Assessing the effectiveness of "Precise" Activity Diagrams in the Context of Business Process Modeling

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 modeling business processes. In this paper, we present a novel precise style for this notation and a controlled experiment to assess its effectiveness. The context of the experiment is constituted of master students in Computer Science at the Free University of Bolzano-Bozen in Italy. The results indicate that the subjects achieved a significantly better comprehension level when business processes are modeled using the precise style with respect to a "lighter" variant, with no significant impact on the effort to accomplish the comprehension tasks.

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

1 Introduction

To be competitive in the global market, many organizations have been changing their business processes [4]. In this context, modeling, management, and enactment of business processes are considered relevant to support organizations in their daily activities. Regarding the modeling of business processes, a number of process definition languages — such as BPMN [9], event-condition-action mechanisms [1], graph rewriting mechanism [5] and Petri Nets [2] — have been proposed in the literature.

More recently, the use of the Unified Modeling Language (UML) [11] has been suggested in the context of business process modeling. This trend is mainly due to the fact that UML has been conceived for the communication among people and so it may represent a natural choice for such a kind of modeling [8]. Further, in favor of the UML notation there is its flexibility that allows choosing the preferred degree of precision/abstractiveness to express models. For example, different options are available ranging from "light" styles, where nodes and arcs of the activity diagrams are decorated by natural language text, to more rigorous ones, where nodes and arcs might be expressed in a formal language. "Light" activity diagrams are simple to write/use but their inherent ambiguity could complicate the communication among participants. On the other hand, more precise/rigorous notations are more complex to use but limit ambiguity and they can be transformed into executable models (e.g., expressed in BPEL) more easily.

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] together with other variants: *ultra-light*, *light*, and *conceptual precise*. In order to compare the proposed precise style with a lighter variant (precisely the *ultra-light* one) a controlled experiment with master students in Computer Science has been conducted and the results have been presented in the paper.

2 Process modeling with UML

2.1 Basic Terminology

The basic activities in a business process are the *basic task* of the process, while the *process objects* are the entities over which an activity of the process is performed. The active entities that perform the tasks are *process participants*, and whenever it will be relevant, we will distinguish the human participants from those corresponding to software and hardware systems.

2.2 Ultra-light style

The *ultra-light* style is the one often used in the industry for business process modeling, see e.g., [7]. Following the *ultra-light style* a process is modeled by a UML activity diagram, where nodes (activity and object) and guards on the arcs leaving the decision nodes are decorated by natural language text and usually they follow neither rules nor patterns.

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

In Fig. 1, we present the *ultra-light* UML model of the business process Process Order, namely one of the experimental objects used in the empirical investigation presented here. 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 produced process objects are shown as rectangles. The activity diagram describes how the order is managed by the on-line shop. It is quite easy to understand and there is no need to further comment it.

[1] the TECDOC project, funded in the framework of research activities of Ligurian Technology District SIIT, aimed to define methodologies to efficiently schedule, coordinate, monitor and manage Complex Organizations.

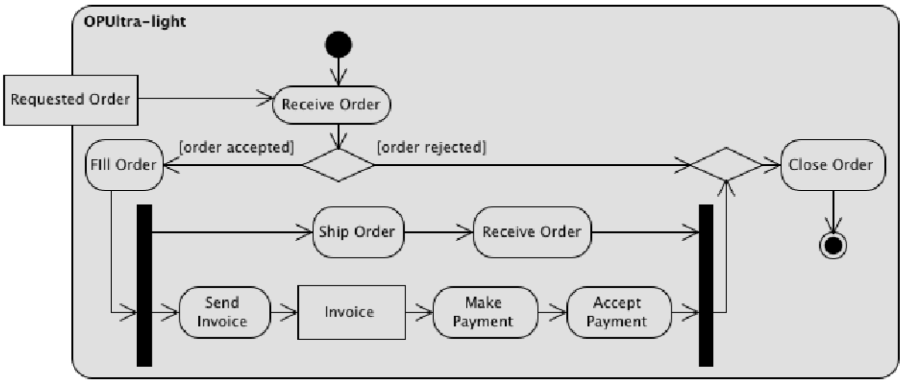

Fig. 1. Ultra-light model of Process Order

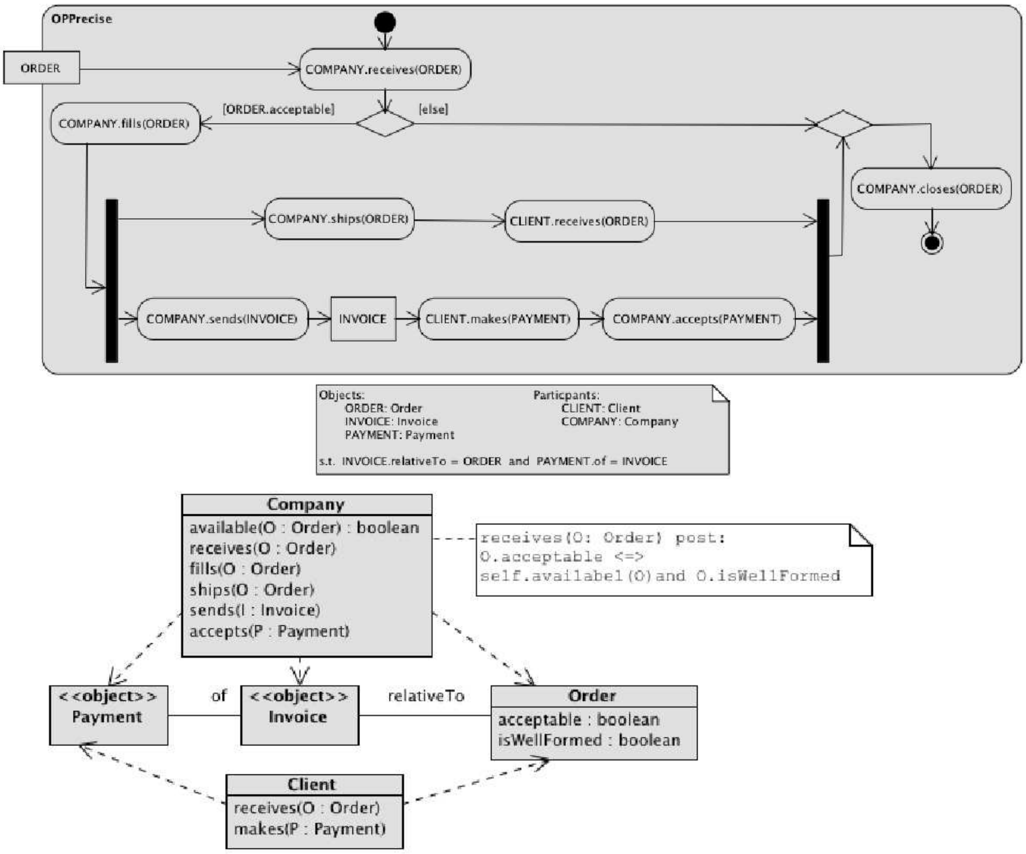

Fig. 2. Precise model of Process Order (activity and class diagram)

2.3 Precise Style

The participants and the objects of a business process modeled by the precise style are explicitly listed and precisely modeled with UML by means of classes. This represents the most remarkable difference with respect to the ultra-light style. Conversely, the behavioral 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 [10] (the textual language for boolean expressions part of 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.

Fig. 2 shows the precise model of the Process Order case. In the process we have two participants (both of kind human beings) 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 action 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 may manage only 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 in OCL that an order is considered acceptable by the company if and only if it is well-formed and available.

3 Experimentation Setup

In this section we highlight the design of the experiment following the guidelines proposed by Wohlin *et al.* in [14]. For replication purposes, the experimental package (in English) and the raw data are made available on the Web[3].

3.1 Context and Hypotheses Formulation

The experiment was conducted with 26 master students in Computer Science at the Free University of Bolzano-Bozen. This experiment represented an optional educational activity of two Software Engineering courses: Infrastructures for Open Service Oriented Architectures and Requirements and Design of Software Systems. As mandatory laboratory activity of the former course, the students had previously individually developed Web services using specification documents that included UML models in terms of class, sequence, and activity

[3] www.scienzemfn.unisa.it/scanniello/BPM/

diagrams. Students of the course Requirements and Design of Software Systems had already made use of UML in the design of a non-trivial software system.

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 null hypotheses, thus accepting the corresponding alternative ones that can be easily derived. H_{l0} is one tailed [14] since we expect a positive effect of the precise style on the process comprehension, while H_{t0} is two-tailed since we cannot postulate an expectation.

3.2 Design

We adopted a counterbalanced design [14] 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 the ultra-light activity diagrams.

The selected business processes refer to application domains on which the subjects were familiar with. PO is for processing orders in an on-line shop (see Fig. 1). Conversely, DMP is in charge of managing the review process of any kind of document (e.g., recipes for culinary dishes).

3.3 Selected Variables

In this experiment, the only independent variable is *Treatment*, which is a nominal and admits two values: *Precise* and *Ultra-light*. Further, we selected the following dependent variables to test the 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 model. To this end, we asked the subjects to answer a comprehension questionnaire (one for each experimental object) composed of multiple choice questions. Fig. 3 shows a sample question (Question 8) concerning the comprehension questionnaire of PO.

The provided answers have been measured using an information retrieval based approach. In particular, the correctness and the completeness of the answers have been measured, similarly to [13], using *precision* and *recall*, respectively. In order to get a single value representing a balance between correctness and completeness, we used the F-measure, i.e., the harmonic mean of precision

	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

Indicate the case/s in which a **company** does not accept an **order**

- □ The order is not well-formed (e.g., the items quantity is not indicated in the order)
- □ The workers of the company hold a strike
- □ The client does not own a credit card
- □ The items requested in the order are not available
- □ The client doesn't want pay

Fig. 3. A sample question of the comprehension questionnaire for PO

and recall values. For example, if a student had answered to Question 8 of the PO task (Fig. 3) picking the first, second and fifth answer, where the correct answers are only the first and the fourth ones, the precision value will be 0.33 (three answers given and only one correct) while the recall value will be 0.5 (one correct answer out of two). The F-measure value will be then 0.39.

The overall comprehension level achieved by each subject, which assumes value ranging from 0 to 1, has been computed using the overall average of the F-Measure values of all the questions. 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* variable measures the time, expressed in minutes, that each subject spends to accomplish a task. We got the comprehension effort values using the start and stop times the subjects were asked to record.

3.4 Execution and Experimental Material

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

To perform the experiment the subjects were provided with the following hard copy material: *(i)* a summary of the modeled business process, *(ii)* the comprehension questionnaires and the models of the business processes, *(iii)* a unique post-experiment questionnaire[2] to be filled in after the two tasks.

4 Analysis and Results

In this section, the results of the data analysis are sketched quickly with respect to the defined null hypotheses. In all the performed statistical tests, we decided

[2] For space reasons, the results of the post-experiment questionnaire are not presented.

	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 statistical test results.

(as it is customary) to accept a probability of 5% of committing Type-I-error [14].

4.1 Comprehension level

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 gathered data. 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 [14].

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, in general, 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%. 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 ($p - value = 0.003$).

4.2 Comprehension effort

Students with precise diagrams employed more time that students with ultra-light diagrams. 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 analyzing 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.

5 Conclusion

UML activity diagrams provide an intuitive and easy way to express business processes models [3], [8], [6]. However, their effectiveness is mostly not assessed by means of controlled experiments. Indeed, only a few other studies perform empirical evaluations in business process formalisms comparisons. An example is [12], where a comparison between UML and BPMN is presented.

In this paper, we have proposed a precise style for the UML activity diagrams in the context of business process modeling. The effectiveness of this style has been investigated with respect to a less rigorous style by using a controlled experiment. The results of this investigation indicated a significant effect of the precise style on the comprehension (+27.61%), whith no impact on the effort.

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. 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.
4. H. E. Eriksson and M. Penker. *Business Modelling with UML*. Wiley Computing Publishing, 2000.
5. 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.
6. 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.
7. 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.
8. 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.
9. OMG. Business process model and notation (BPMN) Version 2.0. *OMG Final Adopted Specification, Object Management Group*, 2006.
10. OMG. Object constraint language (OCL) specification, version 2.2. Technical report, Object Management Group, February 2010.
11. OMG. Unified modeling language (OMG UML) specification, version 2.3. Technical report, Object Management Group, May 2010.
12. 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.
13. 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.
14. C. Wohlin, P. Runeson, M. Höst, M. Ohlsson, B. Regnell, and A. Wesslén. *Experimentation in Software Engineering - An Introduction*. Kluwer, 2000.