Precise vs. Ultra-light Activity Diagrams - An Experimental Assessment in the Context of Business Process Modelling

Francesco Di Cerbo[1], Gabriella Dodero[1] Gianna Reggio[2], Filippo Ricca[2], and Giuseppe Scanniello[3]

[1] CASE, Libera Università di Bolzano-Bozen, Italy,
francesco.dicerbo|gabriella.dodero@unibz.it
[2] DISI, Università di Genova, Italy,
filippo.ricca|gianna.reggio@disi.unige.it
[3] Dipartimento di Matematica e Informatica, Università della Basilicata, Italy,
giuseppe.scanniello@unibas.it

Abstract. UML activity diagrams are a commonly used notation for modelling business processes in the field of both workflow automation and requirements engineering. In this paper, we present a novel precise style for this notation. Further, the effectiveness of this style has been investigated in the context of the modelling of business processes through a controlled experiment conducted with master students in Computer Science at the Free University of Bolzano-Bozen. The results indicate that the subjects achieved a significantly better comprehension level when business processes are modelled using the precise style with respect to a "lighter" variant, with no significant impact on the effort to accomplish the tasks.

Keywords: Business Process modelling, UML activity diagrams, Controlled experiment, Precise and Ultra-light styles.

1 Introduction

In the last years, many organizations have been changing their business processes to be competitive in the global market [8]. In this context, modelling, management, and enactment of business processes are considered relevant to support organizations in their daily activities. Concerning the modelling of business processes, a number of process definition languages have been proposed in the literature, based on several formalisms such as BPMN (Business Process Modeling Notation) [17], event-condition-action mechanisms [1], graph rewriting mechanism [11], Petri Nets [2], etc. More recently, some authors have suggested exploiting UML (Unified Modeling Language) [19] to model business processes [14, 16].

UML represents a natural choice for modelling business processes since it has been conceived for the communication among people and then can be easily understood and used by customers, managers, and developers [16]. Process modelling also plays an important role in the requirement engineering field [10]. It is an essential mechanism for specifying the processes to be supported by a software system as well as for communication with customers and end-users, who have to understand and possibly review these processes [9]. In this scenario UML activity diagrams are a commonly used notation for business process modelling.

In favour of the UML notation is its flexibility allowing the modeller to choose the preferred degree of precision/abstractiveness to build models. Concerning the business process modelling, different options are available ranging from "light" styles, where nodes and arcs of the activity diagrams are simply decorated by natural language text, to more rigorous ones, where for example nodes and arcs are expressed in a formal language. "Light" activity diagrams are simple to write/use but their inherent ambiguity complicates the communication among participants. On the other hand, more precise/rigorous notations are more complex to use but limit ambiguity and have the good quality to be more easily transformed into executable models (e.g., expressed in BPEL).

In this paper, we sketch our *precise* UML activity diagrams to model business processes. This style has been proposed and used in the context of the TECDOC project[1] along with other variants: *ultra-light*, *light*, *precise* and *conceptual precise*[2]. The effectiveness of the *precise* style has been investigated through a controlled experiment conducted with master students in Computer Science at the Free University of Bolzano-Bozen. Indeed, we compared the comprehension of the subjects on business processes specified by using *precise* and *ultra-light* (the "lightest" variant) UML activity diagrams.

The remainder of the paper is organized as follows: Sect. 2 introduces both the precise and the ultra-light styles. Sect. 3 presents the design of the controlled experiment, while Sect. 4 shows and discusses the achieved results. Sect. 5 presents relevant related literature concerning experiments in the use of UML models in comprehension tasks, while final remarks conclude the paper.

2 Process modeling with UML: Ultra-light and precise styles

In this section we present the two styles for business process modelling that we intend to contrast with respect to comprehension level and comprehension effort.

We will use the following terminology: – the basic activities in a business process are the *basic task* of the process; – the *process objects* are the entities over which the activity of the process are performed, obviously these entities are passive, i.e., they are not able to do such activities by themselves; – the active entities that perform the various tasks are *process participants* (entities

[1] Funded in the framework of research activities of Ligurian Technology District SIIT (Integrated Intelligent Systems and Technologies), the TECDOC project aimed to define methodologies to efficiently schedule, coordinate, monitor and manage the different operational activities related to the management of Complex Organizations

[2] See http://softeng.disi.unige.it/tech-rep/TECDOC.pdf for the complete TECDOC document

playing a certain role in a domain), and whenever it will be relevant, we will distinguish the human participants from those corresponding to software and hardware systems.

2.1 Ultra-light style

The ultra-light style is the one currently used in the industry for UML business process modelling, see e.g., [15]. Following the *ultra-light style* a process is modelled by a UML activity diagram, where the nodes (activity and object) and the guards on the arcs leaving the decision nodes are decorated by natural language text, which follows neither rules nor patterns. Notice that it may happen that the sentences defining the activities may be either in active or passive form (e.g., "Clerk fills the form" and "'Form is filled") and that the entity executing the activity may be precisely determined or left undefined (e.g., "Form becomes filled"). It is also possible that nominal sentences are used instead of verbal phrases ("Filling the form"). Also, the objects over which the process activities are performed may be described in different ways, for example by a noun (e.g., "Form", "The form") or by a qualifying sentence (e.g., "Client form", "Filled form", "Sent form").

Participants of the process may be modelled only by introducing swim-lanes and titles of the various lanes. The objects produced and consumed by the activities of a business process may optionally be made explicit by using object nodes.

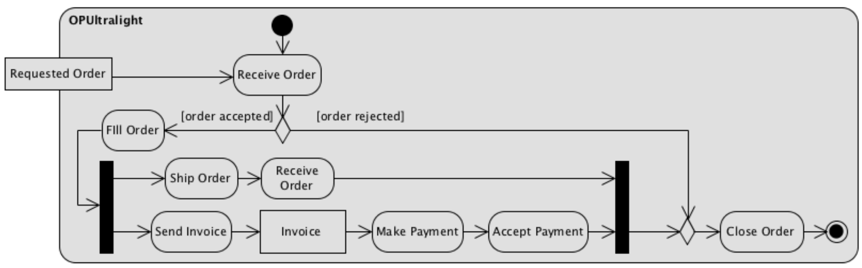

Fig. 1. Ultra-light model of Process Order

In Fig. 1, we present the *ultra-light* UML model of the business process Process Order, namely one of the objects used in the experiment. It is a parametric activity diagram that receives as input the Requested Order (see the node at the boundary of the activity diagram) for an on-line shop. Tasks are represented in the model as rounded rectangles, while the produced objects are depicted as rectangles. The activity diagram describes how the order is managed by the online shop. It is quite easy to understand and there is no need to further comment it.

2.2 Precise style

The participants and the objects of a business process modelled by the precise style are explicitly listed and precisely modelled with UML by means of classes. The behavioural view of the process is given by an activity diagram where actions and conditions will be written by using respectively the language for the action of UML and OCL [18], the textual language for boolean expressions part of the UML. Whenever the object nodes will be used, they should be typed by UML classes and data types; and if swim-lanes are used, they should be given titles by participants.

Thus the *UML precise model of a business process* consists of a class diagram, introducing the classes needed to type its participants and objects, a list of its participants and objects, and an activity diagram representing its behaviour:

- Classes in the class diagram may be stereotyped by ≪object≫ (process objects), ≪businessWorker≫ and ≪system≫ (process participants distinguishing between human beings and hardware/software systems). For readability the stereotype ≪businessWorker≫ is usually omitted. Elements of those classes may be described using the many tools offered by the UML, for example constraints and behavioural diagrams, and their mutual relationships will be expressed by associations and specializations. Dependency (visually depicted by a dashed arrow) will be used to represent the fact that participants of a given class will act over objects of another given class.
- Participants will have a name and will be typed by a class with stereotype either ≪businessWorker≫ or ≪system≫, and objects also will have a name and will be typed by a class stereotyped by ≪object≫. Notice that participants/objects are roles for the entities taking part to the process, and not specific individuals. Constraints might be imposed on participants and objects of a process.
- Basic tasks in the activity diagrams are modelled by UML actions (i.e., calls to class operations, belonging to those classes describing types of participants and objects, and the standard statements, e.g., assignment, creation and destruction of objects). Nodes in the activity diagram will correspond to basic tasks, and thus they will be action nodes, and the conditions on arcs leaving the decision nodes will be OCL expressions (e.g., ORDER.acceptable in Fig. 2). Participants and objects will freely appear both in the actions and in the conditions.

Fig. 2 shows the precise model of the Process Order case. In this process we have two participants (i.e., human being) the Client and the Company, and three business objects: Order, Payment and Invoice. The three objects are related among them as shown by the constraints in the participants/objects box (see the box on the bottom of Fig. 2). The flow of the business object Order is shown by using its name in the various actions nodes, whereas the flow of Invoice has been emphasized by using an object node. The class diagram in Fig. 2 introduces the classes typing the participants and the objects with their relevant operations and attributes, together with their mutual relationships. For example we can see that a Payment and an Invoice are relative to exactly one Order. The dashed arrows, i.e., the dependency relationships denote that the Company may work on the payments, the invoices and the orders, whereas the Client only on the Payment and the Order. Constraints may be used to finely describe the various classes; for example the constraint on the operation receives of class Company (see the note in Fig. 2) expresses that an Order is considered acceptable by the company if it is well-formed and available.

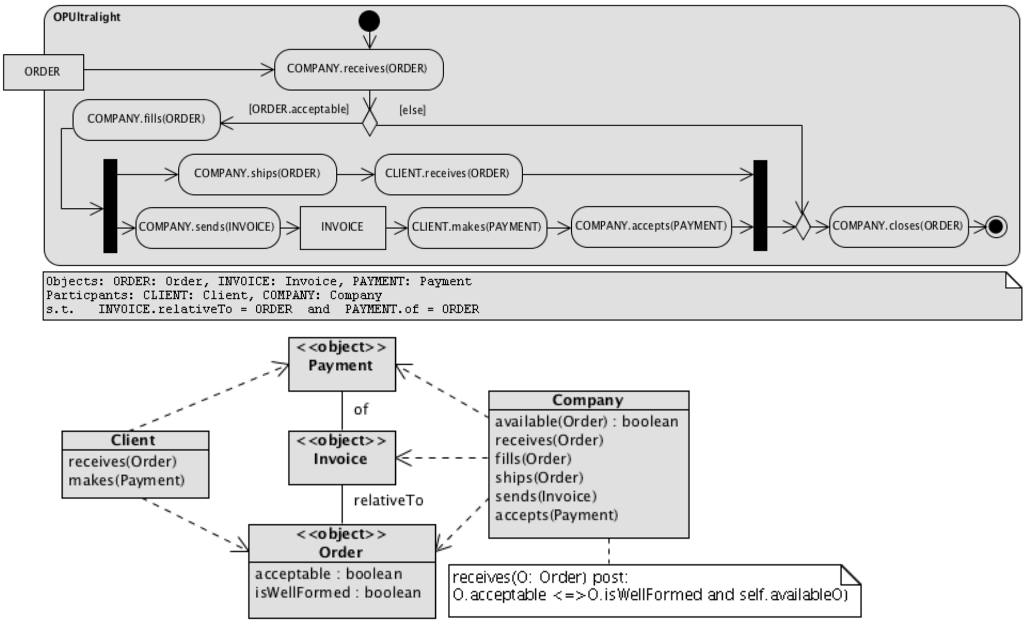

Fig. 2. Precise model of Process Order (activity and class diagram)

3 Experimentation Setup

In this section we present the design of the experiment. We followed the guidelines proposed in [13, 24]. For replication purposes, the experimental package (in English) and the raw data are made available on the Web[3].

[3] www.scienzemfn.unisa.it/scanniello/BPM/ (please cut and past this URL into your web browser)

3.1 Context

The main experiment was conducted with 26 master students in Computer Science at the Free University of Bolzano-Bozen. In addition, 14 bachelor students from the same University took part in a pilot experiment, useful to assess the material, which was executed before the main experiment.

The main experiment represented an optional educational activity of two Software Engineering courses: Infrastructures for Open Service Oriented Architectures and Requirements and Design of Software Systems. The pilot experiment was an optional activity for the Business Information Systems course, attended by third-year bachelor students. As mandatory laboratory activity of the course Infrastructures for Open Service Oriented Architectures, students had previously developed Web services using specification documents that included UML models in terms of class, sequence, and activity diagrams. Students of the course Requirements and Design of Software Systems had already made use of full UML specifications in the design of non-trivial software systems. This year the assignments were defined in cooperation with the Italian National Transplant Organization (CNT): students redesigned the CNT's IT infrastructure according to the SOA paradigm. Note that all the students (master and bachelor) had also previously attended courses on basic and advanced object oriented programming, before carrying out the main experiment and the pilot.

The specifications of two different business processes were considered for the experiment. The business processes refer to application domains in which the subjects are familiar with. The first (i.e., Process Order) is in charge of processing orders for an on-line shop. In particular, this process takes as input an order. The order is accepted and info is filled, payment processing and shipment are done. Finally, the order is closed (see Fig. 1). Instead, the second business process (i.e., Document Management Process) is a business process for managing the on-line review process of any kind of documents. First a document is created by the author and then it is reviewed by a reviewer. Finally, a document is approved (if its quality satisfies the constraints imposed). Note that documents can also be updated and archived. The two business processes are comparable both in complexity and in size as well as in the number of activities and classes. Furthermore, they are small enough to fit the time constraints of the experiment but at the same time they are realistic for small/medium sized comprehension tasks.

3.2 Hypotheses Formulation

The perspective of this study is twofold. From the point of view of researchers, it is an investigation of the effectiveness of using precise activity diagrams in the specification of business processes; and from the point of view of project managers, it is an evaluation of the possibility of adopting this style. Accordingly, we have defined and tested the following null hypotheses:

- H_{lo}: The use of precise activity diagrams **does not significantly improve** the comprehension level of the subjects to perform a task.

– H_{to}: There **is no significant difference** in terms of effort when using precise or ultra-light activity diagrams to perform a comprehension task.

The objective of the statistical analysis is to reject the defined null hypotheses, thus accepting the corresponding alternative ones that admit a positive effect and so can be easily derived. It is worth mentioning that the null hypothesis H_{l0} is one tailed since we expect a positive effect of the precise activity diagrams on the subject performance. On the other hand, H_{t0} is two-tailed since we cannot postulate an expectation of a difference in terms of effort.

3.3 Design

We adopted a counterbalanced design [24] as shown in Table 1. We considered four groups: A, B, C, and D. Each group was formed by subjects randomly selected (precisely: 7 subjects for groups A and D; 6 for groups B and C). Each subject worked on two comprehension *Tasks* (i.e., Task 1 and Task 2) on the following two experimental *Objects*: Process Order (PO) and Document Management Process (DMP). Each time subjects used the precise or ultra-light activity diagrams. For example, the subjects within the group A started to work in Task 1 on PO using the precise activity diagram and then they used the ultra-light activity diagram to perform Task 2 on DMP.

	A	**B**	**C**	**D**
Task 1	PO Precise	PO Ultra-light	DMP Precise	DMP Ultra-light
Task 2	DMP Ultra-light	DMP Precise	PO Ultra-light	PO Precise

Table 1. Experiment design

3.4 Selected Variables

The *control group* indicates the students working with the ultra-light activity diagram, while the *treatment group* indicates the students working with the precise activity diagram. Thus, the only independent variable is *Treatment*, which is a nominal variable that admits two possible values: *Precise* and *Ultra-light*. On the other hand, we selected the following dependent variables to investigate the defined null hypotheses: *comprehension level* and *comprehension effort*.

The *comprehension level* dependent variable is used to measure the comprehension of the subjects on each business process. Similarly to [21], the subjects were asked to answer a comprehension questionnaire (it is equal for both the treatments but different by objects) composed of multiple choice questions. In particular, on each considered business process the questions were 12, each admitting the same number of possible answers (i.e., 5), with one or more correct answers. Fig. 3 shows a sample question (question 1) regarding the comprehension questionnaire of the PO object. The goal of this question is to investigate whether the subjects understood who are the business process participants. Correct answers are the first and the third ones.

1. Indicate the **participant/s**[1] of the workflow

- ☐ Company
- ☐ Invoice
- ☐ Client
- ☐ There are no participants in the workflow
- ☐ Order

Fig. 3. Question 1 of the comprehension questionnaire for PO

Correctness and completeness of the provided answers have been measured, similarly to [21], using an information retrieval based approach. To this end, we defined as: $A_{s,i}$, the set of answers provided by the subject s on the question i, and C_i, the correct set of answers of the question i. The correctness and the completeness of the answers have been measured using, respectively, *precision* and *recall*:

$$precision_{s,i} = \frac{|A_{s,i} \cap C_i|}{|A_{s,i}|} \qquad recall_{s,i} = \frac{|A_{s,i} \cap C_i|}{|C_i|}$$

In order to get a single value representing a balance between correctness and completeness, we used the harmonic mean between precision and recall:

$$F{-}Measure_{s,i} = \frac{2 \cdot precision_{s,i} \cdot recall_{s,i}}{precision_{s,i} + recall_{s,i}}$$

For example, if a student had answered Question 1 of the PO task (Fig. 3) picking the first, second and fifth answer, her precision will be 0.33 (three answers given and only one correct) while her recall will be 0.5 (one correct answer out of two). Instead, her F-measure will be 0.39.

The overall comprehension level achieved by each subject has been computed using the overall average of the F-Measure values of all the questions. This average assumes a value ranging from 0 to 1. A value close to 1 indicates a very good understanding of the business process, while a value close to 0 indicates a very bad comprehension level.

The *comprehension effort* dependent variable measures the time, expressed in minutes, that each subject spent to accomplish a task. We got this value using the start and stop times the subjects were asked to record.

3.5 Experimental Material, Pilot and Execution

In order to assess the experimental material (mainly the comprehension questionnaire) and get an estimation of the time needed to accomplish the task a pilot experiment with 14 bachelor students was accomplished before the main experiment.

Regarding the main experiment, the subjects were asked to use the following procedure to execute both the tasks: *(i)* specify name and start-time in the comprehension questionnaire; *(ii)* answer the questions by consulting the provided material; *(iii)* mark the end-time.

To perform the experiment the subjects were provided with the following hard copy material: *(i)* a summary of the modelled business process, *(ii)* the comprehension questionnaires and the models of the business processes, *(iii)* a unique post-experiment questionnaire to be filled in after the two tasks.

The post-experiment questionnaire aimed at gaining insights about the subjects' behaviour during the experiment. The post-experiment questionnaire was composed of 5 questions concerning the availability of sufficient time to complete the tasks, the clarity of the experimental material and objects, and the ability of subjects to understand the business processes used in the experimentation. The questions expected answers according to the following five point Likert scale: (1) strongly agree, (2) agree, (3) neutral, (4) disagree, (5) strongly disagree. For space reasons, results of the post-experiment questionnaire are only marginally discussed in the following.

4 Analysis and Results

In this section, after a brief summary of the pilot experiment, results of the data analysis of the main study are presented with respect to the defined null hypotheses (Sect. 3.2). In all the performed statistical tests, we decided (as it is customary) to accept a probability of 5% of committing Type-I-error [24], i.e., rejecting the null hypothesis when it is actually true. We conclude the section discussing the effect of the co-factors (i.e., Object, Task and Group) and sketching the potential threats to validity.

4.1 Pilot experiment

All students completed both comprehension tasks of the pilot experiment in 50 minutes. This let us conclude that the pilot was well suited for bachelor/master students. Minor changes were made to improve the comprehension questionnaires. A simplified data analysis showed that students with precise activity diagrams outperformed (the mean comprehension level was 0.69) in comprehension students with ultra-light ones (the mean comprehension level was 0.59). Concerning the effort, students with precise diagrams employed more or less the same time that students with ultra-light diagrams (median for both groups was 19 minutes).

4.2 Comprehension level - main experiment

Table 2 reports some descriptive statistics (i.e., mean, median, and standard deviation) of comprehension level and the results of statistical analysis conducted on the data of the main experiment. Because of the sample size (26 subjects) and mostly non-normality of the data we adopted non-parametric tests to test the first null hypothesis. In particular, we selected Mann-Whitney test for unpaired analysis and Wilcoxon test for paired analysis. We used these tests since they are very robust and sensitive [13, 24].

The overall comparison (i.e., without partitioning by object) is visually presented in Fig. 4 by means of boxplots. From them, it appears that students with precise activity diagrams outperformed in comprehension students with ultra-light ones. We evaluate the first hypothesis overall. The one-way unpaired Mann-Whitney test ($p - value < 0.001$) and the one-way paired Wilcoxon test ($p - value < 0.001$) provide evidence that there exists a significant difference in terms of comprehension level between the two treatments. Therefore, we can reject the null hypothesis H_{l0}. The mean comprehension level improvement achieved with precise diagrams is of 17 points (see means of the "All" row in Table 2), i.e., 27,41%[4]. Similar results can be observed for the primary measures. For space reasons, we report only results of Mann-Whitney tests: precision ($p - value < 0.001$) and recall ($p - value < 0.001$) and for both objects, DMP ($p - value = 0.005$) and PO ($p - value = 0.003$).

	Precise			Ultra-Light				
Object	mean	med	σ	mean	med	σ	MW test	Wilcoxon test
All	0.79	0.84	0.11	0.62	0.66	0.14	< 0.001	< 0.001
DMP	0.76	0.74	0.10	0.64	0.64	0.10	0.005	-
PO	0.80	0.84	0.11	0.58	0.69	0.19	0.003	-

Table 2. Descriptive statistics of comprehension level and the statistical test results

4.3 Comprehension effort - main experiment

Fig. 5 shows the boxplots of comprehension effort versus the treatments. Apparently, students with precise diagrams employed more time that students with ultra-light diagrams (see the two medians). Means and medians are respectively: 22'16" and 20 minutes for precise diagrams; 22'12" and 19'50" minutes for ultra-light diagrams. A two-tailed unpaired Mann-Whitney test returned 0.9 as $p-value$. A similar value is returned by paired Wilcoxon test ($p-value = 0.6$). Therefore, we cannot reject the overall null hypothesis H_{t0}. Even analysing the two objects separately no significant difference was found. The results of the unpaired two-tailed Mann-Whitney test were 0.56 and 0.57 for DMP and PO, respectively.

4.4 Co-factors and Post Questionnaire Results - main experiment

We have analysed the effects of the co-factors on both comprehension level and effort, to find possible interactions with the treatments. This kind of analysis can

[4] The percentage comes from the following equation: 0.62+0.62*x%=0.79

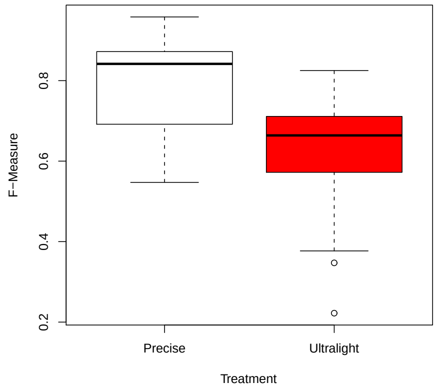

Fig. 4. Comprehension level

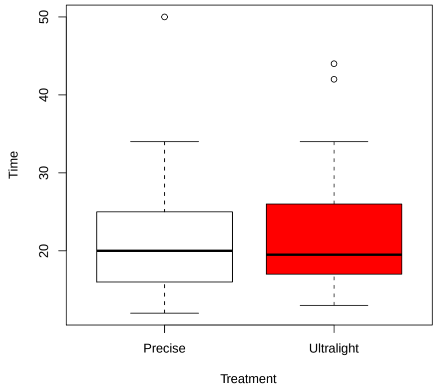

Fig. 5. Comprehension effort

be also useful to discover possible learning, fatigue, and order effects that could "contaminate" the obtained results. For this task, we used a two-way Analysis of Variance (ANOVA) [13, 24]. Even if all the assumptions/conditions for using ANOVA were not checked, this test is quite robust. On the overall data set, we found no significant effect of object on comprehension level ($p-value = 0.77$) and no interaction with treatment ($p-value = 0.22$). Similarly, no significant results were obtained analysing the effects (and interactions) of the co-factors task and group on the comprehension level. Alike, we found no significant effect of objects on effort ($p-value = 0.78$) and no interaction with treatment ($p-value = 0.44$). Instead, an interaction ($p-value = 0.003$) was observed between treatment and task (without effect of task alone). Finally, we found a significant effect of group on effort ($p-value = 0.01$) but no interaction with treatment ($p-value = 0.79$). Looking more in details at the data, we found that students in group A used more time and put more effort than students in other groups. The cause of this difference will be investigated in future.

We computed medians of subjects perception, collected through the perceived agreeing level of the post-experiment questionnaire. Students judged sufficient the time to complete the task , they also found clear: the objectives of the experiment (median=2), comprehension questions (median=2) and answers given as possible options (median=2). Finally they found the exercise useful (median=2).

4.5 Threats to validity - main experiment

Threats to validity that could affect our results belong to four categories [24]: *internal*, *external*, *construct*, and *conclusion*.

The counterbalanced design adopted in this experiment enabled us to mitigate *internal validity* threats. It is well-known that this design balances possible learning, fatigue, and order effects. This was also confirmed by the analysis of the co-factors (see Sect. 4.4). Another issue concerns the information exchanged among the subjects. This was prevented as much as possible by monitoring the students while performing the tasks. In addition, students were not evaluated on their performance to avoid apprehension.

External validity may be threatened when experiments are performed with students, throwing into doubt the representativeness of the subjects with respect to software professionals. However, performed tasks do not require high level of industrial experience, so we believe that this experiment can be considered appropriate, as suggested in the literature [3]. Another possible threat concerns the size and complexity of the tasks. Hence, we plan to replicate the experiment with more complex tasks. Replications with different and more experienced subjects (e.g., professionals and PhD. Students) are planned as well.

In this study, the *construct validity* threats are related to the metrics used to get a quantitative evaluation of the subjects' comprehension and effort. We used questionnaires to assess the comprehension level of the subjects, and answers were evaluated using an information retrieval based approach (as in [21]), in order to avoid as much as possible any subjective evaluation. Furthermore, the comprehension questionnaires were defined to be complex enough without being too obvious. The comprehension effort was measured by means of proper time sheets, and it was validated qualitatively by researchers, who were present during the experiment.

Conclusion validity concerns data collection, reliability of measurements, and validity of statistical tests. Statistical non-parametric tests (Mann-Whitney and Wilcoxon) were used to reject the null hypotheses. Two-way ANOVA was used to detect possible co-factors effects and interactions between each co-factor and the main factor. Even if all the assumptions/conditions for using ANOVA were not checked, this test is quite robust and has been extensively used in the past to conduct analyses similar to ours (see, e.g., [23]).

5 Related Work

Out of the huge body of literature that is based on UML, we highlight in this section just a few papers, that provide useful terms of comparison for our work.

The UML activity diagrams provide an intuitive and easy to learn visual formalism to model business process [7, 16, 12]. For example, Di Nitto *et al.* [16] propose an approach to process modelling by using a subset of UML diagrams, including UML activity diagrams with object flow to model the control and data flow, class diagrams to model structural properties of the process, and state diagrams to model the behaviour of activities. The XMI standard representation of these models produced using a UML CASE tool can then be translated into an executable process description for the OPSS Workflow Management System [5]. Several are the differences between our approach and theirs. The most remarkable ones are that OCL is not used in the process modelling and the validity of the proposal has not been assessed through controlled experiments.

In [7] a case study is presented, mapping UML activity diagrams with object flow on the process definition language of the GENESIS environment [1]. The authors showed that UML activity diagrams do not support all the control flow and data flow rules of the GENESIS process definition language. As a consequence, the syntax and semantics of this type of UML diagrams often need to be extended to make them suitable for modelling business processes in workflow management systems. Similarly, De Lucia *et al.* in [6] present a system offering a visual environment, based on an extension of UML activity diagrams, that allows to graphically design a process model, and to visually monitor its enactment. The main difference with respect to notations used in this experiment is that participants and objects are not explicitly considered. Furthermore, the behavioural conditions are not formally specified.

Differently from us, all the approaches discussed above do not assess the validity of the proposed formalism by means of controlled experiments. To our knowledge, only a few other studies perform empirical evaluations in business process formalisms comparisons. For instance, Peixoto *et al.*, [20] compare UML and BPMN (Business Process Modelling Notation) [17], with respect to their readability in expressing Business Processes. Their analysis is motivated by the consideration that many different stakeholders are interested in the results of a business process modelling. Given their different background, and their need to understand the results of modelling, it is very important that all stakeholders are able to understand business process diagrams. Peixoto *et al.* expected BPMN models to be easier to understand than UML 2.0 activity diagrams, as BPMN is a specialized language, designed for modelling business process and with the primary goal of being understandable by all business stakeholders [17]. However, an experiment with 35 undergraduate students of Computer Science, unskilled in business process modelling, could not confirm their initial hypothesis, therefore UML activity diagrams and BPMN seem to be equivalent in terms of understandability.

Gross and Doerr [10] conducted two experiments, comparing the UML activity diagrams and Event-driven Process Chains [22] with different perspectives. First, they considered the business processes specification from a requirements engineer perspective with a focus on model creation. Second, their attention was on model understandability, seen from a customer's or end user's point of view. The used methodology was in both cases a blocked subject-object study: participants were partitioned in two groups, each of them receiving an assignment in one of the considered formalisms. The authors found evidence that activity diagrams performed better than EPCs from a requirements engineer's perspective. When considering end users, no significant difference was identified between the two methods.

With different objectives, Coman and Sillitti [4] conducted an empirical study on the possibility of mapping low-level to high-level software development activities in an automatic way. The method is based on low level data automatically collected, that are used in order to identify high-level activities. The context of the experiment is similar to ours, as, among other commonalities, the analysed data was related to software systems similar to the one proposed in this work.

6 Conclusion

In this paper, we propose a variant of UML activity diagrams. This variant has been defined in the context of business processes modeling and its effectiveness has been investigated with respect to a less rigorous visual formalism. To this end, a controlled experiment with 26 master students has been conducted and the results have been presented and discussed in this paper. The data analysis indicated a significant effect of the more rigorous style on the comprehension of business processes (+27.61%). Conversely, the effect of the effort to accomplish comprehension tasks is not statistically significant.

Acknowledgements: We would like to thank the students that took part in both the pilot and the main experiment.

References

1. L. Aversano, A. De Lucia, M. Gaeta, P. Ritrovato, S. Stefanucci, and M. L. Villani. Managing coordination and cooperation in distributed software processes: the genesis environment. *Software Process: Improvement and Practice*, 9(4):239–263, 2004.
2. S. Bandinelli, E. Di Nitto, and A. Fuggetta. Supporting cooperation in the spade-1 environment. *IEEE Trans. Softw. Eng.*, 22:841–865, December 1996.
3. V. Basili, F. Shull, and F. Lanubile. Building knowledge through families of experiments. *IEEE Trans. Softw. Eng.*, 25(4):456–473, 1999.
4. I. D. Coman and A. Sillitti. An empirical exporatory study on inferring developpers' activities from low-level data. In *Proceedings of International Conference on Software Engineering and Knowledge*, pages 15–18. Knowledge Systems Institute Graduate School, 2007.
5. G. Cugola, E. Di Nitto, and A. Fuggetta. The jedi event-based infrastructure and its application to the development of the opss wfms. *IEEE Trans. Softw. Eng.*, 27:827–850, September 2001.
6. A. De Lucia, R. Francese, G. Scanniello, and G. Tortora. Distributed workflow management based on UML and web services. In *Encyclopedia of E-Commerce, E-Government, and Mobile Commerce*, pages 217–222. IGI Global, 2006.
7. A. De Lucia, R. Francese, and G. Tortora. Deriving workflow enactment rules from uml activity diagrams: a case study. *Symposium on Human-Centric Computing Languages and Environments*, 0:211–218, 2003.
8. H. E. Eriksson and M. Penker. *Business Modelling with UML*. Wiley Computing Publishing, 2000.
9. J. D. Gonzalez and J. S. Diaz. Business process-driven requirements engineering: a goal-based approach. In *Proc. of the 8th Workshop on Business Process Modeling Development and Support*, pages 1–9. Tapir Academic Press, 2007.

10. A. Gross and J. Doerr. EPC vs. UML activity diagram - two experiments examining their usefulness for requirements engineering. In *Proceedings of Requirements Engineering Conference*, pages 47–56, Washington, DC, USA, 2009. IEEE Computer Society.
11. P. Heimann, G. Joeris, C. Krapp, and B. Westfechtel. Dynamite: Dynamic task nets for software process management. In *Proceedings of the International Conference on Software Engineering*, pages 331–341, 1996.
12. S. Jurack, L. Lambers, K. Mehner, G. Taentzer, and G. Wierse. Object flow definition for refined activity diagrams. In *Proceedings of the 12th International Conference on Fundamental Approaches to Software Engineering*, pages 49–63, Berlin, Heidelberg, 2009. Springer-Verlag.
13. N. Juristo and A. Moreno. *Basics of Software Engineering Experimentation.* Kluwer Academic Publishers, 2001.
14. C. Marshall. *Enterprise modelling with UML: Designing successful software through business analysis.* Addison-Wesley, 2000.
15. R. Monfared, A. West, R. Harrison, and R. Weston. An implementation of the business process modelling approach in the automotive industry. *Journal of Engineering Manufacture*, 216(11):1413–1428, 2002.
16. E. D. Nitto, L. Lavazza, M. Schiavoni, E. Tracanella, and M. Trombetta. Deriving executable process descriptions from UML. In *Proceedings of the 22rd International Conference on Software Engineering*, pages 155–165, 2002.
17. OMG. Business process model and notation (BPMN) Version 2.0. *OMG Final Adopted Specification, Object Management Group*, 2006.
18. OMG. Object constraint language (OCL) specification, version 2.2. Technical report, Object Management Group, February 2010.
19. OMG. Unified modeling language (UML) specification, version 2.3. Technical report, Object Management Group, May 2010.
20. D. Peixoto, V. Batista, A. Atayde, E. Borges, R. Resende, and C. Pádua. A Comparison of BPMN and UML 2.0 Activity Diagrams. In *VII Simposio Brasileiro de Qualidade de Software*, 2008.
21. F. Ricca, M. Di Penta, M. Torchiano, P. Tonella, and M. Ceccato. The role of experience and ability in comprehension tasks supported by uml stereotypes. In *29th International Conference on Software Engineering (ICSE 2007), Minneapolis, MN, USA, May 20-26, 2007*, pages 375–384, 2007.
22. A. Scheer. *ARIS-business process modeling.* Springer Verlag, 2000.
23. M. Torchiano, F. Ricca, and P. Tonella. Empirical comparison of graphical and annotation-based re-documentation approaches. *IET Software*, 4(1):15–31, 2010.
24. C. Wohlin, P. Runeson, M. Höst, M. Ohlsson, B. Regnell, and A. Wesslén. *Experimentation in Software Engineering - An Introduction.* Kluwer, 2000.