UML Acceptance: Analyzing the Students' Perception of UML Diagrams

Adriana Lopes
Instituto de Computação - IComp
Universidade Federal do Amazonas
Manaus-AM/Brasil
adriana@icomp.ufam.edu.br

Igor Steinmacher
School of Informatics, Computing, and Cyber Systems
Northern Arizona University
Flagstaff, AZ, USA
igor.steinmacher@nau.edu

Tayana Conte
Instituto de Computação - IComp
Universidade Federal do Amazonas
Manaus-AM/Brasil
tayana@icomp.ufam.edu.br

ABSTRACT

Unified Modeling Language (UML) has been adopted as a standard modeling language in the software industry for the graphical representation of analysis and design models. Due to its importance, UML is taught in most undergraduate majors in Software Engineering and Computing. However, little is known about the students' perception regarding the UML diagrams. Their perceptions may influence the diagrams' adoption since the students are future software engineers. This paper aims to start filling this gap by better understanding students' perception regarding UML diagrams acceptance. To achieve this goal, we applied a Focus Group to foster a discussion with undergraduate students about five UML diagrams. Besides, we applied the Technology Acceptance Model to assess the students' acceptance regarding these diagrams. Based on the results of this study, students considered use cases and class diagrams useful for designing systems, although they stated that their modeling was not easy. We also noticed that the perceived usefulness was the construct that most influenced the students' intention to use these diagrams. Our research benefits instructors interested in understanding the students' perceptions about UML, enlightening the points that they need to reinforce to show the benefits of the modeling language. Still, the instructors are invited to reflect on other teaching strategies to show the importance of these diagrams in the software development process.

CCS CONCEPTS

• Software and its engineering • Software notations and tools • System description languages • Design languages • Unified Modeling Language (UML)

SBES 2019, September 23–27, 2019, Salvador, Brazil

ACM ISBN 978-1-4503-7651-8/19/09…$15.00
https://doi.org/10.1145/3350768.3352575

KEYWORDS

Software models, UML diagrams, Technology Acceptance, Focus Group.

ACM Reference format:

Adriana Lopes, Igor Steinmacher and Tayana Conte. 2019. UML Acceptance: Analyzing the Students' Perception of UML Diagrams. In *Proceedings of XXXIII Brazilian Symposium on Software Engineering. XXXIII SBES - Education Track (SBES - Education Track'19), 10 pages.*

1 Introduction

Teaching software modeling can be challenging since students must learn the syntax and representation of the problem domain for modeling and the importance of these models in software development [1]. These models can be expressed following the Unified Modeling Language (UML) [2], a standard for the graphical representation of analysis and design models in the software industry [3]. UML is taught in the majority of undergraduate courses in Software Engineering and Computing. UML has a variety of diagrams which aid software analysis and design; for instance, one can use case diagrams – which represents how actors interact with the system – and class diagrams – which represents the static structure of a system in terms of classes and the relationships between classes [4].

UML has been heralded as the *"Lingua Franca of Software Engineering Education."* Several studies have already investigated the use of UML [5] [6] [7]. Petre [5] interviewed 50 software engineers, and most of them mentioned they did not use UML. Besides, the software engineers who adopt them have different ways of using the diagrams. Thus, we noticed that UML diagrams, despite their importance, are not used in specific contexts, and this raises some questions, for example: which UML diagrams are considered easier to use and useful by undergraduate students after attending UML classes? Why do the students plan to adopt (or not) these diagrams as future software engineers? Is training received by software engineers a potential reason why UML is not so widely adopted in the industry?

To start the effort to answer questions like these, in this paper, we present an exploratory study to investigate the students' perception regarding UML diagrams acceptance. In this study, we investigated five UML diagrams taught for the undergraduate students from the Federal University of Amazonas. These UML diagrams are: class diagrams, use case diagrams (and specifications), sequence diagrams, state machine diagrams, and activity diagrams. This exploratory study was conducted with 36 senior undergraduate students.

Regarding UML acceptance, we applied a questionnaire based on Technology Acceptance Model (TAM) [8] and a Focus Group to collect data on the students' perceptions [9]. TAM has been applied to evaluate the acceptance of a large set of technologies regarding the users' perceived ease of use and perceived usefulness [10]. Perceived ease of use is the degree to which a person believes that using a specific technology will be free of effort. Perceived usefulness is the extent to which a person believes that using a specific technology will enhance his or her job performance. According to TAM, the user's behavioral intention to use a specific technology is determined by perceived usefulness and perceived ease of use. Focus Group is a qualitative technique used to collect data by organizing group interviews to discuss a particular object [9]. Although UML diagrams are not new technologies in Software Engineering, we applied TAM based questionnaires to gather the perspective of the students regarding the acceptance of these diagrams. Focus Group was applied to foster an open discussion about the students' perceptions.

To guide our research, we explored specific research questions. These research questions were defined based on the acceptance of these diagrams, as follows:

RQ1 – Which UML diagrams are considered easier to use and useful by undergraduate students?

RQ2 – Why do the students plan to adopt (or not) these diagrams as future software engineers?

The results from this study indicate that the participants considered the use case diagrams (and their specifications) and class diagrams useful for information understanding during the software development and that they intend to use the use case diagrams and specification, and class diagrams in system design. We also found that participants did not intend to use the sequence, activity, and state machine diagrams in the software development process. The participants stated that other diagrams provide a better understanding of the system. These results make us hypothesize the type of training received by students may be one of the causes of UML diagrams are not adopted in the industry. These results are relevant since it is the first that uncovers how students perceive UML diagrams to help instructors to improve how they teach them, highlighting their importance in the software development process.

The main contribution of our research is to provide a first study on the students' perceptions of UML. Thus, instructors may adopt teaching strategies to show the importance of these diagrams in the software development process. Besides, this paper offers a direction for further studies aiming to evaluate whether one of the causes of UML is not so widely adopted in the industry is related to the training received by software engineers. We also consider that the approach adopted in this study, using Focus Group and TAM to help understand students' perceptions, could be useful tools to support the reflection about software engineering topics. This approach allows instructors to understand students' perceptions of the concepts taught.

2 Teaching and Perceptions of UML Diagrams

Regarding UML teaching, there are reports about the difficulty's students face when modeling, such as: difficulties regarding the syntax semantics, and organization of information in the diagrams [11][12]. To support the teaching of UML diagrams, there are several works [1, 13, 14].

Burgueño et al. [13] proposed activities from a case study modeled with UML and OCL/USE [15] that has been successfully used to teach modeling in class, e.g., students specify a system and its views, verified their relations, and performed several kinds of analyses on the overall system specifications. Lethbridge [14] presented the Umple tool for model-oriented programming and modeling teaching.

To the best of our knowledge, there are no works about the students' perception regarding UML acceptance. It is indeed very relevant that the instructor understands how students perceive the topics that were taught. This may foster the instructors to adopt different strategies to support the teaching of the importance of these diagrams.

On the other hand, the literature presents studies about the adoption of UML in software industry. For example, Petre [5] conducted interviews with 50 software engineers about their use (or not) of UML. She identified five different usage patterns by those who use (or know people who use) UML. This indicates that software engineers use the diagrams for different purposes and needs. Petre [5] also points out "*UML has achieved penetration in Software Engineering Education.*". If the UML is so generally taught, why is it not so widely adopted in the industry? Several studies already investigated the use of UML in the industry. However, it is important to identify how students perceive the UML and whether these perceptions influence in the adoption of the UML in the industry since students are future software engineers.

3 Exploratory Study

Federal University of Amazonas offers a 4-year degree in Computer Science. The first two years are dedicated to the core topics, including the introductory course on Software Engineering (SE). The last two years include an introductory course about Software Engineering and an advanced course about Analysis and Design of Software Systems (ADS). The students are required to attend an introductory ES course before ADS course. The SE course syllabus comprises software development processes, requirements engineering, agile methods, testing, and software maintenance.

ADS is a 90-hour course taught in one semester in the junior year of Computer Science degree, with a 6 hours/week schedule. The main goal of ADS course is to teach students the key software modeling concepts and the use of UML diagrams in software development. Regarding the UML topic, the following classes were taught: use case diagrams (4 classes); class diagrams (4); state machine diagrams (3); activity diagrams (2); and sequence diagrams (4). In addition to three full classes with practical assignments focusing on dealing with all these artifacts, totaling 40 hours.

During the classes, the students had been exposed to different problems with different difficulty levels. After these classes, the instructors split the students into teams of up to five students to work on an assignment (each team had to model the solution for a different problem), using such diagrams. All teams presented their deliverables to the class, receiving feedback from the instructors. We clarify that we conducted our study after the students finished the assignment and received feedback.

We invited Computer Science senior undergraduate students that took both courses to participate in this research, explaining that our goal was to understand their perception of UML 2.0. Regarding the students' experience in programming, the participants had learned both structured programming and object-oriented programming before taking the ADS course.

In order to evaluate the perceptions of senior undergraduate students on UML diagrams and thus help instructors improving the teaching of its importance, we applied TAM based questionnaires and Focus Group (FG). TAM posits that two constructs determine the users' behavioral Intention to Use (IU) a technology: Perceived Usefulness (PU) and Perceived Ease of Use (PE). We developed questionnaires based on the TAM constructs (PU, PE, and IU) for each UML diagram. We present the adapted TAM statements below, in which we changed the word "diagram" to the name of each UML diagram. The TAM based questionnaires, applied for each UML diagram evaluated in this study, are available in our technical report [16].

PERCEIVED EASE OF USE

PE1. My interaction (modeling and comprehension of the information) with this diagram is clear and understandable.

PE2. Interacting (modeling and comprehension of the information) with this diagram does not require a lot of my mental effort.

PE3. I find this diagram to be easy to use (for modeling and comprehension of the information).

PE4. I find it easy to get this diagram to do what I want it to do (in terms of modeling and comprehension of the information).

PERCEIVED USEFULNESS

PU1. Using this diagram improves my performance better for understanding aspects of the software.

PU2. Using this diagram in my job has improved my productivity for understanding aspects of the software.

PU3. Using this diagram enhances my effectiveness in understanding aspects of the software.

PU4. I consider this diagram useful for software design.

INTENTION TO USE

I1. Assuming I had enough time to design software, I intend to use this diagram.

I2. Considering that if I could choose any notation for the analysis and design, I predict that I would use this diagram.

Regarding the adapted TAM statements, participants provided their answers on a seven-point Likert scale. The possible answers were: "Totally Agree, Strongly Agree, Partially Agree, Neutral, Partially Disagree, Strongly Disagree, And Totally Disagree".

FG can be used to foster an open discussion about a research object. França et al. [9] present roles called "Lovers" and "Haters", in which lovers must argue in favor of the object and haters against it. In this study, we selected the FG board to support the discussion based on the TAM indicators for each object of analysis. Figure 1 shows an example of the FG board, split into six dimensions, and classified as:

Lovers – "It is useful because…, It is easy to use because…, I will use it in the future because…"; and

Haters – "It is not useful because..., It is difficult to use because…, I will not use it in the future because…".

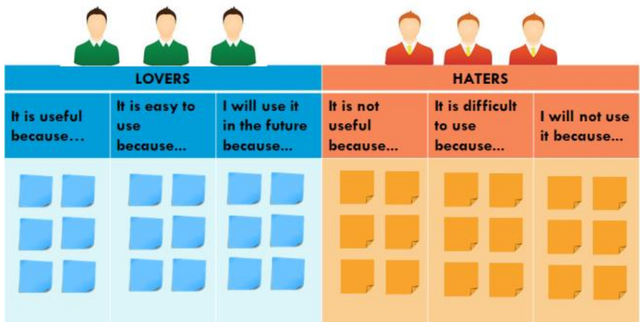

Figure 1: Focus Group board

The participants were asked to highlight their perceptions by placing post-its supporting each argument in the FG board. Also, the moderator promoted a replica section.

Figure 2 presents the planning of the steps for conducting the study (Study Planning and Study Execution). We present the details of these steps in next subsections.

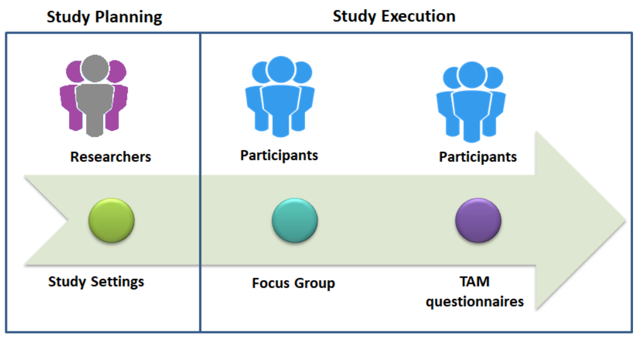

Figure 2: Steps for conducting the study.

3.1. Exploratory Study Planning

We invited 50 Computer Science senior undergraduate students to take part in this research. Thirty-six (36) students accepted to participate in the research. The students had previous knowledge in software modeling with UML diagrams, but had no prior experience with software development projects in the industry. Throughout the text, we identify the students as P# (where # is a number that uniquely identifies each participant).

We randomly assigned the students to groups (with four to five students) to facilitate the discussion in the FG. We also randomly chose the groups that would be lovers or haters for the diagrams. In addition, each group assumed the role of lovers and haters for different diagrams, for example: one group assumed on the role of lovers for the class diagram and the role of haters for the state-machine diagram. In cases when the participants acted as lovers, we asked them what they would report as positive points of that diagram; and the opposite for the haters.

3.2. Exploratory Study Execution

Before the study execution, we conducted a training session with the students. The session lasted approximately 20 minutes. It aimed to help the students remembering the UML diagrams that we analyzed in this study. We conducted the study in a laboratory at Federal University of Amazonas using video cameras to record the FG execution. During the FG, one of the authors of this paper performed the role of moderator. Figure 3 presents an example of students reporting their perceptions with the support of the FG board.

Figure 3: Lovers reporting their insights with the support of the Focus Group board

At the end of the FG, we invited the participants to freely answer to the TAM adapted questionnaires, according to their perceptions. We informed the students that their answers should represent their perceptions regarding each UML diagrams; that they should not assume the previous roles of lovers and haters.

4 Exploratory Study Results

We will present the results of the TAM based questionnaires along with the results from the FG for each UML diagram. We analyzed the reliability to guarantee the internal validity and consistency of the questionnaires.

We applied the Cronbach Alpha (CA) test to assess the questionnaires' reliability [17]. Also, we conducted a factorial analysis to understand whether a group of statements is correlated or not with a factor to PE and PU. The CA test and factor analysis that exceeds a threshold of 0.7 indicate a reliable measure. We applied the test only to the PE and PU constructs since the UI construct is correlated with both. These results show that the statements for PU and PE are reliable (see Table 1).

Table 1 Reliability evaluation of the questionnaires

DIAGRAMAS	PU	PE
Use Case	α = 0.894	α = 0.862
Class	α = 0.891	α = 0.889
Sequence	α = 0.937	α = 0.942
State Machine	α = 0.891	α = 0.889
Activity	α = 0.951	α = 0.878

The results of the factorial analysis show that the statements are correlated (see the highlighted values in Table 2) with results for the most related factors. In this table, PE statements relate to Factor 1 and PU statements to Factor 2. More details about this analysis are available in [18].

4.1. Use Case Diagrams and Specification

Figure 4 presents the results concerning the perceived usefulness (PU1 to PU4), perceived ease of use (PEU1 to PEU4) and intention to use (IU1 and IU2) of the students about use case diagram. In the FG, lovers emphasized that: use cases are useful because of the simplicity of modeling the interaction between the actors and system, and since software engineers understand the scenarios easily, they can be used in the future. Haters reported that: the use case specification is not easy to use because the alternative and exception flows are similar. There was a replica from haters to lovers. Haters prefer other alternatives, such as User Stories.

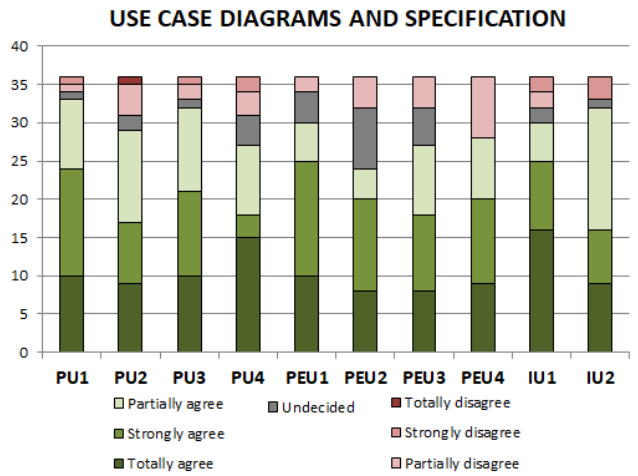

Figure 4: Answers to perceived ease of use, perceived usefulness and intention to use the use case

In the open questions of the questionnaire, P10 reported that use case diagrams make it easier to understand the systems' features and actors: *"I think it easy to understand the information related to features that need to be developed"* (P10). Still, P5 reported that, if the system is complex, the diagram might not be useful due to the effort that will be necessary to read and understand the diagram, as it can be observed in the quote: *"Use case diagrams are easy to understand because you have an overview of the features that you should have in the system. I consider that the difficulty of modeling with this diagram depends on the complexity of the system"* (P5).

Regarding the use case specification, we observed that P4 and P8 explained that the use case specification is useful to complement the use case diagrams. These observations are evidenced in the following quotes: *"The use case specification is useful and complements the diagram"* (P4) and *"The use case specification is useful. However, it is complicated to write because we have to think about several flows"* (P8).

By observing these results, we conclude that the participants considered the use case diagrams easy and useful for software development. However, students argue that the complexity of the system may not be easy for modeling and understanding this diagram. This may affect its future use.

4.2. Class Diagrams

Figure 5 presents the participants' answers to the perceived usefulness (PUC1 to PUC4), perceived ease of use (PEC1 to PEC4) and intention to use (IUC1 and IUC2). As it can be observed, there are some disagreements regarding PU and PEU.

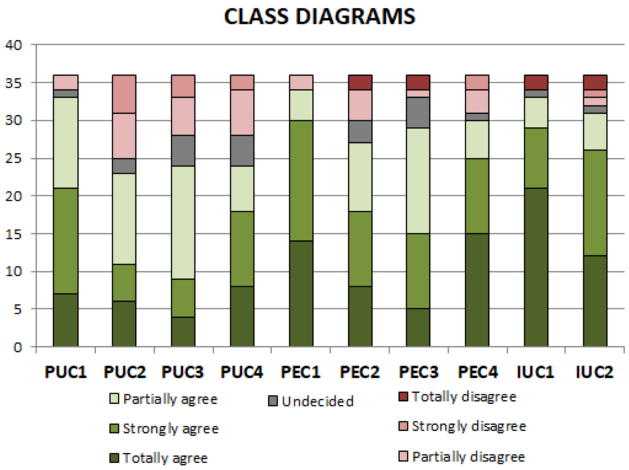

Figure 5: Answers to perceived ease of use, perceived usefulness and intention to use the class diagrams

During the FG, lovers highlighted in the FG board that: these diagrams are easy because they are directly related to the system domain, making it useful for the modeling process as a whole. Lovers intend to use it to develop object-oriented projects and due to the utility for the database.

On the other hand, haters reported that: the class diagram is not easy to use because it requires the knowledge of the whole team that will use it, and this can affect the usefulness of the diagram. In this discussion, there was a replica of the lovers to the haters. Lovers disagreed about using the relational model instead of the class diagram because it is necessary to think about the whole system, and this is possible with the class diagram.

In the open questions, the participants reported that: "The class diagrams are very useful to give information about the structures for those who will program in Java" (P4) and "The understanding of the class diagrams is easy, but modeling is not. I would use this diagram because it is the first structural diagram for the project" (P10).

In summary, students argue that class diagram modeling is not easy, since students should think about the attributes and relationships between classes. The class diagram was considered useful for system design.

4.3. Sequence Diagrams

Figure 6 presents the results regarding the perceived usefulness (PUS1 to PUS4), perceived ease of use (PES1 to PES4) and intention to use (IUS1 and IUS2) statements that are related to sequence diagrams. Most students intend to adopt this diagram in their future projects, despite the disagreement regarding ease of use and usefulness.

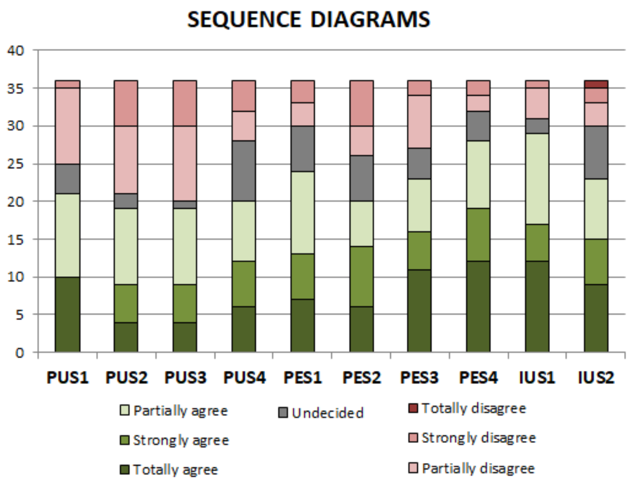

Figure 6: Answers to perceived ease of use, perceived usefulness and intention to use the sequence diagrams

During the FG, lovers emphasized that: sequence diagrams are easy because they show the sequence of methods that occur in the system, making it useful for implementation. For this reason, lovers intend to use it. The haters emphasized that: sequence diagrams are not easy to use alone, as it needs other diagrams, such as use cases and class diagrams.

Table 2 Factorial validity for the TAM constructs

	PE1		PE2		PE3		PE4		PU1		PU2		PU3		PU4	
	F1	F2	F1	F2	F1	F2	F1	F2	F1	F2	F1	F2	F1	F2	F1	F2
Use Case	**0.846**	0.243	**0.801**	0.229	0.532	**0.586**	0.064	**0.886**	0.054	**0.913**	0.381	**0.806**	**0.932**	0.016	**0.825**	0.148
Class	**0.803**	0.419	**0.949**	0.023	0.575	**0.671**	0.095	**0.810**	0.176	**0.889**	0.113	**0.839**	**0.917**	0.201	**0.916**	0.151
Sequence	**0.823**	0.424	**0.904**	0.197	0.515	**0.783**	0.204	**0.918**	0.301	**0.910**	0.162	**0.868**	**0.934**	0.229	**0.852**	0.217
State Machine	**0.886**	0.332	**0.944**	0.226	0.711	**0.564**	0.053	**0.899**	0.414	**0.803**	0.448	**0.769**	**0.921**	0.181	**0.829**	0.276
Activity	**0.898**	0.244	**0.878**	0.274	0.075	**0.868**	0.323	**0.763**	0.196	**0.861**	0.331	**0.848**	**0.920**	0.201	**0.645**	0.463

In the questionnaires, some participants disagreed with PUS2 and PUS3 (P1) and PES2 and PES3 (P30), as noticed in the following quotes: *"I disagree with the ease of use because there are many structures that need to be checked in order to do the modeling. This is more complex than other diagrams"* (P1) and "*The sequence diagram is one of the most complex and difficult to design. Reading the information in this diagram is easy, but abstracting information for this diagram is not an easy task"* (P30). Through these results, we notice that the participants did not consider modeling with this diagram easy.

4.4. State Machine Diagrams

Figure 7 presents the answers to the perceived usefulness (PUSe1 to PUSe4), perceived ease of use (PESe1 to PESe4), and intention to use (IUSe1 and IUSe2) statements related to state machine diagrams. There is disagreement among students regarding the adoption of this diagram in their future projects, despite the few disagreements regarding PU and PE.

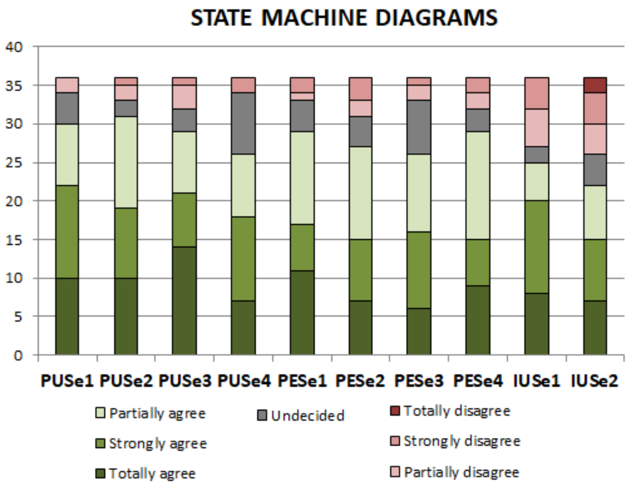

Figure 7: Answers to perceived ease of use, perceived usefulness and intention to use the state machine diagrams.

The FG board pointed out that: state machine diagrams are easy to model because the modeling elements are intuitive. They were also considered useful because they are easily adaptable to the system. Lovers intend to use it to demonstrate complex situations which the state diagram can simplify. Haters reported that this diagram could be confused with the activity diagram, especially by some tools that enable the creation of this diagram and have similar elements to those of the activity diagram.

During the FG, there was no replica from lovers to haters. In the questionnaires, most parts of the participants stated that they would use it. For example: *"The UML has many diagrams that do the same thing. So, I do not find this diagram so useful compared to the others"* (P6) and *"This diagram makes me see the way some system functionalities behave, which helps the understanding of the problem as the whole"* (P20). Some participants disagreed with the statements related to IU of this diagram. We noticed that these participants consider that other diagrams provide a better understanding of the information when compared to this diagram.

4.5. Activity Diagrams

Figure 8, once again, presents the participants' answers to the perceived usefulness (PUA1 to PUA4), perceived ease of use (PEA1 to PEA4) and intention to use (IUA1 and IUA2) statements related to activity diagrams. In this diagram, there are few disagreements among students about PE, PU, and IU.

In the FG discussion, the lovers reported that: activity diagrams are easy to model because they follow a logical sequence, making it useful for demonstrating the application operation flow. Thus, lovers intend to use it to understand what each actor can do in the system. Haters reported in the FG board that they do not intend to use this diagram because of the preference of using formal algorithms. In the FG of this diagram, there were no replicas because the lovers and haters stated that they highlighted different points. In the questionnaire, P6 reported that: *"This diagram also shows the activities of the system that can be understood through other diagrams"*, and the P28 reported: *"I consider that this diagram requires much effort to do, due to the delay in writing all the activities and inserting the flow..."*

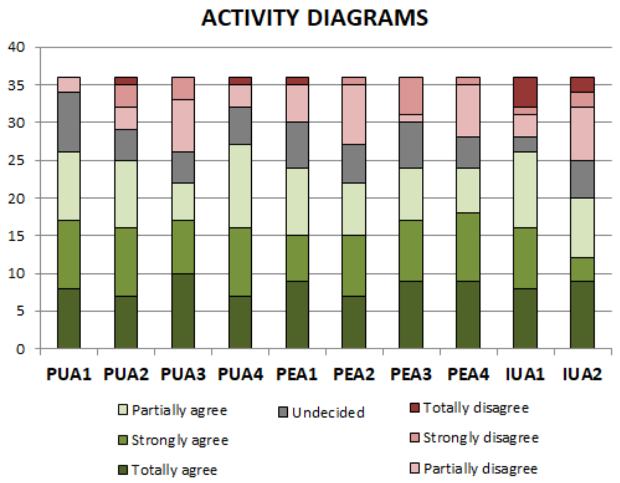

Figure 8: Answers to perceived ease of use, perceived usefulness and intention to use the activity diagrams

This diagram may not be adopted because participants do not consider modeling with this diagram an easy task. The participants also do not consider it useful, since other diagrams could represent the information presented by it.

5 Discussion

We start this discussion with considerations about the students' perceptions about usefulness and ease for modeling and comprehending of these models, answering RQ1 – "**Which UML diagrams are considered easier to use and useful by undergraduate students?**". Table 3 presents a summary of the findings of each UML diagrams investigated in this study.

In Table 3, we highlighted the findings regarding the UML diagrams considered easier to use and useful. We notice that the use case diagram and specifications, and class diagram were considered more useful, but modeling with these diagrams is not easy. As future software engineers, the students perceive use cases as useful to understand the information related to features that need to be developed in the system. The class diagrams are useful to understand the information about the structures related to the system. State machine diagrams were considered easy for modeling and useful for understanding. However, the participants stated that other diagrams provide a better understanding of the information when compared to this diagram.

Regarding the RQ2 **– Why do the students plan to adopt (or not) these diagrams as future software engineers?** - students perceived the use case diagrams and class diagrams as useful to be adopted in the software development process. We noticed that these students intend to use only UML use case and class diagrams. The perceived ease of use did not have much influence since the participants stated that they intend to use these diagrams, although the modeling with this diagram is not easy. Perceived usefulness construct may have influenced the intention to use these diagrams since some participants stated their modeling is not easy. With these results, we hypothesize that the training received by students may be one of the causes of UML diagrams are not adopted in the industry. These results are relevant to help instructors to improve how they teach UML diagrams, highlighting their importance in the software development process. We encourage the instructors to adopt teaching strategies to show the importance of UML diagrams in the software development process, since some diagrams were not considered useful in our study results.

Table 3 Summary of the perception of undergraduate students on UML diagrams

DIAGRAMS	FINDINGS
Use Case	It provides an overview of the system and was considered easy and useful for modeling and understanding. **If the system is complex, the modeling and understanding of this diagram can be considered difficult**.
Class Diagrams	It was **easy to understand its information. However, modeling with this diagram is not easy**. It is useful for understanding the implementation of systems with object-oriented languages.
Sequence Diagrams	It was considered **useful for understanding the information.** The students said that it is necessary to have the support of other diagrams that they find useful. **However, modeling with this diagram is not easy**.
Activity Diagrams	It **was not considered easy for modeling and may not be considered useful for the comprehension of the software** due to the usefulness other diagrams.
State Machine Diagrams	It was considered **easy for modeling and useful for understanding**. However, the participants stated that other diagrams provide a better understanding of the information when compared to this diagram.

The Focus Group and TAM enabled us to understand the students' perceptions about UML acceptance. We also considered that the Focus Group and TAM can be applied for other software engineering topics taught by different instructors. Instructors play an essential role in showing the importance of software engineering topics and this approach can help instructors to improve how they teach these topics.

This study was limited to 36 participants. Our goal is to investigate a first case of students' perception the UML diagrams. This paper can point to a direction for further studies to evaluate the underlying reasons. This is the first study with results from a single context, but other researchers may replicate it, providing outcomes in different contexts.

6 Threats to Validity

We discuss these threats with the classification of threats to validity presented by Wohlin et al. [19].

Internal validity. We acknowledge that there are threats related to internal validity, such as *total time used for the exploratory study*. To mitigate this threat, we conducted a pilot phase to adjust the time required to perform the tasks. Regarding the methods used, threats related to internal validity concerns with the way that *we elicited the perceptions from discussions and if they represent students' perceptions on UML diagrams*. We reduced this threat by using it with TAM based questionnaires. Students were informed that their answers should be following with their perceptions regarding each UML diagrams; that they should not assume the previous roles of lovers and haters.

External validity. We identify threats related to external validity, such as *participants of a single educational institution*. This study was limited to 36 participants of a single educational institution analyzing different UML diagrams. To alleviate this threat, we invited students from two different classes, taught by different instructors. Regarding *UML diagrams evaluated*, we analyzed a subset of five UML diagrams. However, these UML diagrams are often used in the industry [20].

Construct validity. Regarding the *validity of the TAM questionnaires adapted* in our research, we analyzed the reliability to guarantee the internal validity and consistency of the surveys. The results of the Cronbach Alpha and factorial analysis showed that the results are reliable, and the items are loading to the correct factors.

Conclusion validity. Threats related to conclusion validity concerns the relation between treatment and outcome. We tried to mitigate it by combining data from different resources, such as quantitative and qualitative data. These data were obtained with videos and questionnaires. We analyzed these data to answer our research questions.

7 Conclusions and Future Work

In this paper, we presented an exploratory study to understand student's perception regarding different UML diagrams better. We applied the Technology Acceptance Model (TAM) to analyze their acceptance and Focus Group (FG) to better explore the student's perception regarding these diagrams. Although there are several studies on the use of UML, an investigation about the student's perception of these diagrams was neglected by the literature.

In our study, the students' perceptions of UML diagrams indicate that they intend to use the use case and class diagrams in system design. They considered the use case is useful to understand the information related to features that need to be developed in the system and the class diagrams are useful to understand the information about the structures related to the system. The perceived ease of use did not have much influence since the participants stated that they intend to use these diagrams, although the modeling with this diagram is not easy. We noticed that these students did not intend to use the sequence, activity, and state machine diagrams in the software development process. These results may be related to the training received by students. With these results, we hypothesize that the training received by students may be one of the causes of UML diagrams are not adopted in the industry, yet further studies are required to confirm this. Other researchers may replicate it (with the instruments made available [16]), providing outcomes in different contexts. Thus, this paper can point to a direction for further studies to evaluate the students' perceptions of UML diagrams to confirm whether one of the causes of UML diagrams is not adopted by some software engineers, as pointed out in [5].

In addition, we believe that our results may help instructors to reflect on how they teach UML. Our findings can benefit instructors who are interested in understanding the perceptions of students regarding the ease of use and usefulness of UML diagrams. Thus, instructors can reinforce to students the benefits that were not perceived by them regarding these UML diagrams. Different approaches for teaching can be adopted to minimize such gaps, such as Active Learning [21] and Problem Based Learning [22]. These approaches can stimulate the immersion of students in the design of systems as future practitioners during learning. Through the use of methods and approaches, students' disagreements with perceived ease of use can be reduced.

As future work, we intend to investigate the instructors' point of view on the students' perception regarding the UML diagrams taught, analyzing the approaches' influence used for teaching UML. Thus, we hope that this research advance and we will understand whether one of the causes of UML diagrams is not adopted is related to training received by software engineers.

ACKNOWLEDGMENTS

We thank the undergraduate students for their participation in the exploratory study. We also thank our reviewers for doing an insightful review, which has greatly improved the content and clarity of this paper. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nivel Superior – Brasil (CAPES) – Finance Code 001, and through process number 175956/2013. We also would like to thank the financial support granted by CNPq (through processes numbers 430642/2016-4, 423149/2016-4 and 311494/2017-0) and National Science Foundation (award number 1900903).

REFERENCES

[1] B. Wei, H. S. Delugach, E. Colmenares, and C. Stringfellow, A Conceptual Graphs Framework for Teaching UML Model-Based Requirements Acquisition, in Proceeding 29th International Conference on Software Engineering Education and Training, 2016, pp. 71–75.

[2] OMG, Unified Modeling Language Specification (v 2.0), 2004.

[3] M. H. Osman and M. Chaudron, UML Usage in Open Source Software Development: A Field Study, in International Workshop on Experiences and Empirical Studies in Software Modelling, 2013, pp. 23–32.

[4] A. M. Bhutto and D. M. A. Hussain, Formal verification of UML profile, in Australian Journal of Basic and Applied Sciences, 2011, vol. 5, no. 6, pp.1594-1598.
[5] M. Petre, UML in Practice, in Proceedings of the 35th International Conference on Software Engineering, 2013, pp. 722–731.
[6] J. A. Cruz-Lemus, A. Maes, M. Genero, G. Poels, M. Piattini, The impact of structural complexity on the understandability of UML statechart diagrams, 2010, in Journal Information Sciences, vol. 180, no. 11, pp. 2209-2220.
[7] G. Jošt, J. Huber, M. Heričko, G. Polančič, An empirical investigation of intuitive understandability of process diagrams, in Journal Computer Standards and Interfaces, vol. 48, no.1, 2016, pp. 90-111.
[8] V. Venkatesh, and F. D. Davis, A Theoretical Extension of the Technology Acceptance Model: Four Longitudinal Field Studies, in Management Science, vol. 46, no. 2, 2000, pp. 186-204.
[9] B. B. de França, T. V. Ribeiro, P. S. M. Santos, and G. H. Travassos, Using Focus Group in Software Engineering: Lessons Learned on Characterizing Software Technologies in Academia and Industry, in Proceedings of the 18th Conferencia Iberoamericana en Software Engineering, 2015, pp. 351-364.
[10] N. Marangunić and A. Granić, Technology Acceptance Model: A Literature Review from 1986 to 2013, in Information Society, vol. 14, no.1, 2015, pp. 81-95.
[11] K. Siau and P-P. Loo, Identifying difficulties in learning UML. Journal of Information Systems Management, vol. 23, no. 1, 2006, pp. 43-51.
[12] R. Szmurło, F. Ortin, D. Zapico and J. M. Cueva, Design Patterns for Teaching Type Checking in a Compiler Construction Course, in IEEE Transactions on Education, vol. 50, no. 3, 2007, pp. 273 – 283.
[13] L. Burgueño, A. Vallecillo and M. Gogolla, Teaching UML and OCL models and their validation to software engineering students: an experience report, in Journal Computer Science Education, vol. 28, no. 1, 2018, pp. 23 -41.
[14] T. C. Lethbridge, Teaching Modeling Using Umple: Principles for the Development of an Effective Tool, in Proceedings of the 27th Conference on Software Engineering Education and Training, 2014, pp. 23-28.
[15] OMG, Object constraint language specification, chapter 7 (Number OMG ptc/08-06-08), 2000.
[16] A. Lopes, I. Steinmacher, and T. Conte. 2019. Analyzing the Students' Perception of UML Diagrams: Instruments Used in Evaluation. USES Research Group Technical Report, TR-USES-2019-0008.
Available: https://doi.org/10.6084/m9.figshare.9118949.v2
[17] D. G. Bonett and T. A. Wright, Cronbach's Alpha Reliability: Interval Estimation, Hypothesis Testing, and Sample Size Planning, in Journal of Organizational Behavior, vol. 36, no. 1, 2014, pp. 3–15.
[18] A. Lopes, I. Steinmacher, and T. Conte. 2019. Analyzing the Students' Perception of UML Diagrams: Results of the TAM (Technology Acceptance Model). USES Research Group Technical Report. TR-USES-2019-0007.
Available: https://doi.org/10.6084/m9.figshare.9118952.v1
[19] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén. 2000. Experimentation in Software Engineering: An Introduction (1st ed.). Kluwer Academic Publishers, Norwell, MA, USA.
[20] G. Reggio, M. Leotta, F. Ricca, and D. Clerissi, What are the used activity diagram constructs? A survey, in Proceedings of the 2nd International Conference on Model-Driven Engineering and Software Development, 2014, pp. 87–98.
[21] S. A. A. Freitas, W. C. M. P. Silva, G. Marsicano, Using an Active Learning Environment to Increase Students Engagement, in Proceedings of the 29th IEEE Conference on Software Engineering Education and Training (CSEET 2016), 2016, pp. 232-236.
[22] S. C. Santos, M. C. M. Batista, A. P. C. Cavalcanti, J. Albuquerque, S. R. L Meira, Applying PBL in Software Engineering Education, in Proceedings of the 29th IEEE Conference on Software Engineering Education and Training (CSEET 2009), 2009, pp. 182-189.