Modeling Structured and Unstructured Processes: An Empirical Evaluation

Evellin Cardoso[1], Katsiaryna Labunets[1], Fabiano Dalpiaz[2], John Mylopoulos[1], Paolo Giorgini[1]

1 University of Trento, Italy
{evellin.souzacardoso, katsiaryna.labunets, john.mylopoulos, paolo.giorgini}@unitn.it
2 Utrecht University, the Netherlands
f.dalpiaz@uu.nl

Abstract. Imperative process languages, such as BPMN, describe business processes in terms of collections of activities and control flows among them. Despite their popularity, such languages remain useful mostly for structured processes whose flow of activities is well-known and does not vary greatly. For unstructured processes, on the other hand, the verdict is still out as to the best way to represent them. In our previous work, we have proposed Azzurra, a specification language for business processes founded on social concepts, such as roles, agents and commitments. In this paper, we present the results of an experiment that comparatively evaluates Azzurra and BPMN in terms of their ability to represent structured and unstructured processes. Our results suggest that Azzurra is better suited than BPMN for unstructured business processes.

Keywords: Azzurra, BPMN, Specification Languages, Empirical Evaluation

1 Introduction

Business Process Management (BPM) is founded on the premise that process behavior has to be explicitly modeled, analyzed and managed along with software as a means for improving enterprise operations. In order to support such models, many process modeling languages have been proposed, including BPMN, EPCs, BPEL and more. Such languages are predominantly activity-centered [5, 12], in the sense that their modeling primitives [5] are founded on the notion of activity. Within this paradigm, imperative models express business processes as a set of activities inter-connected by control flow primitives inspired by Petri nets, finite state machines, and other system modeling frameworks dating back to the 50s and 60s. The distinguishing feature of imperative models is that they explicitly capture all possible execution paths for a business process.

Despite the popularity of activity-centered, imperative models—as evidenced by large industrial and academic adoption of the BPMN modeling language as de facto standard for process representation [16, 10]—activity-centered languages remain especially useful for routine, structured processes defined in terms of a specific set of behaviors. For unstructured processes, however, execution order is context-dependent and even the activities needed are unclear and/or undefined at design time. For such processes, as also pointed out by van der Aalst [1], activity-centered languages are an inflexible solution as they demand the identification of activities and control flows for the construction of a process model.

In our previous research, we have introduced Azzurra [6], a specification language for business processes that shifts the focus of representation from activities to *social commitments*. Formally, a commitment C(x,y,p,q) is a promise with contractual validity made by an agent x (debtor) to another agent y (creditor) that, if proposition p is brought about (antecedent), then proposition q will be brought about (consequent). By introducing correctness criteria for the enactment of a process, Azzurra abstracts away from specific activities (operationalizations) for achieving a goal; rather, Azzurra focuses on the outcomes of a process through the notion of a commitment's consequent. The elements of Azzurra suggest the hypothesis that it is more appropriate than its imperative cousins for unstructured processes, as they require more flexible specifications. To confirm/deny this hypothesis, we have conducted a preliminary study using scenarios that have been elaborated in [6], hoping to gain insights on the suitability of Azzurra for modeling unstructured processes.

The contribution of this paper is to report the results of a preliminary experiment performed with master's students at the University of Trento, to examine the suitability of Azzurra for unstructured processes. To this purpose, we designed and enacted an experiment to test two propositions about quality of structured and unstructured processes models represented in both Azzurra and BPMN. Here, model quality is defined in terms of the metrics of *precision* and *coverage* used in Ontology Engineering for evaluating the quality of ontologies [19]. Our results suggest that Azzura is less usable than BPMN in the sense that the social concepts it is founded on are less familiar to master's-level students in Computer Science. On the other hand, Azzura leads to better models, where "better" is defined in terms of the metrics of precision and coverage.

The rest of the paper is structured as follows: Section 2 provides the research baseline for our work, including classifications of processes from the BPM literature, together with a sketchy overview of current process modeling languages. Section 3 describes the experimental process, covering scope, plan, execution, analysis and interpretation of the results, including a general discussion of the findings. Finally, Section 4 summarizes the results and outlines future work.

2 Baseline

We discuss classifications for business processes in Section 2.1, and we briefly review the most prominent business process modeling languages in Section 2.2.

2.1 The Spectrum of Work in BPM

There exist several classifications for business processes according to their characteristics [8, 2]. A common classification scheme considers the level of structuring or predictability, thus dividing business processes into a spectrum of work of four types (see Figure 1) [8, 2, 4]. The level of structuring and predictability basically considers the extent to which the behavior of a given business process is predictable at modeling time:

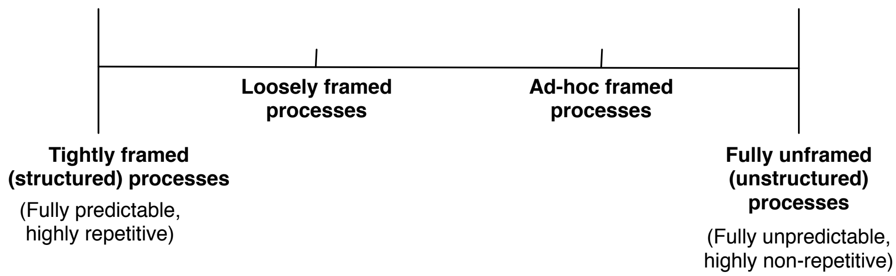

Fig. 1. The Spectrum of Work in BPM adapted from [17]

In the leftmost extreme of the spectrum, a *tightly framed (or structured) process* comprehends those processes whose execution of activities consistently follows a predefined process model [8, 2]. Since a formal representation of these processes can be easily described prior to their execution, tightly framed processes are characterized as fully predictable and repetitive and after their design-time description they can be repeatedly instantiated at runtime. Examples of this category are production and administrative processes [7] and as well as bank transactions that are executed in an exact sequence to comply with legal norms.

Even though tightly framed processes usually have a predictable behavior, a certain degree of unpredictability is expected due to the occurrence of exceptions and evolutions within the domain. Therefore, a *loosely framed process* corresponds to a process in which it is possible to represent the process behavior and a set of constraints a priori [2], such that the process model describes the "standard way of doing things" while requiring additions, removals or generation of alternative sequence of activities during runtime [7].

Differently from a tightly and loosely framed processes that can be described a priori by an explicit process model, the behavior of *ad-hoc framed process* cannot be determined in terms of a explicit process logic during design time due to a lack of domain knowledge or the complexity of task combinations. Instead, only structured fragments can be identified a priori and properly composed on a per-case basis, while process parts that are undefined or uncertain can only be specified and incorporated as the process evolves [7].

Finally, within the rightmost category of the spectrum, *fully unframed (or unstructured) processes* have sufficient variability in such way that no process description can be pre-defined at all [7, 2]. As a result, process participants need to make decisions using their knowledge to create activities on demand. The creation of such activities is based on situation-specific parameters whose values are determined as the process execution proceeds. Besides choosing activities on demand, they also dynamically decide the execution order of such activities.

2.2 Process Modeling Languages

Although the disparities regarding the nature of process behavior in reality trigger process modeling languages to accommodate such diversity, contemporary techniques for process modeling are predominantly *activity-centered* [5, 12], although over the past years an *artifact-centered* approach has also emerged [12].

The activity-centered paradigm elects the concept of *activity* as its first class modeling construct [5] in order to express business processes as a set of activities. Within activity-centered models, a plethora of conceptual languages like BPMN, BPEL, UML, EPCs represent business processes within an imperative (or procedural) paradigm that is basically founded on the notions of activities and a number of causal dependencies among such activities. The paradigm requires modelers to explicitly represent the causal activations of activities and therefore, all possible paths executed by the business process have to be also exhaustively enumerated during modeling time.

The rigidity imposed by the imperative paradigm triggered the development of (activity-centered) declarative languages. In this context, declarative workflows [1] have arisen as a more flexible alternative for the specification of business processes by enabling the representation of behavior in terms of minimal precedence constraints among activities. By default, all execution paths are allowed and prohibited execution paths are specified by constraints on the execution order between activities.

Unlike the activity-centered paradigm, the artifact-centered paradigm represents the states of artifacts (also denominated as data objects) that are used throughout the process and how these states are changed/updated by activities [17]. Further, the paradigm also complements the representation of processes in relation to declaratives languages as it focuses on a hybrid approach of the representation of data and activities that update such data objects.

Deviating from the current trends of process representation, we introduced the Azzurra [6] specification language for business processes that shifts the focus from activities and data objects to agents, roles, *social commitments* and protocols. In Azzurra, business processes are represented as protocols that are carried out by intentional agents and roles. Such agents and roles have expectations in relation to each other that are modeled in terms of *social commitments*. Formally speaking, a social commitment C(x,y,p,q) is a promise with contractual validity made by an agent x (debtor) to another agent y (creditor) that, if proposition p is brought about (antecedent), then proposition q will be brought about (consequent). Commitments' consequents specify correctness criteria that have to be respected, rather than capturing how to achieve a determined business goal through a prescription of a number of steps (activities). This shift in the modeling paradigm opens up the possibility of providing more flexible specifications for business processes as it allows the participating agents to decide the best operationalizations to achieve the outcomes during runtime.

In light of the assumption that Azzurra provides a more flexible solution for the specification of business processes, we have performed a preliminary evaluation of the language by means of two scenarios in [6]. Both scenarios have been extracted from the BPM literature as representatives of business processes that require flexible specifications. More specifically, Scenario 1 (Fracture treatment) intended to compare Azzurra's representational features with the representational features of current modeling languages, namely procedural, declarative and data-centered approaches. The conclusion of such comparison led us to the realization that Azzurra focuses on different aspects of current modeling languages in order to represent business processes. As a consequence of that, our intuition rests on the realization that this shift of focus can better capture the features of unstructured business processes. Therefore, Scenario 2 (Transient Ischemic Attack (TIA) Clinical Guideline) has been chosen due to its unstructured nature to check this intuition that Azzurra better supports the representation of such kind of processes. In this context, we enumerated the domain representational needs of unstructured processes such as the absence of genuine activities to be executed as well as the lack of ordering constraints between such activities and compared both representations in Azzurra and BPMN of the TIA clinical guideline. A direct conclusion of such comparison indeed established that Azzurra is better than BPMN for unstructured processes.

With these insights at hand, in this paper we perform an experiment with students to check the validity of our insights regarding the suitability of Azzurra and BPMN for structured and unstructured processes. BPMN has been chosen for the comparison under consideration due to its wide acceptance and popularity as a standard for business processes representation [16, 10]. More specifically, with this experiment, we want to acquire objective and statistically significant evidence regarding the suitability of Azzurra for unstructured processes. In order to perform the experiment, we elaborated the following propositions:

P1. Azzurra produces models of better quality than BPMN in the representation of unframed (unstructured) business processes;

P2. BPMN produces models of better quality than Azzurra in the representation of tightly framed (structured) business processes.

3 The Experiment Process

The design of our experiment has been conducted on the basis of guidelines for experimentation in software engineering [20, 13]. According to such guidelines, the experiment process can be divided into five main activities depicted in Fig. 2.

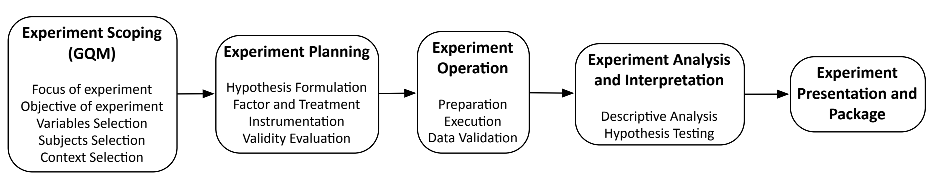

Fig. 2. The Experimentation Process According to [20]

Within the **Scoping** activity, the experiment is defined in terms of problem statement and goals, defining *why* the experiment is needed. According to the Wohlin's guidelines [20], the Goal, Question, Metric (GQM) template [3] comprehends a suitable instrument for defining the scope of a given experiment. Our GQM template is described in Section 3.1.

The **Planning** activity is the phase in which the foundation of the experiment is laid, defining *how* it is conducted. The steps conducted in the scope of our planning activity are described in Section 3.1.

The **Operation** activity encompasses the preparation of subjects and required material on which the experiment in executed (i.e., *objects*), the actual execution of the experiment as well as the collection of measurements (see Section 3.2). The **Analysis and Interpretation** activity focuses on qualitatively and quantitatively processing the outcomes of the experiment (Section 3.3 and Section 3.4). Finally, the results are presented in the course of the **Presentation and Package** (leading to the present paper).

3.1 Experiment Scoping and Planning

Our experiment starts by scoping its objectives using the GQM template depicted in Table 1:

Table 1. GQM for our experiment

Focus of the experiment: Analyze Azzurra specification language and compare it with the BPMN modeling language.
Objective of the experiment: Checking the adequacy of the Azzurra and BPMN languages for the representation of structured and unstructured business processes.
Variables selection: We compare Azzurra and BPMN modeling languages in terms of model quality.
Subject: From the point of view of M.Sc. students enrolled in classes of Organizational Information Systems.
Context of the experiment: M.Sc. students creating Azzurra and BPMN models.

In the following, the planning phase of our experiment required us to elaborate the *hypotheses* (together with the *independent* and *dependent variables*), *factors* and *treatments* applied to our experiment.

Hypothesis Formulation. As we intend to compare Azzurra and BPMN for structured and unstructured processes, we construct three null hypotheses, one for each factor and a third one for the interaction between the factors [20].

- **Null Hypothesis $H_{0\text{-}1}$:** There is no significant difference in model quality of Azzurra and BPMN modeling languages.
- **$H_{a\text{-}1}$:** There is significant difference in the model quality of Azzurra and BPMN modeling languages.
- **Null Hypothesis $H_{0\text{-}2}$:** There is no significant difference in model quality of structured and unstructured scenarios.
- **$H_{a\text{-}2}$:** There is significant difference in model quality of structured and unstructured scenarios.

- **Null Hypothesis H_3:** There are no significant interactions between the type of modeling language and types of business processes in terms of model quality.
- **H_{a}-3:** There are significant interactions between the type of modeling language and types of business processes in terms of model quality.

Note that our hypotheses are elaborated in terms of model quality (*dependent variable*). In order to select the metrics for measuring model quality in our evaluation, we get inspiration from the field of Ontology Engineering; more precisely, we use a formal evaluation framework [19] that defines the dimensions of *precision* and *coverage* to define the quality of a given ontology (model).

In [19], a conceptualization comprehends a set of conceptual relations about a certain portion of reality perceived by an agent, defining a set of intended models I_K. In this context, the role of an ontology is to provide a specification of such conceptualization, precisely capturing the intended models according to such conceptualization and excluding the non-intended ones. Considering that it is not always easy to find the right set of entities so that an ontology admits only the intended models [9], ontologies are considered only approximations of conceptualizations. Consequently, the formal framework of Staab et al. [19] proposes a schema for evaluating ontologies with respect to the degree of approximation they can provide to their respective conceptualizations. To evaluate such degree of approximation, the *precision* and *coverage* metrics are introduced and can be mathematically defined as:

$$P = \frac{|I_K \cap O_K|}{|O_K|} \textit{(precision)} \quad C = \frac{|I_K \cap O_K|}{|I_K|} \textit{(coverage)}$$

In Ontology Engineering, *precision* measures how much the represented models O_K are relevant according to the set of intended models I_K, while *coverage* measures how much of the intended models I_K are represented by the ontology O_K. We use analog reasoning for our evaluation of Azzurra and BPMN modeling languages. In our case, business processes are considered the target conceptualization that can be represented by two distinct ontologies, i.e., the Azzurra and BPMN modeling languages. Every business process has a natural language description that admits a number of execution paths (in our case, the set of intended models I_K corresponds to the set of intended execution paths $I_{execPath}$) and specifications in BPMN and Azzurra provide representation of such execution paths ($R_{execPath}$). Therefore, *precision* measures how many paths which are represented in the model are correct in relation to the intended paths prescribed by the natural language description, while *coverage* measures how many paths provided in the natural language description are indeed captured in the model representation. In our case, *precision* and *coverage* are mathematically defined as follows:

$$P = \frac{|I_{execPath} \cap R_{execPath}|}{|R_{execPath}|} \textit{(precision)} \; C = \frac{|I_{execPath} \cap R_{execPath}|}{|I_{execPath}|} \textit{(coverage)}$$

Factor and Treatment. As the aim of our experiment is to investigate whether the Azzurra modeling language has a more faithful representation of unstructured business process than the BPMN modeling language, we have two factors: factor A is the type business process modeling language (whose treatments are Azzurra and BPMN modeling languages) and factor B is the type of business process under consideration (whose treatments are unstructured and structured business processes). Factors and treatments are depicted in Table 2:

Table 2. Factors and Treatments applied in our experiment

Language Type (Factor A) / BP Type (Factor B)	Azzurra	BPMN
Structured		
Unstructured		

Instrumentation. Participants used a free online modeling tool[1] for the elaboration of BPMN 2.0 models and a plug-in[2] developed at University of Trento for the elaboration of Azzurra models. In the end of the experiment, they provided the source of Azzurra and BPMN models for later evaluation of the results.

Validity evaluation. We enumerate the main threats to the validity of our experiment using the Wohlin's categorization [20]:

Threats to construct validity. The threats in this category are: (i) a major threat to construct validity is that the chosen business processes may not be representative samples for the structured and unstructured types of business processes. To mitigate this issue, we have chosen already consolidated scenarios within the BPM literature as representatives from structured and unstructured processes; (ii) furthermore, the domain knowledge involved in the description of the scenarios may entail some difficulty during the modeling process; (iii) the fact that BPMN is an imperative language, while Azzurra is declarative may also entail additional difficulties as there is some evidence that imperative languages are more understandable than declarative ones [15]; (iv) hypothesis guessing may also represent a threat as subjects can be conditioned by the results they are providing. We mitigated this threat by carefully formulating questions on the basis of correct usage and preference of modeling languages.

Threats to external validity. Here, our largest threat is the usage of students as subjects in our experiment. Further, they had prior training in BPMN and UML activity diagrams during the course lectures. To mitigate these issues and make their background more uniform, we have provided preliminary training in both Azzurra and BPMN languages by means of one example. In order to encourage subjects to participate, they could earn at most one point in the overall course grade on the basis of the correct usage of languages constructs.

[1] www.lucidchart.com

[2] https://trinity.disi.unitn.it/azura/azura/

Threats to conclusion validity. The two threats to conclusion validity are the low number and homogeneity of the samples (students) that may impact our ability to reveal patterns in the data. Besides that, the first author of this paper evaluated the number of admissible execution paths for each scenario, together with their respective representations in Azzurra and BPMN.
Threats to internal validity. This type of validity is threatened by the effect of order in which the subjects apply the treatments (structured and unstructured) as students may learn the content of natural languages descriptions, and the second models are easier to produce. To mitigate the effect of order, the order is assigned randomly to each subject. By having the same number of subjects starting with the first treatment as with the second, the design is balanced [20].

3.2 Experiment Operation

Preparation. We continue following the same rationale of evaluation through modeling scenarios. In particular, we have used same business process from Scenario 2 used in [6] (i.e., the TIA clinical guideline) as a representative of unstructured business process and the X-Ray Medical Order (extracted from [17]) as the representative of structured business process. The selection of both scenarios as representatives of unstructured and structured business process has been supported by BPM literature that positions clinical guidelines as unstructured processes [7] and the X-Ray Medical Order as a structured process [17].

Next, a natural language description[3] has been extracted from literature in order to be applied on the subjects. Further, the corresponding Azzurra and BPMN models have been built in advance for each scenario by the first author with the purpose of ensuring that process models to be built in each scenario indeed covered the core concepts of both modeling languages.

Experiment execution. The experiment has been conducted in July 2015 with master's students in Computer Science in the scope of the Organizational Information Systems Course at University of Trento. In total, 17 subjects participated in this empirical test. The experiment has been structured in different parts:

- **Introduction Phase (15 min)**: General instructions about the experiment and introduction to Azzurra modeling language and modeling tool together with a presentation about BPMN. It is also important to note that students had prior contact with BPMN along the course lectures;
- **Experiment phase (40 min, i.e., 20 min for each language)**: Group 1 models the structured scenario using Azzurra and BPMN, whereas group 2 models the unstructured scenario using Azzurra and BPMN;
- **Questionnaire phase (15 min)**: General questions concerning the background of the subject and questions regarding the elaboration of models relative to scenario 1 and 2.

[3] Scenario descriptions, experimental results and data analysis are available at https://www.dropbox.com/s/8qlwd5svqbt3hmw/Empirical%20evaluation.zip?dl=0

Data validation. The obtained data were checked for consistency and plausibility. We discarded the inputs from two students due to incompleteness; thus, we could employ data from 15 students in the data analysis.

3.3 Experiment Analysis and Interpretation

To report experimental results, Table 3 shows mean, median and standard deviation values for precision and coverage by language and process type:

Table 3. Precision and Coverage by Language and Process Type

		Azzurra			BPMN		
		Mean	Median	Std. dev.	Mean	Median	Std. dev.
Unstructured	Precision	1.00	1.00	0.00	1.00	1.00	0.00
	Coverage	0.89	1	0.18	0.34	0.36	0.07
Structured	Precision	1.00	1.00	0.00	0.95	1.00	0.13
	Coverage	0.82	0.75	0.19	0.82	0.75	0.19
Overall	Precision	1.00	1.00	0.00	0.97	1.00	0.09
	Coverage	0.85	1.00	0.18	0.60	0.50	0.28

We conducted statistical analysis to test whether the null hypothesis H_0 can be rejected, thereby allowing us to draw conclusions about our studied phenomenon: the modeling of structured and unstructured business processes.

For the selection of the statistical tests, we followed the guidelines prescribed by Harvey [11, Chap. 37]. As the participants of our experiment applied both methods, to test $H_{0\text{-}1}$, we can use paired t-test or its non-parametric analog, Wilcoxon test. However, the participants did not switch scenario type and, therefore, to test $H_{0\text{-}2}$ we use unpaired t-test or its non-parametric analog, Mann-Whitney (MW) test. Finally, to test $H_{0\text{-}3}$ we need to investigate the difference between the combination of two factors (type of language and type of process), which requires ANOVA test or its non-parametric analog, Kruskal-Wallis (KW) test [20]. We checked the normality of data by Shapiro-Wilk test which returned $p\text{-}value = 0.0013$ for coverage and $p\text{-}value = 6.8 \cdot 10^{-11}$ for precision. Thus, we used non-parametric tests for all three hypothesis. Further, for all statistical tests we use a threshold of 5% for α, the probability of committing Type-I error [20].

Null Hypothesis $H_{0\text{-}1}$ (Azzurra vs. BPMN): The results of the Wilcoxon test revealed a statistically significant difference between two modeling languages with respect to coverage (test results: $W = 7$, $Z = 2.09$, $p\text{-}value = 0.04$, Cohen's d = 1.06) and no significant difference in precision ($p\text{-}value = 0.32$). The power of the Wilcoxon test for coverage is 0.72. Therefore, we cannot reject the null hypothesis both for coverage and precision. However, to achieve 80% power for coverage we would need a sample size of 16 participants, while we had 13 participants. For Azzurra, the overall mean coverage is 0.85, whereas for BPMN the overall mean coverage is 0.6. As coverage describes the percentage of the intended interpretations (according to the natural language description) that are indeed captured by the model, a mean coverage of 0.85 means that

85% of all intended paths are captured in the model, whereas 15% of them are not. In fact, this is a reasonable advantage from Azzurra, once the language specifies process paths in terms of correctness criteria, whereas BPMN requires a more verbose style of specification, demanding exhaustive specification of all potential process paths. It is natural that some intended process paths are not captured in the BPMN representation. Observe also the significant difference in terms of coverage between Azzurra (0.893) and BPMN (0.345) for unstructured processes. As unstructured processes potentially have a large number of process paths, this difference in terms of coverage between both languages becomes even more evident for such kind of processes.

Null Hypothesis $H_{0\text{-}2}$ (Structured vs. Unstructured): To test this hypothesis, we should use MW test which assumes the equality of variance. However, the Levene's test for homogeneity of variance returned *p-value* $= 0.37$ for precision and *p-value* $= 0.04$ for coverage. Therefore, we cannot rely on the results of the MW test for coverage. To mitigate this issue, we cross-validate the results of MW test with KW test which does not require equal variance. The MW test results did not reveal significant difference between two process types both for precision (*p-value* $= 0.35$) and coverage (*p-value* $= 0.11$). The KW test returned *p-value* $= 0.11$ for coverage, which supported the results of MW test. In order to achieve statistically significant results for coverage with 80% power we would need a sample size of 54 participants. The results show that the process type did not affect the performance of the participants. The null hypothesis $H_{0\text{-}2}$ cannot be rejected for any of the variables.

Null Hypothesis $H_{0\text{-}3}$ (Language & Process Type): The results of KW test revealed a statiscally significant effect of the combination of language and process type on coverage ($\chi^2(3) = 15$, *p-value* $= 0.002$) and no effect on precision (*p-value* $= 0.44$). Therefore, the null hypothesis $H_{0\text{-}3}$ can be rejected only for coverage. A post-hoc test using MW test with Holm correction showed the significant differences between coverage of the results produced by participants who used BPMN on unstructured process and other participants who used BPMN on structured process (MW test results: *p-value* $= 0.002$, Cohen's d $= 3.23$) or Azzurra on unstructured (*p-value* $= 0.003$, Cohen's d $= 4.02$) and structured process (*p-value* $= 0.002$, Cohen's d $= 3.23$). It means that there is a significant difference in terms of coverage between Azzurra and BPMN for unstructured processes, as described above, whereas for structured processes both Azzurra and BPMN have equal performance in terms of coverage.

3.4 Discussion

Our aim is to investigate the suitability of the Azzurra language for representing unstructured processes and its superiority in terms of model quality in relation to BPMN. In our approach, model quality is measured in terms of *precision* and *coverage*, two metrics extracted from the field of Ontology Engineering for the evaluation of ontology quality. Regarding our propositions introduced in Section 2.2, our findings suggest that:

P1. The Azzurra modeling language is significantly better than BPMN in terms of coverage for the representation of unstructured processes, but the power of the test is not enough to completely reject null hypothesis H_{0-1} (see the discussion of null hypothesis $H_{0\text{-}1}$).

P2. No definite conclusion can be drawn, due to the absence of statistically significant difference between the two modeling languages with respect to precision (see the discussion of null hypothesis $H_{0\text{-}1}$).

The superiority of Azzurra over BPMN in terms of coverage for unstructured processes can be explained by the representational style of Azzurra and BPMN: Azzurra requires correctness criteria to be specified as commitment's consequents, whereas BPMN imposes the need of exhaustive specification of all activities and paths. First, if we consider the advantage of Azzurra over BPMN in terms of coverage—therefore measuring how many intended paths are indeed captured by its corresponding representation—, an Azzurra representation "covers" more paths than its counterpart in BPMN, as Azzurra's correctness criteria captures all possible paths in an implicit way as opposed to explicitly capturing all paths. Therefore, there is a higher chance that some paths are indeed forgotten during the modeling process in a BPMN representation.

Second, considering Azzurra's suitability for unstructured processes, these processes are characterized by an "on-the-fly" creation of activities, lacking also a pre-defined execution order among activities. Therefore, their textual description allows several interpretations regarding the potential paths to be captured (e.g., for three activities A, B and C, it is possible to capture 3! paths). Azzurra's features can cope better than BPMN with both aspects of unstructured processes: via commitments, modelers can specify obligations to be fulfilled and participants can dynamically select which activities to perform to fulfill such obligations at runtime. Further, a commitment-based representation also allows one to specify lack of structure necessary for unstructured processes, refraining from capturing a specific order to fulfill them. Differently, as we have noticed during the evaluation of experiment's results, students commonly captured only the most trivial sequence of activities in BPMN, missing all the other possible interpretations according to the natural language description.

Our experimental evaluation considered the metrics of *precision* and *coverage* to determine the quality of models representations in terms of domain faithfulness and language expressiveness, rather than the focusing on the modelers' perception. To overcome this issue, we distributed a questionnaire among participants. In this survey, there is significant preference of BPMN in relation to Azzurra. This answer should be interpreted with care for two reasons. First, the questionnaire revealed prior process modeling experience of subjects in BPMN both in academia and industry. Second, imperative process modeling has its roots in imperative and declarative computer programming languages which have been used in computer science since the 50s and 60s. Third, there is evidence that imperative languages are more understandable than declaratives ones [15]. As familiarity is a very important aspect for the usability of modeling languages, preference of BPMN seems to natural in this case.

Although we effectively conducted the experiment with a homogeneous group of master's students, some limitations must be considered. In particular, the relatively low number of experimental subjects constitutes a limitation in terms of statistically significance of our conclusions. Moreover, while BPMN models have been produced on the basis of a professional tool, the usage of a prototypical implementation of the Azzurra modeling tool may be also considered a disadvantage in relation to its respective counterpart in BPMN models.

4 Conclusion

In this paper, we empirically evaluated the Azzurra and BPMN modeling languages for the representation of structured and unstructured processes in terms *precision* and *coverage*, two metrics used in the evaluation of ontology quality in the field of Ontology Engineering. Our empirical results indicate that Azzurra can be considered superior to BPMN for the representation of unstructured processes. However, no further claims can be stated concerning the superiority of BPMN over Azzurra for the representation of structured processes.

A very natural direction for our future work is the replication of our experiment. In that respect, we first envision an experimental design that encompasses a higher number os students in order to be able to validate some of our hypothesis (e.g., the difference of structured and unstructured processes). Alternatively, we would be also interested in repeating the similar experiment with BPM experts within an industrial setting. The adoption of industrial experts would allow us to not only gain more statistical power in our analysis, but could be also instrumental for acquiring insights regarding the acceptance of Azzurra within the industry. A second future work direction for our work concerns the elaboration of modeling patterns and guidelines for process representation using Azzurra, similarly as the existent ones for BPMN [14]. Finally, the usage of the same dataset with different metrics for the evaluation of process models (as the one proposed in [18]) could yield us different conclusions regarding the suitability of both process languages.

Acknowledgement

The research leading this paper has been funded by ERC advanced grant 267856 "Lucretius: Foundations for Software Evolution", unfolding during the period of April 2011 - March 2016. It has also received fundings from the SESAR Joint Undertaking under grant agreement No 699306 under the European Union's Horizon 2020 research and innovation program.

References

1. van der Aalst, W.M.P., Pesic, M., Schonenberg, H.: Declarative Workflows: Balancing between Flexibility and Support. Computer Science-Research and Development 23(2), 99–113 (2009)

2. van der Aalst, W.M.P.: Business Process Management: A Comprehensive Survey. ISRN Software Engineering 2013 (2013)
3. Basili, V.R., Caldiera, G., Rombach, H.D.: The Goal Question Metric Approach. In: Encyclopedia of Software Engineering. Wiley (1994)
4. vom Brocke, J., Rosemann, M.: Handbook on Business Process Management 1: Introduction, Methods, and Information Systems (2010)
5. Cohn, D., Hull, R.: Business Artifacts: A Data-centric Approach to Modeling Business Operations and Processes. IEEE Data Engineering Bulletin 32(3), 3–9 (2009)
6. Dalpiaz, F., Cardoso, E., Canobbio, G., Giorgini, P., Mylopoulos, J.: Social Specifications of Business Processes with Azzurra. In: Proc. of the International Conference on Research Challenges in Information Science. pp. 7–18 (2015)
7. Di Ciccio, C., Marrella, A., Russo, A.: Knowledge-intensive Processes: Characteristics, Requirements and Analysis of Contemporary Approaches. Journal on Data Semantics 4(1), 29–57 (2015)
8. Dumas, M., van der Aalst, W.M., ter Hofstede, A.H.: Process-Aware Information Systems: Bridging People and Software Through Process Technology. Wiley (2005)
9. Guarino, N.: Formal ontology and information systems. In: Proc. of Formal Ontology in Information Systems (1998)
10. Harmon, P.: The State of Business Process Management 2016. Tech. rep., BPTrends (2016)
11. Harvey, M.: Intuitive Biostatistics. New York: Oxford University Press (1995)
12. Hull, R.: Artifact-Centric Business Process Models: Brief Survey of Research Results and Challenges. In: Proc. of the OTM Confederated International Conferences. pp. 1152–1163 (2008)
13. Juristo, N., Moreno, A.M.: Basics of Software Engineering Experimentation. Springer (2010)
14. Mendling, J., Reijers, H.A., van der Aalst, W.M.P.: Seven Process Modeling Guidelines (7PMG). Information and Software Technology 52(2), 127–136 (2010)
15. Pichler, P., Weber, B., Zugal, S., Pinggera, J., Mendling, J., Reijers, H.A.: Imperative versus Declarative Process Modeling Languages: An Empirical Investigation. In: Proc. of BPM International Workshops, pp. 383–394 (2012)
16. Recker, J.: BPMN Research: What We Know and What We Don't Know. In: Proc. of the International Workshop on Business Process Model and Notation. pp. 1–7 (2012)
17. Reichert, M., Weber, B.: Enabling Flexibility in Process-Aware Information Systems - Challenges, Methods, Technologies. Springer (2012)
18. Rozinat, A., De Medeiros, A.A., Günther, C.W., Weijters, A., Van der Aalst, W.M.: Towards an Evaluation Framework for Process Mining Algorithms. BPM Center Report BPM-07-06, BPMcenter. org p. 10 (2007)
19. Staab, S., Gomez-Perez, A., Daelemana, W., Reinberger, M.L., Noy, N.F.: Why Evaluate Ontology Technologies? Because it Works! IEEE Intelligent Systems 19(4), 74–81 (Jul 2004)
20. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A.: Experimentation in Software Engineering: An Introduction. Kluwer Academic Publishers, Norwell, MA, USA (2000)