Professional Status and Expertise for UML Class Diagram Comprehension: An Empirical Study

Zéphyrin Soh[1,2], Zohreh Sharafi[1], Bertrand Van den Plas[3], Gerardo Cepeda Porras[4],
Yann-Gaël Guéhéneuc[1], and Giuliano Antoniol[1]
[1] Ptidej Team, SOCCER Lab, DGIGL, École Polytechnique de Montréal, Canada
[2] Department of Computer Engineering, UIT, University of Ngaoundéré, Cameroon
[3] FUNDP de Namur, Faculté d'Informatique, Belgique
[4] DIRO, Université de Montréal, Canada
{zephyrin.soh, zohreh.sharafi, yann-gael.gueheneuc, giuliano.antoniol}@polymtl.ca
vandenplas.bertrand@gmail.com, gerardocepeda@gmail.com

***Abstract*—Professional experience is one of the most important criteria for almost any job offer in software engineering. Professional experience refers both to professional status (practitioner vs. student) and expertise (expert vs. novice). We perform an experiment with 21 subjects including both practitioners and students, and experts and novices. We seek to understand the relation between the speed and accuracy of the subjects and their status and expertise in performing maintenance tasks on UML class diagrams. We also study the impact of the formulation of the maintenance task. We use an eye-tracking system to gather the fixations of the subjects when performing the task. We measure the subjects' comprehension using their accuracy, the time spent, the search effort, the overall effort, and the question comprehension effort. We found that (1) practitioners are more accurate than students while students spend around 35 percent less time than practitioners, (2) experts are more accurate than novices while novices spending around 33 percent less time than experts, (3) expertise is the most important factor for accuracy and speed, (4) experienced students are more accurate and spend around 37 percent less time than experienced practitioners, and (5) when the description of the task is precise, the novice students can be accurate. We conclude that it is an illusion for project managers to focus on status only when recruiting a software engineer. Our result is the starting point to consider the differences between status and expertise when studying software engineers' productivity. Thus, it can help project managers to recruit productive engineers and motivated students to acquire the experience and ability in the projects.**

***Index Terms*—Program Comprehension, Professional status, Expertise, Eye–tracking.**

I. Introduction

Context and Problem: A software engineer (practitioner) is a software designer who has been working in an industrial context on the design of software systems, on the contrary to a student designer who has performed design activities during her studies only. On the one hand, the "environment" in which the designers worked defines their *professional status* as practitioner or student. On the other hand, a *designer's experience* relates to the years of experience that a designer has worked on design projects. Project managers wonder whether experience or status can help them to choose a new software engineer. Empirical study with practitioners and students, experts and novices is a starting point to assess one designer against the others. In this paper, we work on the design of UML class diagrams and consider that the maintenance process can be divided into three different steps: the comprehension of the task–question to perform, the comprehension of a class diagram (through the search of the elements to use), and the realization of the task.

Goal: In this paper, we study the relations between professional status, years of experience, and the comprehension of UML class diagrams. We study whether the professional status of a designer and–or her years of experience affect her accuracy and speed to understand class diagrams when performing a maintenance task. Knowing that the effect of status and experience can also be impacted by the level of details in the formulation of the task to perform, we also study the possible effect of the precision of the question on class diagram comprehension (the level of details.)

Motivations: Previous work (in Section V-A) studied expertise but did not consider maintenance tasks on UML class diagrams. They did not distinguish between status and expertise and, thus, did not provide project managers with a clear difference between these factors. Thus, managers may be wondering which factor to emphasize for both their job descriptions and interviews. Eye-tracking systems have been used to study the comprehension of UML class diagram (in Section V-B) but not to investigate separately the effect of professional status and expertise on program comprehension.

Study: We conducted an experiment with three UML class diagrams for three systems: ArgoUML, JUnit, and QuickUML. The experiment was performed with 21 subjects (9 practitioners and 12 students). The student subjects were undergraduate and graduate students in software engineering programs. All subjects had experience (means = 3.04 years and variance = 1.88) with the design of UML class diagram. We asked the subjects to perform one maintenance task on each UML class diagram. We used an EyeLink II eye-tracking system to gather the eye fixations of the subjects while performing the task. Then, we measured the subjects' comprehension with fixation-related metrics. A fixation is the stabilization of the eye for a sufficiently-long period of time to allow cognitive processing.

Relevance: As project managers, we conjectured that the professional status and expertise affect the comprehension of class diagrams when performing maintenance tasks. We also conjectured that the comprehension is affected by the precision of questions. Knowing that the professional status and–or expertise affect the comprehension of the class diagrams could (1) prove the differences between status and expertise and (2) arouse the consideration of these differences by researchers in the new research. In practice, understanding the impact of status and expertise can also guide project managers to recruit more productive engineers, depending on their needs. It can also motivate students to work on course projects that might allow them to acquire "experience" and "practitioner's ability".

Results: Our study shows that practitioners and experts are more accurate than students and novices. Yet, students and novices spend less time to perform a maintenance task given a same accuracy, as shown in Figures 1 and 3 and discuss in the following sections. Our results also show that the expertise is the most important factor for the comprehension of a UML class diagram. We find evidence that novice students can be accurate if provided with precise tasks.

Organization: The paper is organized as follow: Section II presents the empirical definition and design. Section III presents the study results and discussions while Section IV discusses the threats to its validity. Section V presents the related work. Section VI concludes with future work.

II. Study Definition and Design

The *goal* of our experiment is to study the relations between professional status, expertise, the precision of a question, and the comprehension of UML class diagrams. The *quality focus* are the designer expertise and her professional status. When a designer performs a maintenance task on UML class diagrams, her expertise (expert or novice) and–or professional status (practitioner or student) can affect the comprehension of the class diagrams. The *perspective* is that the effect of expertise and status on comprehension can enhance their differences. Thus, researchers should consider these difference when studying class diagram comprehension. Also, the effect of experience and status on comprehension can help project managers choose the most helpful designers. The result can also bring evidence to students about the necessity to acquire "experience" and "practitioner's ability" within their study program. The *context* of the study consists of three UML class diagrams from three systems: ArgoUML, JUnit, and QuickUML. When the subjects perform the tasks, we use an EyeLink II eye-tracker to gather the subjects' eye fixations.

A. Research Questions and Hypotheses

We now formulate our research questions and related null hypotheses. For the sake of simplicity, we define a "parameterized" version of the hypotheses. The values of the "parameters" are in Tables I and II.

RQ1: What is the relation between a designer's professional status and her class diagram comprehension? We study whether the professional status of a designer affects her ability to understand a class diagram when performing a maintenance task. The parameterized null hypothesis related to the professional status is the following H_{0N_S}, where N is the dependent variable number (See Table II) and S is the factor (See Table I): H_{0N_S}: There is no difference in **Dependent Variable** between **Value 1** and **Value 2** when performing a maintenance task with a UML class diagram.

TABLE I
Independent Variables

Factor	Value 1	Value 2
S (Status)	practitioner	student
E (Expertise)	expert	novice
P (Question Precision)	precise	not precise

TABLE II
Dependent Variables

Number	Dependent Variable
1	the average accuracy
2	the time spent
3	the search effort
4	the overall effort spent
5	the question comprehension effort

For example, we obtain hypothesis H_{01_S} by replacing the appropriate parameters corresponding to the independent and dependent variables in Tables I and II: H_{01_S}: There is no difference in *the average accuracy* between *the practitioners* and *the students* when performing a maintenance task using an UML class diagram.

RQ2: What is the relation between the expertise of a designer and class diagram comprehension? We study whether the expertise of a designer affects her ability to understand a class diagram when performing a maintenance task. The related null hypotheses are similar to the **RQ1** hypotheses, but use the factor E for **Value 1** and **Value 2** parameters. For example, H_{02_E} is: H_{02_E}: There is no difference in *the time spent* between *the experts* and *the novices* when performing a maintenance task using an UML class diagram.

RQ3: What is the most important factor between the expertise and the professional status? This research question is a more refined and actionnable answer to the two previous research questions.

RQ4: What is the effect of the question formulation on the comprehension of a UML class diagram? We study the relation between the precision of the question (the level of detail) and the comprehension of class diagrams. The parameterized null hypothesis related to the precision of the question is: H_{0N_P}: There is no difference in **Dependent Variable** when performing **Value 1** and **Value 2** maintenance tasks using a UML class diagram.

For example, the concrete hypothesis H_{03_P} is: H_{03_P}: There is no difference in *the search effort* when performing *a precise* maintenance task and *a not precise* maintenance task using an UML class diagram.

TABLE III
CHARACTERISTICS OF THE CLASS DIAGRAMS

	Number of classes/ Interfaces	Average number of attributes per Class/Interface	Average number of methods per Class/Interface
ArgoUML	10	0.4	8.6
JUnit	14	0.57	6.14
QuickUML	16	1.75	3.87

B. Objects

For our study, we extracted three UML class diagrams from three different open-source systems: ArgoUML, JUnit, and QuickUML. ArgoUML is an UML-based modelling tool. JUnit is a unit testing framework for the Java programming language. QuickUML is a design object-oriented software with an integrated, core set of UML models. We chose these systems because (1) we had access to their source code to generate class diagrams and (2) the systems features were in different domains (modelling and testing). Also, these systems were already used in the previous experiments, *e.g.*, [1][2], and they are available to use to replicate our experiment.

We used reverse-engineering tools to obtain the complete class diagrams of the three systems. We extracted parts of the diagrams needed to perform the maintenance tasks. Table III shows the characteristics of the extracted UML class diagrams.

C. Questions

The questions asked to the subjects, summarized in Table IV, triggered their maintenance tasks.

D. Independent variables

We base our independent variables on the characteristics that a project managers look for in a candidate software engineer:

- Professional status (Practitioner or Student): practitioners are the subjects for whom the software design is a main activity. They worked or are working in the software engineering industry. The students are still studying.
- Expertise (Expert or Novice): we focused on the relative expertise of the subjects [3] and we use the years of experience to identify experts and novices. We perform the two-tailed version of the unpaired Wilcoxon test, using the Bonferroni correction, to split the subjects in two groups: experts or novices. The detailed splitting procedure is discussed in Section II-I.

E. Dependent Variables

We measure the comprehension of class diagrams by using the following dependent variables:

- **Accuracy**: It is the correctness of the subjects' answers. We use the Percentage of Correct Answer (PCA), as previous works [4], [5], [6], [7], [8].
- **Time Spent**: It denotes the time taken by a subject to perform a given task [8].
- **Search Effort**: It measures the effort spent by a subject to search the constituents to use to perform the task. We use both the convex hull area and the spatial density of the fixations to determine the search effort. The convex hull area represents the smallest convex set of fixations that contains all the subject' fixations. Goldberg and Kotval [9] used this measure to evaluate the quality of the user interfaces. A smaller value indicates that the fixations are closed together and that the subject spends less effort (search effort) to find usable elements (class, method, and so on) in the class diagram. The spatial density [9] measures the dispersion of the subjects' fixations. A lower value indicates that the subject uses the information gathered from the previous fixations or her knowledge: she spends less time and effort exploring the diagram.
- **Overall Effort**: It measures the overall effort spent by a subject during the task. We use the Average Fixation Duration (AFD) [4] and the Normalized Rate of Relevant Fixations (NRRF) [10] to determine the overall effort spent. Both measures are processing measures (*e.g.*, correlated to cognitive functions) [9]. The formula of the two measures are:

$$AFD = \frac{\sum_{a \in AOI} d(a)}{\sum_{a \in AOI} f(a)}$$

$$NRRF = \frac{\frac{\sum_{a \in AORI} f(a)}{\#AORI}}{\frac{\sum_{a \in AOII} f(a)}{\#AOII}}$$

  where $d(a)$ is a function that returns the total *duration* of the fixations made to an AOI (Area Of Interest) a, and $f(a)$ is a function that, given an AOI a, returns the number of *fixations* performed by a subject in that area. An Area Of Relevant Interest (AORI) is any class that is relevant for the task at hand and, therefore, should receive more subjects' attention during the task [10]. An Area Of Irrelevant Interest is other classes in a diagram. An Area Of Interest (AOI) is any class in a class diagram:

$$AOI = \{AORI\} \cup \{AOII\}$$

- **Question Comprehension Effort** (QCE): It measures the effort of the subject to comprehend a question. We define the Normalized Duration in Question Area (NDQA) and Normalized Fixations in Question Area (NFQA) to access the question comprehension effort. The NDQA is the normalized time spent on the Question Area (QA). The formula of NDQA and NFQA are:

$$NDQA = \frac{d(QA)}{\sum_{a \in AOI} d(a)}$$

$$NFQA = \frac{f(QA)}{\sum_{a \in AOI} f(a)}$$

The dependent variables Search effort, Overall effort, and Question comprehension effort are all defined using two measures. However, for the sake of simplicity and space, we do not divide each corresponding hypothesis in two, we describe in our analysis method in Section II-I how we will interpret the testing of these hypotheses.

TABLE IV
QUESTIONS FOR EACH SYSTEM

Objects	Questions
ArgoUML	We want to add a class named "Consultant" capable of adding some items to the todo list of a designer. How would you do that? Be specific about classes/methods/attributes.
QuickUML	We want to add a class "StatusBar" telling if a tool is selected. How would you do that? Be specific about classes/methods/attributes.
JUnit	We want to count the number of test runs. How would you do that? Be specific about classes/methods/attributes.

F. Mitigating Variables

We consider as main mitigating variables the choice of the question, because its precision (or lack thereof) could impact our results not matter the status or expertise of the subjects. Therefore, we have the following (extra) mitigating/independent variable:

- Question precision (Precise or Not precise): The precision is the level of details in the formulation of a question. We consider the question as precise if its formulation provides the kind of operation to perform (add/remove/update) and the kind of target element (class/method/attribute). For example, we consider the questions related to ArgoUML and QuickUML, shown in Table IV, precise because they guide the subjects by indicating that they will add a new class.

We also gather data about the numbers of hours that the subjects sleep per night on average and how many hours they slept the night before the experiment to control fatigue.

G. Subjects

The 21 subjects were all volunteers and we receive the approval from the Ethical Review Board of École Polytechnique de Montréal to conduct this experiment. There were 9 practitioners and 12 students and only one female student. Among the 9 professionals, 8 worked at Pyxis[1]. The student subjects were recruited through a registration Web page after announcing the experiment with posters. All subjects had at least a minimal experience with UML class diagrams.

H. Design and Procedure

We used three class diagrams in our experiment and we asked one maintenance task for each diagram. The experiment included displaying the stimuli on a screen and gathering the subjects' fixations while they performed a task. Each subject performed all tasks. Thus, our design is a within–subject design [11]. We combine one maintenance task with one class diagram to form a stimulus. After preparing the stimuli and recruiting the subjects, we performed the experiment. We followed the same procedure for each subject. First, we welcomed the subject and explained the goal of the study, how the eye-tracker system works, and that the experiment is related to UML class diagrams. We presented the context of the study by giving three kind of information about the objects to use in the experiment: the name of the systems, a brief presentation of its features, and a brief presentation of the part of system that we use. We also informed the subjects that the EyeLink II system is not dangerous, the collected data is anonymous, and they can abandon the experiment whenever they want without any consequences. Then, we simulated an experiment during which the subject could ask the questions. Second, we asked if the subject agreed to proceed with the study. Then, we calibrated the EyeLink II system to allow the system to record the eyes position of the subject, and other parameters to detect her eye movement. We recommended subjects to rest between two stimuli but without shifting their heads to avoid calibration problems. Also, the subjects announce their answer to each question out loud.

I. Analysis Method

We preprocessed and analyzed the collected data with the Taupe system [12]. We defined the Areas Of Interest on each stimuli and the subjects' characteristics such as gender. Then, Taupe provided the results of preprocessing in CSV files that we used into R [13] to apply statistical tests. All material and collected data is available on-line[2].

Because our data is not normally distributed, we used the non-parametric Wilcoxon un-paired test to investigate whether the comprehension of class diagrams is affected by the professional status and the precision of the question.

For the effect of expertise, the subjects had experience from 1 to 5 years. Thus, there were five groups (based on the years of experience) of subjects, *i.e.*, more than two samples to compare. Therefore, (1) we used the pair-wise Wilcoxon comparison with Bonferroni correction to assess any clear distinction; (2) we built four categories based on the years of experience, *i.e.*, $\{1,2\}$ vs. $\{3,4,5\}$, and $\{1,2,3\}$ vs. $\{4,5\}$, and so on, and assessed in which categorization (again using Bonferroni) were the differences in accuracy between groups significant; (3) among the statistically-significant categorizations, we chose the one with the highest Cliff's delta value.

Because we defined the three dependent variables Search effort, Overall effort, Question comprehension effort with two metrics, we will test each metric independently and will conclude on the corresponding hypotheses as follows: we will reject the hypotheses if there are no statistically-significant differences for both measures; we will accept the hypotheses if there are statistically-significant differences for both measures; finally, we will not conclude if one of the difference is statistically-significant but not the other.

[1]http://www.pyxis-tech.com

[2]http://www.ptidej.net/download/experiments/icpc12a/

TABLE V
SUMMARY OF HYPOTHESES TESTING (GRAY CELLS CONTAIN STATISTICALLY-SIGNIFICANT RESULTS)

	p-values							
	Accuracy	Time	Search Effort		Overall Effort		QCE	
			Convex hull	Spatial density	AFD	NRRF	NDQA	NFQA
Status	0.03867	0.01712	0.08634	0.02442	0.006578	0.5301	0.09718	0.8458
Expertise	4.818e-05	0.02798	0.01145	0.0006826	0.4772	0.4772	0.05053	0.5832
Question Precision	0.01937	0.8566	0.1986	0.9535	0.6486	0.2998	0.01375	0.5164

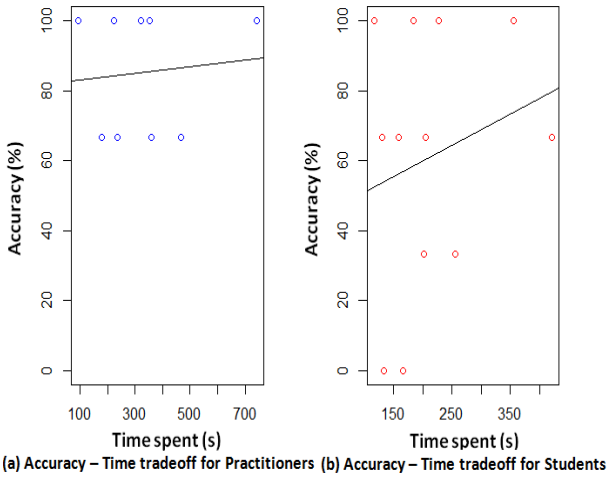

Fig. 1. Accuracy-Time Tradeoff for Professional status

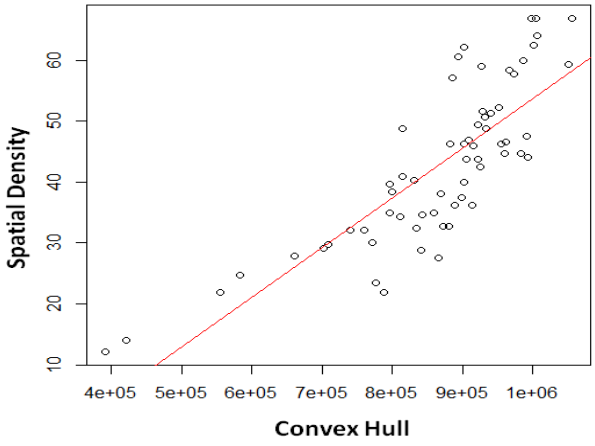

Fig. 2. Correlation for search effort measures

III. RESULTS AND DISCUSSIONS

We now present and discuss the results of our experiment. Table V summarizes the results of testing our hypotheses.

A. RQ1: Practitioners vs. Students

1) Results: Table VI shows in the last column that the practitioners are more accurate than the students (85.18% versus 61.11%). The difference in accuracy is statistically significant as shown in Table V (p-value = 0.03867). Moreover, there is significant difference for the time spent between practitioners and students (p-value = 0.01712) to perform the tasks. The students spent around 35 percent less time than practitioners (212.80 seconds versus 330.30 seconds). Figure 1 shows the tradeoff between accuracy and time spent for practitioners and students. It indicates that students could be more accurate when spending more time.

The spatial density difference (p-value = 0.02442) shows that the distribution of fixations are not similar for practitioners and students. Thus, some have more directed search strategy than others. The significant difference on Average Fixation Duration (p-value = 0.006578) between practitioners and students indicates that they have not the same ability to interpret or use class diagram elements.

In contrast, there are no statistically-significant differences for the convex hull and question comprehension effort. Thus, the practitioners and students have the same ability to understand the task to perform. A surprising result is that, even if the convex hull and the spatial density are related to the search effort, there is not significant difference for the convex hull. Figure 2 shows that the convex hull and spatial density are correlated with the Spearman non-parametric correlation coefficients equals to 0.84. We also observe that the convex-hull difference between practitioners and students is significant at 0.1 level of significance (p-value = 0.08634 $<$ 0.1).

2) Discussions: We can reject the null hypothesis H_{01_S}. Our study shows that practitioners are significantly more accurate than students. The fact that practitioners are more active and face real, industrial design problems daily can justify their ability to perform more accurately than students.

We can also reject the null hypothesis H_{02_S}. The time spent difference is similar to the experimental results of Arisholm and Sjøberg [14] who argues that "graduate students were faster than junior and intermediate professional consultants". We consider that our practitioners are at junior or intermediate levels because they had a maximum of five years experience.

Even with a correlation between the convex hull and spatial density, there is no significant difference on the convex hull between practitioners and students. The significant difference on spatial density and lack thereof on convex hull do not allow us to reject or accept the null hypothesis H_{03_S}. The spatial density difference indicates that practitioners and students explore class diagrams differently, yet with the similar efficiency.

The difference on AFD and lack of difference on NRRF do not allow us to reject or accept the null hypothesis H_{04_S}. The difference on AFD seems to indicate that practitioners and students have different abilities to interpret diagrams. We again explain this difference by practitioners' daily activities.

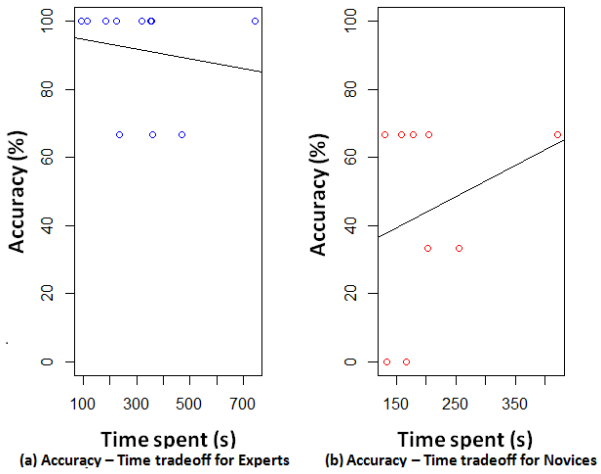

Fig. 3. Accuracy-Time Tradeoff for Expertise

We cannot reject the null hypothesis H_{05_S}. The non significant difference on question comprehension effort is probably because neither practitioners nor students had worked on the problem domains of the used class diagrams.

Finally, based on the metrics used, the main differences between practitioners and students are their accuracy and the time spent to perform the tasks. While practitioners are more accurate, students are faster. Therefore, practitioners are the best candidates to satisfy demanding users. On the contrary, students are the best designer for time-constrained projects. A design team may include both practitioners and students to benefit from both time and accuracy advantages.

B. RQ2: Experts vs. Novices

1) Results: The pair-wise comparison using two-tailed unpaired Wilcoxon test and the use of Cliff's delta ($d = 0.47$, major effect) allow us to distinguish two groups of subjects: the experts' group with 3, 4, and 5 years of experience and the novices' group with 1 and 2 years of experience. By considering the experts vs. novices categorization, Table VI, in its last column, shows that experts are more accurate than novices (91.66% versus 44.44%). Novices spent around 33% less time than the experts to perform the task (205.50 seconds versus 306.50 seconds). The accuracy–time tradeoff shown in Figure 3 suggests that novices are more accurate when spending more time.

To answer $RQ2$, we use the metrics defined in Section II-E to study whether the comprehension of the class diagrams depends on the subjects' expertises. The differences in accuracy and time are statistically significant, see Table V with p-values = 4.818e-05 and 0.02798, respectively. The effect of expertise is significant for the convex hull area (p-value = 0.01145) and the spatial density (p-value = 0.0006826). For the overall effort and the question-comprehension effort, the p-values show no significant differences.

2) Discussions: We can reject the null hypotheses H_{01_E} and H_{02_E}. The general characteristics of any experts identified by Chi [3] indicate that experts can be faster and more accurate than novices. The accuracy difference is confirmed in the software design field by our results. Thus, we can reject the null hypothesis H_{01_E}. In contrast, Atman *et al.* [15] contradict the time difference by arguing that "experts spent more time than novices qualitatively analyzing the problem posed". The experiment done by Burkhardt *et al.* [16] indicated that novices spent less time than experts, even if the difference was not significant. Also, Schenk *et al.* [17] found that experts spent more time than novices when performing analysis tasks. These results are similar to our findings because the design process includes analysis activities [14][3]. Thus, we can reject H_{02_E}

The null hypothesis H_{03_E} can be rejected. Due to the fact that the convex hull area and the spatial density are correlated and measure the exploration-search ability of the subjects, the significant differences with these two measures show that, while experts have a directed and efficient search of the relevant constituents, novices have an extended and inefficient search. These findings can be related to the results of Lee and Pennington [18], which show that, while the experts think directly from the understanding of the problem to the identification of the relevant elements, the novices are not able to analyze the problem directly through their objects.

The null hypotheses H_{04_E} and H_{05_E} cannot be rejected. There is no significant difference between experts and novices for the overall effort and question comprehension effort. Burkhardt *et al.* [16] found that there is no effect of expertise on the construction of the program and the situation models. The fact that the mental model (both program and situation models) is the consequence of the cognitive process can justify our results. Moreover, the experience of our participants are too close (from 1 to 5 years) and they can have almost the same cognitive abilities.

For the differences of experts and novices, we found that they differ for the accuracy ability, the time spent to perform the task, and the effort to find the relevant constituents. Experts and novices (with close years of experience) spent almost the same overall effort and question comprehension effort to perform the maintenance task on UML class diagram.

As for the practitioners and students difference, the experts are the best designers for accuracy and the novices are the best for time spent. But due to the difference between the experts' and novices' percentages of correct answers (47.22%), it seems that, although novices are faster than experts, the novices need more time to reach the same accuracy as experts.

C. RQ3: Status vs. Expertise

The aim of RQ3 is to compare two independent variables. We studied the most important factor impacting accuracy and time spent and we also considered the groups defined in the previous Section III-B.

Table VI shows that experts are more accurate and spend less time than practitioners. Experts spent around 7% time less than practitioners (306.50 seconds versus 330.30 seconds).

[3]"The existing literature provides no clear distinction between object-oriented analysis and object-oriented design" [14].

TABLE VI
OVERVIEW OF TIME SPENT (IN SECOND) AND PERCENTAGE OF CORRECT ANSWER (PCA)

Status	Expertise	Number	Mean	Std	Min	Q1	Median	Q3	Max	PCA (%)
Practitioner	**Expert**	8	349.40	219.5036	78.42	185.00	309.10	460.80	1035.00	87.5
	Novice	1	177.8	28.10993	145.4	168.5	191.7	194.0	196.2	66.66
	All	9	330.30	213.7843	78.42	183.80	271.60	436.70	1035.00	85.18
Student	**Expert**	4	220.70	118.4734	81.46	164.80	206.10	233.30	550.50	100
	Novice	8	208.90	105.3635	79.69	134.20	173.40	280.60	463.00	41.66
	All	12	212.80	108.3439	79.69	142.70	182.40	277.20	550.50	61.11
Total	**Expert**	12	306.50	199.6449	78.42	174.80	240.30	387.70	1035.00	91.66
	Novice	9	205.50	99.9035	79.69	140.80	173.80	278.00	463.00	44.44
	All	21	263.20	170.9586	78.42	148.10	205.30	309.10	1035.00	71.42

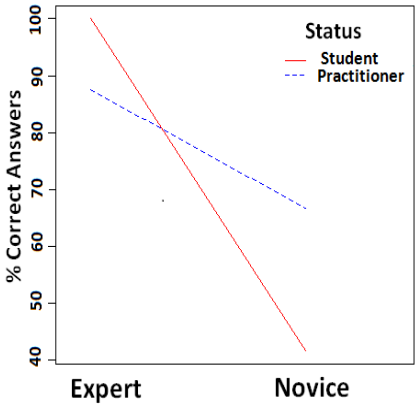

Fig. 4. Interaction Between the Expertise and the Status on the Accuracy

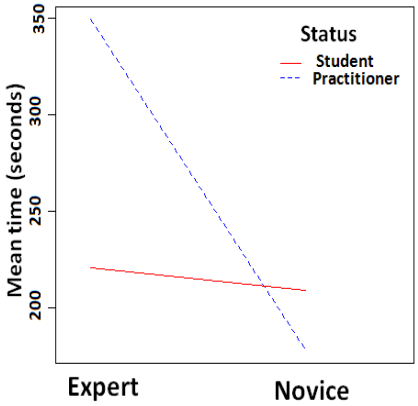

Fig. 5. Interaction Between the Expertise and the Status on the Time

This result indicates that, for the accuracy and the time spent to perform a task, expertise is the most important factor. Moreover, experienced students perform better (100% versus 87.5%) and spent around 37% percent less time than experienced practitioners (220.70 seconds versus 349.40 seconds).

The two-way interactions between expertise and status, in Figures 4 and 5, show that the effects of expertise on accuracy and time depends on the status. The professional status affects more the time spent by the experts than by the novices. On the contrary, the interaction between the professional status and expertise affects more the novices' accuracy.

Our results show that expertise is the most important factor for software engineers, in particular when the experience was acquired during studies. Thus, it is an illusion for project managers to focus on the professional status when recruiting a software design immigrant [19]. For practitioners, the fact that the students want to prove their know-how and that they frequently have class exams with short durations could justify their performance on accuracy and time spent.

D. RQ4: Precise vs. Not Precise Question

The precision of the question significantly affects the accuracy (p-value = 0.01937) and time spent on the question area (p-value = 0.01375). There is no other effect of the precision of the question. Thus, we can reject the null hypothesis H_{01_P}. The null hypotheses H_{02_P} to H_{04_P} can not be rejected. For the null hypothesis H_{05_P}, we cannot reject nor accept it because there is no effect of the precision of the question on the Normalized Fixation in the Question Area (NFQA).

The effect of the precisions of the questions on the time spent in the question area shows that some subjects understood the questions faster than others. If we consider experts and novices separately, the statistical test shows that the precision of the question affects the experts' time spent on question area (p-value = 0.02643) whereas it does not affect the novices' one (p-value = 0.1601). The same observation is true for practitioners (p-value = 0.01992) and students (p-value = 0.1767). Moreover, Table VII shows that the precision of the question affects more students' accuracy than practitioners' accuracy. The effect of the precision of the question is statistically significant for students (p-value = 0.01809) but not for practitioners (p-value = 0.4777). The same result is observed for the expertise where novices' accuracy is more affected than experts' accuracy, as reported in Table VII. The difference of accuracy is significant for novices (p-value = 0.01690) but not for experts (p-value = 0.2201).

The fact that the precision of the questions affects the accuracy but not the overall time spent to perform a task indicates that the precision does not help to save time. Moreover, the precision of the question does not assist the subjects in the exploration of the class diagram (no difference in the search effort). In addition, the fact that precision of the questions affects the students' accuracy more than practitioners' accuracy on the one hand and the novices' accuracy more than experts'

TABLE VII
OVERVIEW OF THE PCA FOR THE QUESTION PRECISION

Questions	Percentage of Correct Answer (%)				
	Status		Expertise		All
	Practitioner	Student	Expert	Novice	
Precise	88.88	75.00	95.83	61.11	80.95
Not precise	77.77	33.33	83.33	11.11	52.38

accuracy on the other hand indicates that novice students can be accurate when the description of the task is precise.

IV. THREATS TO VALIDITY

We now discuss the threats to the validity of our results.

A. Construct Validity

Our study can be impacted by two main threats to its construct validity: the metric used and the instrumentation of our independent variables. For the first threat, we avoided the consequence of using one single metric to measure the subjects' ability to understand an UML class diagram by using (1) the percentage of correct answer, (2) the time spent, and (3) the fixations-based metrics.

For the threats to instrumentation, we did not use a formal definition of the precision of questions. This lack of definition could make our experiment difficult to replicate. In future studies, we plan to define precision using some precision rate collected from post-experiment questionnaires. Apart from the precision of the question, the subjects' expertise and professional status were not predefined and instrumented by ourselves. They were defined by the data gathered from questionnaires. However, we did not inform the subject about the precise goal of the study. Expertise could be affected by other factors but we chose to use accuracy and years of experience to maintain the complexity of the categorization at a reasonable level. The subjects gave their answers orally, which we could have mis-recorded and–or could have made subjects uncomfortable. However, we were careful to record accurately each answer and all answers are anonymous.

The fact that we did not have all possible combinations of systems and precisions of questions could affect the results of our study. To address this threat, the subjects were not informed that there were precise and not precise questions. Moreover, we preferred to avoid learning bias rather than to define precise and not precise questions for a same system.

B. Conclusion Validity

The possible threats that can affect the conclusion validity of our study are the violated assumption of the statistical tests, the experimental settings, and the heterogeneity of the subjects. We made sure not to violate the statistical assumptions by using non-parametric tests and correlations that do not assume particular data distributions. On the contrary, we did not use no special analysis to answer *RQ3* because it studied the comparison of two different independent variables.

For the experimental settings, we performed our experiment in a quiet laboratory without distraction.

Regarding the subjects' heterogeneity, their differences were enough in term of professional status and years of experience. The practitioners were from the same software company and, due to the difficulty to find inexperienced practitioners, we had only one novice practitioner.

C. Internal Validity

We identify four threats to internal validity: learning, selection, fatigue, and instrumentation. We mitigated the learning threat by defining only one task per system. Thus, a subject could not learn by doing one task on one system.

For the selection threat, the differences in the subjects' diversity could affect our study. To mitigate the selection threat, all subjects were volunteers. In addition, the fact that we used the within–subject design in which all subjects perform all tasks also mitigated the selection threat. Finally, the subjects are representative of the junior software engineers that can work in a software company.

We did not limit the time to perform the tasks, which could affect our study because of the resulting fatigue and disadvantage the tasks always given at the end. Consequently, we presented the tasks to the subjects in different orders.

The use of the EyeLink II system could affect our study. As mentioned in the experiment procedure in Section II-H, we asked the subjects to minimize their head movements to avoid decalibration. However, head movements are unavoidable. To mitigate the instrumentation threat, we used a dentist chair and a travel pillow to give support to subjects' neck and head both to make them comfortable and avoid head movements.

D. External Validity

We conducted our experiment on three different systems with different characteristics as shown in Table III. The subjects were 9 practitioners and 12 students and only one female student. Consequently, we cannot argue that our results can be generalized to other systems and–or practitioners from other companies. The year of experience of the subjects varied between one and five years. We cannot assert that our results can be generalized to the large range of years of experience.

V. RELATED WORK

Our work is related to other works on the difference between experts and novices, and the class diagram comprehension using Eye-tracking system.

A. Experts and Novices Comparison

"Expertise" is the most studied factor that can impact engineers' productivity. It relates both to the engineers' performance (or acquired skills) and the way to improve their skills. In software engineering, the study of expertise can help project managers recruiting design immigrants (designers new to a project, similar to "programmers new to a project" [19]).

Chi [3] identified two approaches (*absolute* and *relative*) to study experts' characteristics. The most used approach is *relative expertise* in which academic qualification, years of experience, consensus among peers, and domain-specific knowledge are used to measure the experts' proficiency level.

Schenk *et al.* [17] used verbal reports to examine the difference in the requirements analysis process between experts and novices. Novices were identified using the years of experience and the experts were categorized with the rating scale of their supervisors. The quantity of verbalizations and task duration were use to compare experts and novices for the analysis task. They found that novices averaged less time than experts analysts. Our work differs in the kind of task and the method use for the comparison.

Arisholm and Sjøberg [14] presented a controlled experiment with two design alternatives of a system to investigate the difference between students (undergraduate and graduate) and three categories of professionals (junior, intermediate, and senior). They found that the graduate students were faster than juniors and intermediates professionals.

Adelson [20] studied the differences between experts and novices regarding the kind of question to be answered. She found that experts are better than novices for abstract ("what") questions and the novices are better than experts when a concrete ("how") question was asked. The novices were undergraduate students, whereas the experts were fellow teachers.

Others works focused on domain expertise. Adelson and Soloway [21] found that the behavior in the construction of the mental model depends on experience and familiarity with the problem domain.

When study the cognitive activities of object-oriented experts and object-oriented novices, Lee and Pennington [18] found that "the object-oriented novices were not able to analyze the situation directly through their objects, as were the object-oriented experts".

Burkhardt et al. [16] studied the difference between expert and novice on documentation and reuse task. They found that experts and novices differ on situation model, but not in program model for documentation tasks. For reuse tasks, they found no difference between experts and novices.

Previous works on the study of expertise identify the differences between the experts and the novices. The differences are found particularly in requirement analysis, design style, kind of questions to be answered, construction of mental model (both program and situation models), domain knowledge and familiarity. The method used in the previous works is the protocol analysis and the studies are performed on source code or textual descriptions of requirements. The previous works did not explicitly distinguish the professional status to the expertise when categorizing participants for their experiments. For example, while Adelson and Soloway [21] use unexperienced practitioners as novices, Burkhardt *et al.* [16] use unexperience students as novices. We think that the fact that some novices are practitioners and the others are students can affect the study results. Moreover, Arisholm and Sjøberg [14] [22] considered senior professionals with less years of programming experience than graduate students. We proposed to study "expertise" difference by distinguish the years of experience to the professional status in the categorization of the participants. Due to the limitations of verbal report, we used an eye-tracking system and the related metrics to study the expertise and professional status differences. In addition, we performed the study with the maintenance task on UML class diagram instead the others kind of task (analysis, documentation, reuse, etc.) on the source code and the textual description of the problem.

B. Class Diagram Comprehension Using Eye-Tracking

The eye-tracking system has been used to study the comprehension of the UML class diagram. Yusuf *et al.* [23] performed a study to identify the UML class diagram characteristics that can affect program comprehension. It shows that, depending on the UML expertise of the subjects, characteristics such as layout, color, and stereotypes play an important role on the UML class diagram comprehension. Sharif and Maletic [7] used accuracy and time to evaluate the effect of layout on UML class diagram comprehension. They found that the multi-cluster layout had a high accuracy and took less time compared to the orthogonal layout. When considering the three-cluster layout, they confirmed that both multi-cluster and three-cluster layouts were equally good when compared to the orthogonal layout for design tasks.

Jeanmart *et al.* [10] used the eye-tracking system to perform the experiment to investigate whether the visitor pattern affect the comprehension and maintenance of UML class diagram. They found that the visitor pattern does not affect the subject's effort when performing the comprehension tasks whereas it significantly reduce the effort for the maintenance tasks.

Cepeda and Guéhéneuc [4] studied the efficiency of the design pattern representation in the UML class diagram. They consider three representations of Design Pattern (stereotype-enhanced, pattern-enhanced, and canonical representation). They results indicate that the stereotype-enhanced representation is more efficient than others for identifying design patterns composition and role whereas the pattern-enhanced and canonical representations are more efficient than stereotype-enhanced for identifying participation in design pattern.

Existing works using eye-tracking to study the comprehension of UML class diagrams did not study professional status. They use the subjects' performance in performing the tasks [6], [23] and the subject's grades in the course in which they were enrolled [7] to distinguish experts and novices.

To the best of our knowledge, there is no previous work that uses the maintenance task on UML class diagrams and eye-tracking system to study and compare separately the professional status and the expertise. In our study, we distinguish the professional status and expertise for the categorization of the subjects and combine the study of UML class diagram with the comparison of experts and novices studies.

Unfortunately, most of our results on professional status difference can not be related to the existing work because the existing works do not distinguish the professional status and expertise when categorize their participants.

VI. Conclusion And Future Work

Professional experience is one of the most important criteria for almost any job offer in software engineering. We studied

the relations between professional status, expertise, precision of the question, and comprehension of UML class diagrams. We used the percentage of correct answers, the time spent to perform the maintenance tasks, the search effort, the overall effort, and the question comprehension effort to compare practitioners vs. students, experts vs. novices, and precise vs. not precise questions. We also investigated whether the professional status or expertise is the most important factor for UML class diagram comprehension.

We performed an experiment with 21 subjects performing maintenance tasks on UML class diagrams from three systems, ArgoUML, JUnit, and QuickUML. We used an EyeLink II system to gather data while the subjects performed a maintenance task. Table V summarizes the results of our hypotheses testing. Within the threats to the validity of our study detiled in Section IV, we found that practitioners and experts, respectively, are more accurate than students and novices while students and novices, respectively, spent less time than practitioners and experts. Expertise is the most important factor impacting the comprehension of UML class diagrams: we observed that expert students were more accurate than other categories of subjects. We also found that novices can be accurate when the description of the tasks to perform is precise.

Our results suggest that (1) project managers should be wary when recruiting a new software engineer by considering the experience acquired by students in their study program, (2) researchers should consider the differences between professional status and expertise when studying software engineers' productivity, and (3) students should acquire as much expertise as possible during their study, for example through projects realized when studying.

In future work, we plan to reproduce our experiment with other professionals from other software companies. We also plan to replicate our study with other systems. In addition to the above points, we will set a fixed time to perform the task to investigate how much such limited time affects the subjects' accuracy. We will also refine our study with finer-grain dependent variables and replicate it on source code.

ACKNOWLEDGMENT

The authors deeply thank Pyxis, their engineers, and the students who participated to our experiment. They also thanks the anonymous reviewers for their constructive comments. This work has been partly funded by the Canada Research Chairs on Software Patterns and Patterns of Software and on Software Cost-effective Change and Evolution, and the Agence Universitaire de la Francophonie.

REFERENCES

[1] N. Moha, Y.-G. Guéhéneuc, L. Duchien, and A.-F. Le Meur, "Decor: A method for the specification and detection of code and design smells," *Software Engineering, IEEE Transactions on*, vol. 36, no. 1, pp. 20–36, jan.-feb. 2010.

[2] L. Aversano, G. Canfora, L. Cerulo, C. Del Grosso, and M. Di Penta, "An empirical study on the evolution of design patterns," in *Proceedings of the the 6th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering*, 2007, pp. 385–394.

[3] M. T. H. Chi, "Two approaches to the study of experts' characteristics," in *The Cambridge Handbook of Expertise and Expert Performance*. Cambridge, U.K.: Cambridge University Press, 2006, pp. 21–30.

[4] G. Cepeda Porras and Y.-G. Guéhéneuc, "An empirical study on the efficiency of different design pattern representations in UML class diagrams," *Empirical Software Engineering*, vol. 15, no. 5, pp. 493–522, 2010.

[5] B. Sharif and J. I. Maletic, "An eye tracking study on camelcase and under_score identifier styles," in *IEEE 18th International Conference on Program Comprehension (ICPC)*, july 2010, pp. 196–205.

[6] ——, "An empirical study on the comprehension of stereotyped UML class diagram layouts," in *IEEE 17th International Conference on Program Comprehension*, may 2009, pp. 268–272.

[7] ——, "The effect of layout on the comprehension of UML class diagrams: A controlled experiment," in *5th IEEE International Workshop on Visualizing Software for Understanding and Analysis, 2009. VISSOFT 2009*, sept. 2009, pp. 11–18.

[8] M. Abbes, F. Khomh, Y.-G. Guéhéneuc, and G. Antoniol, "An empirical study of the impact of two antipatterns, blob and spaghetti code, on program comprehension," in *Proceedings of the 2011 15th European Conference on Software Maintenance and Reengineering*. IEEE Computer Society, 2011, pp. 181–190.

[9] J. H. Goldberg and X. P. Kotval, "Computer interface evaluation using eye movements: methods and constructs," *International Journal of Industrial Ergonomics*, vol. 24, no. 6, pp. 631–645, 1999.

[10] S. Jeanmart, Y.-G. Guéhéneuc, H. Sahraoui, and N. Habra, "Impact of the visitor pattern on program comprehension and maintenance," in *Proceedings of the 3rd International Symposium on Empirical Software Engineering and Measurement*, ser. ESEM '09. Washington, DC, USA: IEEE Computer Society, Oct 2009, pp. 69–78.

[11] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in Software Engineering - An Introduction*. Kluwer Academic Publishers, 2000.

[12] B. D. Smet, L. Lempereur, Z. Sharafi, Y. G. Guéhéneuc, G. Antoniol, and N. Habra, "Taupe: Visualising and analysing eye-tracking data," *Science of Computer Programming*, 2011, 2nd special issue on Experimental Software and Toolkits.

[13] "R web page." [Online]. Available: http://www.r-project.org/

[14] E. Arisholm and D. I. K. Sjøberg, "Evaluating the effect of a delegated versus centralized control style on the maintainability of object-oriented software," *IEEE Transactions on Software Engineering*, vol. 30, no. 8, pp. 521–534, aug. 2004.

[15] C. J. Atman, R. S. Adams, M. E. Cardella, J. Turns, S. Mosberg, and J. Saleem, "Engineering design processes: A comparison of students and expert practitioners," *Journal of Engineering Education*, vol. 96, no. 4, pp. 359–379, Oct. 2007.

[16] J.-M. Burkhardt, F. Détienne, and S. Wiedenbeck, "Object-oriented program comprehension: Effect of expertise, task and phase," *Empirical Software Engineering*, vol. 7, no. 2, pp. 115–156, 2002.

[17] K. D. Schenk, N. P. Vitalari, and K. S. Davis, "Differences between novice and expert systems analysts: what do we know and what do we do?" *Journal of Management Information System*, vol. 15, pp. 9–50, June 1998.

[18] A. Lee and N. Pennington, "The effects of paradigm on cognitive activities in design," *International Journal of Human–Computer Studies*, vol. 40, no. 4, pp. 577–601, April 1994.

[19] M.-A. Storey, "Theories, tools and research methods in program comprehension: past, present and future," *Software Quality Journal*, vol. 14, no. 3, pp. 187–208, 2006.

[20] B. Adelson, "When novices surpass experts: The difficulty of a task may increase with expertise," *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 10, pp. 483–495, Jul. 1984.

[21] B. Adelson and E. Soloway, "The role of domain expenence in software design," *IEEE Transactions on Software Engineering*, vol. 11, pp. 1351–1360, November 1985.

[22] E. Arisholm and D. I. K. Sjøberg, "A controlled experiment with professionals to evaluate the effect of a delegated versus centralized control style on the maintainability of object-oriented software," Simula Research Laboratory, Simula Technical Report 2003-6, 2003.

[23] S. Yusuf, H. Kagdi, and J. I. Maletic, "Assessing the comprehension of UML diagrams via eye tracking," in *Proceedings of the 15th IEEE International Conference on Program Comprehension*. Washington, DC, USA: IEEE Computer Society, Jun 2007, pp. 113–122.