted for other rules. The average percentage for Problem 1 (M=100%) is significantly higher than for Problem 2 (M=51.88%), but still giving a global average of 75.94%, thus supporting H_2.

Table 7 Percentage of Completeness of the Mapping Rules.

Problem 1	% Completeness	Problem 2	% Completeness
Step 1	100%	Step 1	67%
Step 2	100%	Step 2	0%
Step 3	100%	Step 3	0%
Step 4	100%	Step 4	0%
Step 5	100%	Step 5	100%
		Step 6	100%
		Step 7	0%
		Step 8	100%
		Step 9	100%
Average	100.00 %	Average	51.88%
		Global average	**75.94%**

5.3 Perceived Usefulness of the Mapping Rules

Fig. 12a shows a divergent stacked bar of the answers provided by the participants to the questionnaire for this property: roughly 28% totally agreed and 67% fairly agreed (more than 90% in total), thus suggesting that most participants estimated that the existence of rules could be useful. A minority of participants are indecisive and do not perceive the existence of rules as useful, thus supporting H_3.

After applying a Bonferroni Type I correction, a Kruskal-Wallis H test reveals that there is a significant difference between the answers provided by participants to these eight statements of this questionnaire (*H-stat*=14.36, *H'-ties*=16.53, df=7, $p^* = 0.02$). Since this Kruskal-Wallis test shows a significant difference between the eight questions, then pairwise comparisons or contrasts can be used to pinpoint the differences following a single factor analysis of variance. After performing pairwise Mann-Whitney tests, we only found five statistically significant differences between the answers provided by participants to this questionnaire (no significance= $p > .05$, *= $p \leq .05$, **= $p \leq .01$, ***= $p \leq .001$):

1. There is a significant difference in the answers for Question 3 (M=3.69, SD=.84, SEM=.15) and for Question 7 (M=4.14, SD=.81, SEM=.12); df=1, M=.45, p^*=.010, thus suggesting that participants estimate usefulness (Q7) even more important than the improvement brought by the rules (Q3)

2. There is a significant difference in the answers for Question 4 (M=3.67, SD=.93, SEM=.14) and for Question 7; df=1, M=.48, p^*=.013, thus suggesting that usefulness (Q7) is more important than scalability (Q4).
3. There is a highly significant difference in the answers for Question 5 (M=3.57, SD=.89, SEM=.13) and for Question 6 (M=4.02, SD=0.75, SEM=.13); df=1, M=.45, p^*=.017, thus suggesting that understandility (Q6) is more important than effectiveness (Q5).
4. There is a highly significant difference in the answers for Question 5 and for Question 7; df=1, M=.57, p^{**} = .0018, thus suggesting that usefulness is more important than effectiveness.
5. There is a highly significant difference in the answers for Question 7 and for Question 8 (M=3.71, SD=.74, SEM=.12); df=1, M=.42, p^{**}=.0078, thus suggesting that usefulness (the best rated question in the end) is more important than its practicality (Q8).

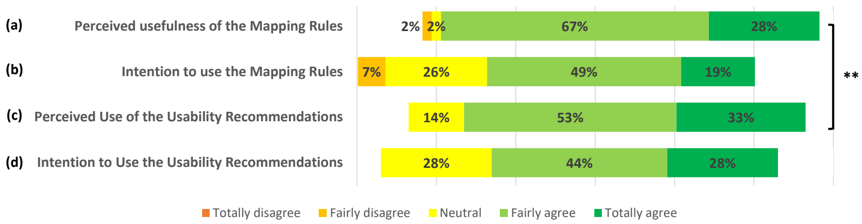

Fig. 12 Distribution of answers to questionnaires: perceived usefulness of rules (a), intention to use of rules (b), perceived usefulness of the usability recommendations (c), intention to use of the usability recommendations (d).

5.4 Intention to Use the Mapping Rules

Fig. 12b shows a divergent stacked bar of the answers provided by the participants to the two related questions: roughly 19% totally agreed and 49% fairly agreed (more than 68% in total), thus suggesting that most participants still have the intention to use these rules in the future. A minority of participants (7%) is indecisive and do not plan to use these rules again, thus supporting H_4.

5.5 Correctness of the Usability Recommendations

Fig. 13 shows the diagram of frequency of the percentage of correctness of the usability recommendations. Widgets related to questions 2, 4, 5, 7, 10, 11, 12 and 13 receive the best values for correctness (most with a percentage higher than 50%), while related to questions 1, 3, 6, 8, 9 and 14 receive the worst values, thus suggesting that most participants tend to prefer alternative widgets than those recommended in our usability recommendations. The preferred widgets are respectively:

1. Questions 1 and 3: a tabbed dialog box, whereas a wizard is recommended.
2. Question 6: a combo box, whereas a list box is recommended.
3. Question 8: a group box, whereas a tabbed dialog box is recommended.
4. Question 9: a wizard, whereas a group box is recommended.
5. Question 14: a push button instead of a radio button.

Convergence with the usability recommendations is obtained for eight questions, divergence for six questions, thus suggesting that H_5 is partially supported.

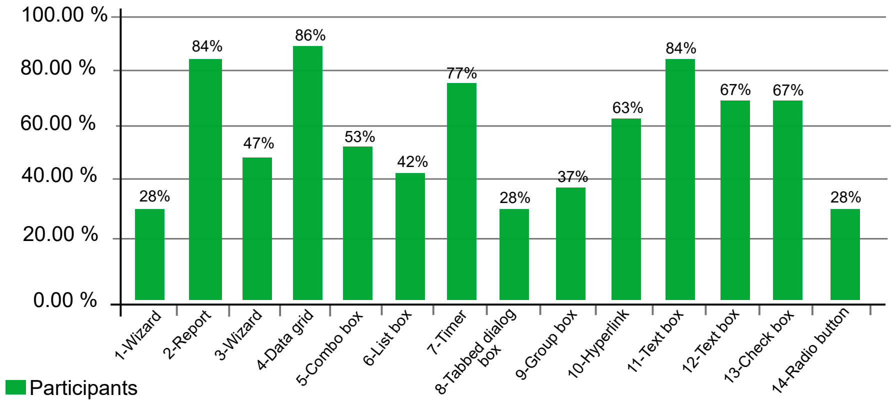

Fig. 13 Percentage of correctness of the usability recommendations.

5.6 Perceived Usefulness of Usability Recommendations

Fig. 12c shows a divergent stacked bar of the answers provided by the participants to the questions related to perceived usefulness of the usability recommendations: roughly 33% totally agreed and 53% fairly agreed (86% in total), thus suggesting that most participants still have the intention to use these usability recommendations in the future. A minority of participants (14%) is indecisive and do not plan to use these recommendations again, thus supporting H_6. Note that no participant disagree with the statements of this property, as opposed to the mapping rules. Among all possible combinations between the properties, there is only one correlation between this property and perceived usefulness of the mapping rules, though (Fig. 12), through a highly significant pairwise Mann-Whitney test ($M = .28$, $p^{**} = .004$). All other tests were not significant.

5.7 Intention to Use of Usability Recommendations

Fig. 12d shows a divergent stacked bar of the answers provided by the participants to the questions related to the intention to use the usability recommendations: roughly

28% totally agreed and 44% fairly agreed (72% in total), thus suggesting that most participants are still intended to use these usability recommendations in the future. On third of the participants (28%) is indecisive and does not plan to use these recommendations again, thus partially supporting H_7.

5.8 Discussion

This section reviews all the results of the experiment. For each response variable, we discuss the interpretation of results.

The result related with **correctness of the mapping rules** shows that most participants used the same rules that we have proposed. The proposed rules that yield the best values (Rules R0, R1, R4, R5 and R6) are because these rules are the most straightforward. For R0 and R1, most participants think that user type tasks must be transformed into forms; whereas the widgets text box, list box, combo box, radio button, check box are extracted from the Class Diagram. This suggests that R0 and R1 are applied unconsciously by the participants as we propose. The results for R4 and R5 show that most participants think that a gateway of exclusive decision must be transformed into a label that displays a question of two alternatives: Yes/No in radio button or push button, which agrees with our proposal. For R1 and R6, most participants transform tasks of synchronization gateway in the same lane into a wizard, a group box, a tabbed dialog box, and form, which also agrees with our proposal. Both rules could be used indistinctly according to participants preferences. Our proposed solution opted for R6 instead of R1 because, at first glance, a wizard, a tabbed dialog box or a group box (R6) is more convenient than a single form to represent a synchronization pattern (R1). The proposed rules that yield the worst values (Rules R3, R7, and R8) are because they are ambiguous and lead to a free interpretation for a wide GUI variety. For R3, most participants transform a service type task into a message box, while we propose a report or a data grid. Maybe this choice assumes that a message box is more frequent in any information system rather than reports or data grids. For R7, most participants think that when an event type is simple, a unique form that contains all the information is the best option. The navigation from one to another can be done though a hyperlink or a menu bar. This discrepancy could be due to the fact that using a single form is easier to design for the designer, even though end users will have too much information in the same interface. For R8, most participants think that when an event type is timer, they do not use any widget on the interface. Note that the use of a timer is not so frequent in existing interfaces as the use of a form, a group box, or a radio button among others, so drawing a timer for the participants is not so intuitive.

The result related with **completeness of the mapping rules** shows that most of the proposed rules were used by participants unconsciously. The proposed rules that were widely used by the participants in the experimental problems are R0, R1 R4, R5, R6. Rules R4 and R5 are used in exclusive decision patterns and R6 is used in the synchronization pattern as we propose. These rules are frequently used in common systems, so participants knew them in advance. R0 is used to draw widgets in forms from the class diagram and R1 is used in user type task to generate a form. The proposed rules that were not used by participants are: R2, R3, R7 and R8. R2 and R3 belong to a sequence pattern but participants preferred using other widgets. For example, R2 propose using a wizard, group box or tabbed dialog box, but participants preferred using only forms. Other example is in R3, where a report or a data grid was initiated, but participants preferred using a message box, maybe because they see this widget more frequently in existing GUIs. R7 and R8 belong to an advanced pattern (event-based pattern) that were not very intuitive for the participants. In both rules, most of the participants preferred to draw widgets different from our reference set. For example, navigations in R7 (hyperlink or a menu bar) were not chosen by the participants, who opted for a unique form that gathers all the information without any navigation.

Note that results obtained for correctness and completeness of the rules are very similar. This is because we used a few rules in our experiment (nine rules), so the difference between number of rules that match with our proposed rules (correctness) and number of rules that match with rules used by participants (completeness) is not so large.

The results related with **perceived usefulness of the rules** show that most participants accepted the idea of using rules to generate user interfaces, even though they still do not know them. Results yield that most participants opted for ”Fairly agree” or ”Totally agree”, which means that they think that the use of generation rules would facilitate to map GUIs from BPMN models. This reinforces the idea of automating the rules application as much as possible to generate interfaces from BPMN models through Model-to-Code (M2C) generation.

The results related with **intention to use of the rules** show that most participants are interested in using rules to generate interfaces. Most participants marked the option ”Fairly agree” or ”Totally agree” since they have the intention to use them. There is a small number of participants that are indecisive in the intention to use the rules. Maybe, if they had known the rules more precisely, they would have had a clearer answer (positive or negative). Note that the idea of automating the rules application could also change the intention to use to a more positive level, even though this automation is out of scope of the current paper. This result suggests the development of a tool to apply the mapping rules in an assisted way to automate the process as much as possible.

The result related with **correctness of the usability recommendations** shows that most participants have preferred our usability recommendations. The questions with the highest value exceed the 50% of correctness (Questions 2-report, 4-data grid, 5-list box, 7-timer, 10-hyperlink, 11 -text box, 12-text box, 13-check box) were the most intuitive and the clearest for the participants. Questions 2, 4 and 11 exceed 80% of correctness, thus suggesting that most participants agree with our recommendations because they consider that these widgets are the most suitable. Questions 5, 7, 10, 12 and 13 were between 50% and 80% of correctness, this means that a high number of participants agree with our recommendations, but there is a smaller number of them that preferred other widgets. For example, in Question 5, 47% of the participants prefered a list box (not our recommendation) and in Question 10, 37% of participants preferred a combo box instead of our recommendation (hyperlink). The questions with the worst values are below 50% of correctness (Questions 1-wizard, 3-wizard, 6-list box, 8-tabbed dialog box, 9-group box, 14-radio button), these result may be due to the understandability of questions. In some cases. participants considered other widgets different from our proposal to get usable interfaces. For example, in Question 1 most participants preferred using tabbed dialog box, in Question 3-tabbed dialog box, in Question 6-combo box, in Question 8-group box, in Question 9-wizard and in Question 14-push button, which are different widgets of our proposal to support the usability.

The result related with **perceived usefulness of the usability recommendations** shows that participants consider the idea of working with usability recommendations as useful. Even though participants did not know the usability recommendations explicitly, most of them marked the option ”Fairly agree” or ”Totally agree” because these recommendations could help them to improve the interface usability without being an expert at usability. This results means that participants would use these usability recommendations if they were clear, understandable and unambiguous.

The result related with **intention to use of the usability recommendation** shows that most participants have the intention to use the usability recommendations even though they do not know them explicitly. Most participants marked ”Fairly agree’” or ”Totally agree”, and a small part of the participants were indecisive. This result leads to think that we must facilitate the use of these recommendations in a simple and understandable way, like a tool.

6 Threads to Validity

This section discusses the threats to validity that could affect the mapping rules and the usability recommendations. We have focused on the threats that could affect the design and the conduction of the experiment. In the following, we describe the threats according to Wohlin’s classification [64]. For each group of threats, we made a distinction among 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 types:

- *Conclusion validity*: this type of threat deals with the ability to draw the correct conclusion about relations between the treatment and the outcome. The experiment may suffer the following threats of this type: *Fishing*, which means that the experimenters are looking for a specific result. In order to minimize this threat, experimenters did not give guides to draw the interfaces. Another threat that appears is *Reliability of measures*, which means that the measures used in the statistical analysis might include mistakes. In order to minimize this threat, all the measures are applied by one experimenter (not by the participants). Another threat that appears is *Reliability of treatment implementation*, which means that the treatment might be operationalized wrongly. In order to minimize this threat, a short document that describes BPMN was given to the participants one week before the experiment to solve the lack of previous knowledge of BPMN. Moreover, the BPMN model was transcribed in text so that participants without knowledge in the BPMN could understand its meaning. Another threat that appears is *Random heterogeneity of participants*, which means that when there is heterogeneity in a study group, there is a risk that the variation due to individual differences is larger than due to the treatment. In order to minimize this threat, we recruited participants with similar profiles(students are undergraduate subjects who have previously taken Human-Computer Interaction (HCI) courses, with knowledge of interface programming). Moreover, we used demographic questionnaires to detect differences among the participants profiles.

- *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 participants*, which means that the experience of the participants is not enough to conduct the experiment. Our participants had two years of experience in coding GUI. In order to mitigate the lack of experience in BPMN, we trained them through a tutorial and they had to pass a test before participating. Note that participants only had to draw GUIs, they did not need a large experience in BPMN modeling. *History*, which means that differences may arise when treatments are applied at different times. In order to avoid this threat, we conducted the experiment in one session of two hours. 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, we checked the spreadsheets used to gather all the data. We selected two random participants to conduct a pilot test to ensure that the metrics are applied correctly through the spreadsheet. Another threat that appears is *Interactions with selection*, which means that when there are different groups of participants, different behaviors with the instruments might appear. In order to minimize this threat, we uniformly used the same instruments to all participants of the experiment. 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 participants with extra points in their marks. Another threat that appears is *Ecological validity* [26], which means that the context of use in which the experiment was conducted may affect the results. In order to minimize this threat, participants conducted the experiment in a controlled environment (i.e., a usability laboratory) and not in a real environment, such as a 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 might 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 participants that these experimental tasks are exercises as a part of the course, without mentioning the term ”experiment” or ”test”. Another threat that appears is *Hypothesis guessing*, which means that when people take part in an experiment they might try to figure out what the purpose and intended result of the experiment is. In order minimize this threat, we do not talk about research questions.
- *External validity*: this type of threat concerns with 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 participant population not representative of the population we want to generalize. In our experiment, we cannot ensure

that results can be generalized to participants with different profiles of our sample. Another threat that appears is *Interaction of settings 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 must highlight that the context of our experiment is just an academic environment.

7 Conclusion and Future Work

This paper summarizes a set of rules to generate interfaces from BPMN models complemented with Class Diagrams previously published. As a contribution of this work, for each rule, when there is more than one possible alternative to generate a widget, we have recommended the choice that optimizes the usability of the final user interface according to the literature. We have designed and conducted and empirical validation to check both approaches through two experimental problems and questionnaires online. This experiment is based on several measures: correctness of the mapping rules, completeness of the mapping rules, perceived usefulness of the mapping rules, intention to use the mapping rules, correctness of the usability recommendations, perceived usefulness of the usability recommendations and intention to use the usability recommendations.

The results of these measures are: most participants used unconsciously the same rules that we have proposed, most of the proposed rules were used by participants unconsciously, most participants accepted the idea of using a set rules to generate user interfaces, most participants have interested in using a set of rules to generate user interfaces, there are several widgets preferred by the participants beyond our recommendations, most participants agree with our usability recommendations, most participants have a stake in using usability recommendations as a guide when several design alternatives compete with each other.

The experiment suffers from the following limitations: the BPMN models have dependency with the class diagrams, not every BPMN model has a class diagrams to represent its persistence; the experimental problems were straightforward, the experiment was developed with participants who are undergraduate students with knowledge of interaction and usability, but limited experience of BPMN models.

The advantages are still present. Currently, there is little or no literature tackling with experiments on GUI mapping from BPMN models. Results show that most of the proposed rules are used by the participants who agree with our recommendations. The sample size is moderately large enough to generalize our results for participants with a similar background as the recruited in the experiment. This work is a step forward to automate the software development process.

As future work, we plan to replicate this experiment changing some elements of the design. First we are planning to use more complex problems. Second, the recruitment of participants with a higher knowledge in BPMN models. We plan to extend our approach with more rules and more usability recommendations such a way we can cover the generation of any BPMN pattern.

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 participants for conducting the experiment.

References

1. Acerbis, R., Bongio, A., Brambilla, M., Butti, S.: WebRatio 5: An Eclipse-Based CASE Tool for Engineering Web Applications. In: Proceedings of the 7th International Conference on Web Engineering, Como, Italy, July 16-20, 2007, *Lecture Notes in Computer Science*, vol. 4607, pp. 501–505. Springer, Berlin (2007). DOI 10.1007/978-3-540-73597-7_44
2. Barforooshi, S.Y., Moghadam, S.M., Nasiri, R.: Improvement of test management in bpmn with aris. In: 2010 2nd International Conference on Electronic Computer Technology, pp. 59–62. IEEE (2010)
3. Bizagi: Business Model Patterns. `http://resources.bizagi.com/docs/Workflow_Patterns_using_BizAgi_Process_Modeler_Esp.pdf`. Accessed: 2019-07-19
4. Bonitasoft: Bonita Software. `https://www.bonitasoft.com/downloads`. Accessed: 2019-07-19
5. Borst, C., Visser, R.M., Van Paassen, M.M., Mulder, M.: Exploring short-term training effects of ecological interfaces: A case study in air traffic control. IEEE Transactions on Human-Machine Systems **49**(6), 623–632 (2019)
6. Bouchelligua, W., Mahfoudhi, A., Mezhoudi, N., Daassi, O., Abed, M.: User interfaces modelling of workflow information systems. In: J. Barjis (ed.) Enterprise and Organizational Modeling and Simulation, pp. 143–163. Springer, Berlin, Heidelberg (2010)
7. Bouillon, L., Limbourg, Q., , Vanderdonckt, J., Michotte, B.: Reverse engineering of web pages based on derivations and transformations. In: Third Latin American Web Congress (LA-WEB'2005), pp. 11 pp.– (2005). DOI 10.1109/LAWEB.2005.29
8. Bouzidi, A., Haddar, N., Abdallah, M.B., Haddar, K.: Alignment of business processes and requirements through model integration. In: 2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA), pp. 1–8. IEEE (2018)
9. Brambilla, M., Butti, S., Fraternali, P.: WebRatio BPM: A Tool for Designing and Deploying Business Processes on the Web. In: B. Benatallah, F. Casati, G. Kappel, G. Rossi (eds.) Proceedings of International Conference on Web Engineering, ICWE '10, pp. 415–429. Springer, Berlin, Heidelberg (2010)
10. Brambilla, M., Fraternali, P., Vaca, C.: BPMN and Design Patterns for Engineering Social BPM Solutions. In: F. Daniel, K. Barkaoui, S. Dustdar (eds.) Business Process Management Workshops, pp. 219–230. Springer, Berlin, Heidelberg (2012)
11. Bush, A., Purao, S.: Mapping uml techniques to design activities. In: M. Rossi, K. Siau (eds.) Information Modeling in the New Millennium, chap. 12, pp. 199–217. IDEA Publishing Group (2011)
12. Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Bouillon, L., Vanderdonckt, J.: A unifying reference framework for multi-target user interfaces. Interacting with computers **15**(3), 289–308 (2003)
13. de Carvalho, E.A., Gomes, J.O., Jatobá, A., da Silva, M.F., de Carvalho, P.V.R.: Employing resilience engineering in eliciting software requirements for complex systems: experiments with the functional resonance analysis method (fram). Cognition, Technology & Work pp. 1–19 (2020)
14. Cruz, E.F., da Cruz, A.M.R.: Deriving integrated software design models from bpmn business process models. In: ICSOFT, pp. 605–616 (2018)
15. Davis, F.D., Bagozzi, R.P., Warshaw, P.R.: User acceptance of computer technology: A comparison of two theoretical models. Management Science **35**(8), 982–1003 (1989). DOI 10.1287/mnsc.35.8.982. URL `https://doi.org/10.1287/mnsc.35.8.982`
16. Diaz, E., Panach, J.I., Rueda, S., Pastor, O.: Towards a method to generate gui prototypes from bpmn. In: 2018 12th International Conference on Research Challenges in Information Science (RCIS), pp. 1–12. IEEE (2018)
17. Driss, M., Aljehani, A., Boulila, W., Ghandorh, H., Al-Sarem, M.: Servicing your requirements: An fca and rca-driven approach for semantic web services composition. IEEE Access **8**, 59326–59339 (2020)
18. Galitz, W.O.: The Essential Guide to User Interface Design: An Introduction to GUI Design Principles and Techniques, 2 edn. John Wiley & Sons, New York, USA (2002)

19. Giner, P., Torres, V., Pelechano, V.: Bridging the gap between BPMN and WS-BPEL. M2M transformations in practice. In: N. Koch, A. Vallecillo, G. Houben (eds.) Proceedings of the 3rd International Workshop on Model-Driven Web Engineering MDWE 2007, Como, Italy, July 17, 2007, *CEUR Workshop Proceedings*, vol. 261. CEUR-WS.org (2007). URL http://ceur-ws.org/Vol-261/paper06.pdf
20. Gonzalez-Huerta, J., Boubaker, A., Mili, H.: A business process re-engineering approach to transform bpmn models to software artifacts. In: E. Aïmeur, U. Ruhi, M. Weiss (eds.) E-Technologies: Embracing the Internet of Things, pp. 170–184. Springer International Publishing, Cham (2017)
21. Google, I.: Material Design. https://material.io/design/. Accessed: 2019-07-19
22. Han, L., Zhao, W., Yang, J.: An approach towards user interface derivation from business process model. In: J. Cao, X. Liu, K. Ren (eds.) Process-Aware Systems, pp. 19–28. Springer Singapore, Singapore (2016)
23. International Organization for Standards: ISO/IEC/IEEE 24765:2017 Systems and software engineering – Vocabulary. https://www.iso.org/standard/71952.html. Accessed: 2019-07-19
24. Johnson, J.: GUI Bloopers 2.0: Common User Interface Design Don'ts and Dos (Interactive Technologies) , 2 edn. Morgan Kaufmann (2007)
25. Juric, M.B., Mathew, B., Sarang, P.G.: Business process execution language for web services: an architect and developer's guide to orchestrating web services using BPEL4WS. Packt Publishing Ltd (2006)
26. Kieffer, S.: Ecoval: Ecological validity of cues and representative design in user experience evaluations. AIS Transactions on Human-Computer Interaction **9**(2), 149–172 (2017). URL https://aisel.aisnet.org/thci/vol9/iss2/4
27. Kolb, J., Hübner, P., Reichert, M.: Automatically generating and updating user interface components in process-aware information systems. In: R. Meersman, H. Panetto, T. Dillon, S. Rinderle-Ma, P. Dadam, X. Zhou, S. Pearson, A. Ferscha, S. Bergamaschi, I.F. Cruz (eds.) On the Move to Meaningful Internet Systems: OTM 2012, pp. 444–454. Springer, Berlin, Heidelberg (2012)
28. Kolb, J., Hübner, P., Reichert, M.: Model-driven user interface generation and adaptation in process-aware information systems. Technical Report UIB-2012-04, University of Ulm, Ulm (2012). URL http://dbis.eprints.uni-ulm.de/828/
29. Kotronis, C., Routis, I., Tsadimas, A., Nikolaidou, M., Anagnostopoulos, D.: A model-based approach for the design of cyber-physical human systems emphasizing human concerns. In: 2019 IEEE International Congress on Internet of Things (ICIOT), pp. 100–107. IEEE (2019)
30. Kristiansen, R., Trætteberg, H.: Model-based user interface design in the context of workflow models. In: M. Winckler, H. Johnson, P. Palanque (eds.) Task Models and Diagrams for User Interface Design, pp. 227–239. Springer Berlin Heidelberg, Berlin, Heidelberg (2007)
31. Langer, P., Mayerhofer, T., Wimmer, M., Kappel, G.: On the usage of UML: initial results of analyzing open UML models. In: H. Fill, D. Karagiannis, U. Reimer (eds.) Modellierung 2014, 19.-21. März 2014, Wien, Österreich, *LNI*, vol. 225, pp. 289–304. GI (2014). URL http://subs.emis.de/LNI/Proceedings/Proceedings225/article21.html
32. Li, X., Liu, Z., Schäf, M., Yin, L.: Autopa: Automatic prototyping from requirements. In: T. Margaria, B. Steffen (eds.) Leveraging Applications of Formal Methods, Verification, and Validation, pp. 609–624. Springer, Berlin, Heidelberg (2010)
33. Likert, R.: A technique for the measurement of attitudes. Archives of Psychology **22**(140), 55– (1932). URL http://psycnet.apa.org/record/1933-01885-001
34. Limbourg, Q., Vanderdonckt, J.: USIXML: A user interface description language supporting multiple levels of independence. In: M. Matera, S. Comai (eds.) Engineering Advanced Web Applications: Proceedings of Workshops in connection with the 4th International Conference on Web Engineering (ICWE 2004), Munich, Germany, 28-30 July, 2004, pp. 325–338. Rinton Press (2004)
35. Lindland, O.I., Sindre, G., Solvberg, A.: Understanding quality in conceptual modeling. IEEE Software **11**(2), 42–49 (1994). DOI 10.1109/52.268955
36. Lopez-Arredondo, L.P., Perez, C.B., Villavicencio-Navarro, J., Mercado, K.E., Encinas, M., Inzunza-Mejia, P.: Reengineering of the software development process in a technology services company. Business Process Management Journal (2019)
37. Machines, I.B.: AuraPortal. http://www.auraportal.com (2017)
38. Maqbool, B., Azam, F., Anwar, M.W., Butt, W.H., Zeb, J., Zafar, I., Nazir, A.K., Umair, Z.: A comprehensive investigation of bpmn models generation from textual requirementstechniques, tools and trends. In: International Conference on Information Science and Applications, pp. 543–557. Springer (2018)

39. Microsoft: Common UI Controls and Text Guidelines. https://docs.microsoft.com/en-us/previous-versions/windows/desktop/bb246433(v=vs.85) (2007)
40. Miers, D., White, S.A.: BPMN Modeling and Reference Guide: Understanding and Using BPMN, *10*, vol. 4, 2 edn. Future Strategies, Inc., Lighthouse Point, FL (2008)
41. Moody, D.L.: The method evaluation model: a theoretical model for validating information systems design methods. In: C.U. Ciborra, R. Mercurio, M. de Marco, M. Martinez, A. Carignani (eds.) Proceedings of the 11th European Conference on Information Systems, ECIS 2003, Naples, Italy 16-21 June 2003, pp. 1327–1336 (2003). URL http://aisel.aisnet.org/ecis2003/79
42. Object Management Group: Business Process Model and Notation. http://www.bpmn.org/. Accessed: 2019-07-19
43. Object Management Group: MOF 2.0 Query / Views / Transformations RFP. https://www.omg.org/spec/QVT/About-QVT/. Accessed: 2019-07-19
44. Office of Products and Programs, Technology Transformation Service: United States Web Design System. https://designsystem.digital.gov/documentation/. Accessed: 2019-07-19
45. Ouyang, C., Dumas, M., Aalst, W.M.P.V.D., Hofstede, A.H.M.T., Mendling, J.: From business process models to process-oriented software systems. ACM Transactions on Software Engineering and Methodology **19**(1), 2:1–2:37 (2009). DOI 10.1145/1555392.1555395. URL http://doi.acm.org/10.1145/1555392.1555395
46. Pintus, A., Patern, F., Santoro, C.: Modelling user interactions in web service-based business processes. In: Proceedings of the 6th International Conference on Web Information Systems and Technology - Volume 2: WEBIST,, pp. 175–180. INSTICC, SciTePress (2010). DOI 10.5220/0002774401750180
47. Polančič, G., Orban, B.: A bpmn-based language for modeling corporate communications. Computer Standards & Interfaces **65**, 45–60 (2019)
48. Radeke, F., Forbrig, P.: Patterns in task-based modeling of user interfaces. In: M. Winckler, H. Johnson, P. Palanque (eds.) Task Models and Diagrams for User Interface Design, pp. 184–197. Springer Berlin Heidelberg, Berlin, Heidelberg (2007)
49. Ramadan, Q., Strüber, D., Salnitri, M., Jürjens, J., Riediger, V., Staab, S.: A semi-automated bpmn-based framework for detecting conflicts between security, data-minimization, and fairness requirements. Software and Systems Modeling pp. 1–37 (2020)
50. Rieger, C.: Evaluating a graphical model-driven approach to codeless business app development. In: Proceedings of the 51st Hawaii International Conference on System Sciences (2018)
51. Rieger, C.: Interoperability of bpmn and maml for model-driven development of business apps. In: International Symposium on Business Modeling and Software Design, pp. 149–166. Springer (2018)
52. Ruiz, J., Serral, E., Snoeck, M.: Evaluating user interface generation approaches: model-based versus model-driven development. Software & Systems Modeling (2018). DOI 10.1007/s10270-018-0698-x. URL https://doi.org/10.1007/s10270-018-0698-x
53. Rusu, C., Rusu, V., Quiñones, D., Roncagliolo, S., Rusu, V.Z.: Evaluating online travel agencies usability: What heuristics should we use? In: International Conference on Social Computing and Social Media, pp. 121–130. Springer (2018)
54. Shamim, A., Balakrishnan, V., Tahir, M., Ahsan Qureshi, M.: Age and domain specific usability analysis of opinion visualisation techniques. Behaviour & Information Technology **35**(8), 680–689 (2016)
55. Softeam: e-Citiz. http://www.https://www.e-citiz.com/SOFTEAM (2019). Accessed: 2019-07-24
56. Sousa, K., Mendonça, H., Vanderdonckt, J.: A rule-based approach for model management in a user interface – business alignment framework. In: D. England, P. Palanque, J. Vanderdonckt, P.J. Wild (eds.) Task Models and Diagrams for User Interface Design, pp. 1–14. Springer, Berlin, Heidelberg (2010)
57. Sousa, K., Mendonça, H., Vanderdonckt, J., Rogier, E., Vandermeulen, J.: User interface derivation from business processes: A model-driven approach for organizational engineering. In: Proceedings of the 2008 ACM Symposium on Applied Computing, SAC '08, pp. 553–560. ACM, New York, NY, USA (2008). DOI 10.1145/1363686.1363821. URL http://doi.acm.org/10.1145/1363686.1363821
58. Sousa, K.S., Mendonça, H., Lievyns, A., Vanderdonckt, J.: Getting users involved in aligning their needs with business processes models and systems. Business Process Management Journal **17**(5), 748–786 (2011). DOI 10.1108/14637151111166178. URL https://doi.org/10.1108/14637151111166178
59. Systems Applications and Products in data processing (SAP): Fiori Design Guidelines. https://experience.sap.com/fiori-design/ (2007). Accessed: 2019-07-19

60. Torres, V., Pelechano, V.: Building business process driven web applications. In: S. Dustdar, J.L. Fiadeiro, A.P. Sheth (eds.) Business Process Management, pp. 322–337. Springer, Berlin, Heidelberg (2006)
61. Trætteberg, H.: Ui design without a task modeling language – using bpmn and diamodl for task modeling and dialog design. In: P. Forbrig, F. Paternò (eds.) Engineering Interactive Systems, pp. 110–117. Springer, Berlin, Heidelberg (2008)
62. van Welie, M., Trtteberg, H.: Interaction patterns in user interfaces. In: Proc. Seventh Pattern Languages of Programs Conference: PLoP 2000, pp. 13–16 (2000)
63. Whitefield, A., Wilson, F., Dowell, J.: A framework for human factors evaluation. Behaviour & Information Technology **10**(1), 65–79 (1991). DOI 10.1080/01449299108924272. URL `https://doi.org/10.1080/01449299108924272`
64. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M., Regnell, B., Wesslén, A.: Experimentation in Software Engineering: An Introduction. International Series in Software Engineering. Springer, Berlin (2012)
65. Yadav, S.B., Bravoco, R.R., Chatfield, A.T., Rajkumar, T.M.: Comparison of analysis techniques for information requirement determination. Commun. ACM **31**(9), 1090–1097 (1988). DOI 10.1145/48529.48533. URL `http://doi.acm.org/10.1145/48529.48533`
66. Yin, L., Liu, J., Ding, Z.: Modeling and Prototyping Business Processes in AutoPA. In: Proceedings of Fifth International Conference on Theoretical Aspects of Software Engineering, TASE '11, pp. 169–176 (2011). DOI 10.1109/TASE.2011.8
67. Yongchareon, S., Liu, C., Zhao, X.: Uniflexview: A unified framework for consistent construction of bpmn and bpel process views. Concurrency and Computation: Practice and Experience p. e5646 (2019)
68. Yongchareon, S., Liu, C., Zhao, X., Xu, J.: An artifact-centric approach to generating web-based business process driven user interfaces. In: L. Chen, P. Triantafillou, T. Suel (eds.) Web Information Systems Engineering – WISE 2010, pp. 419–427. Springer, Berlin, Heidelberg (2010)
69. Yongchareon, S., Liu, C., Zhao, X., Yu, J., Ngamakeur, K., Xu, J.: Deriving user interface flow models for artifact-centric business processes. Computers in Industry **96**, 66 – 85 (2018). DOI https://doi.org/10.1016/j.compind.2017.11.001. URL `http://www.sciencedirect.com/science/article/pii/S0166361516301762`
70. Zafar, I., Azam, F., Anwar, M.W., Maqbool, B., Butt, W.H., Nazir, A.: A novel framework to automatically generate executable web services from bpmn models. IEEE Access **7**, 93653–93677 (2019)
71. Zhang, Q., Chen, R., Zou, Y.: Reengineering user interfaces of e-commerce applications using business processes. In: Proceedings of 22nd IEEE International Conference on Software Maintenance, ICSM '06, pp. 428–437 (2006). DOI 10.1109/ICSM.2006.51
72. Zhao, X., Zou, Y., Hawkins, J., Madapusi, B.: A Business-Process-Driven Approach for Generating E-Commerce User Interfaces. In: G. Engels, B. Opdyke, D.C. Schmidt, F. Weil (eds.) Model Driven Engineering Languages and Systems, pp. 256–270. Springer, Berlin, Heidelberg (2007)