First Step on Engineering Comprehension

Use of Eye Tracking to understand computer experts thinking

Duc-Loc Huynh and YannGaël Guéhéneuc
Ptidej Team
LaiGLE – Labaratoire de Génie Logiciel Expérimental
GEODES – Group of Open and Distributed
Systems, Experimental Software Engineering
University of Montreal
Montréal, Québec, Canada
[huynhduc, guehene]@iro.umontreal.ca

1 Introduction

The goal of our work is to show the usefulness of Design Patterns with respect to the performance of computer experts or architect when they do their analysis and engineering phases in a software system via class diagrams.

The EyeTraking system allows to record, in real time, all attracting points of movements of a person looking on the screen.

This equipment allows us to have the gain to let the subject to do his task on screen without inferring his work.

While the fusion between the EyeTraking technology and the software engineering are from those beginnings, we will present some related works opening the path of this fusion. And then, we will present the modelisation, how harvested data will be use to verify our premise and also a presentation of our test protocol, at last the conclusion of our observations.

2 Related work

The EyeTracking technology is largely use in visualisation area. Research on human perception versus computer perception and research about interactions between human and computer. We make parallel between those research results and ours to understand the behaviour and the comportment of data harvested.

3 Experiments

3.1 Fixation and Saccade

The EyeTracking equipment allows us to highlight, in the terms of quantification, the fixation and the saccades. Fixations are the interest points, when a subject takes time to look a specific point, area onscreen. Saccades are the movements of fixations, it allow us to determine what level is the insurance or hesitation of the subject. The conclusion of this feeling flows from comparisons and consistency of works on the research area of cognitive psychology. Where the EyeTracking is a standard tool.

3.2 Naïve Approach

Our first premise is that persons know design patterns perform better, while engineering task, than other computer experts that doesn't know it.

To ensure our assumption, we did experiments on classes diagrams about engineering tasks. It consists to ask a subject to do some engineering tasks on classes diagrams representing a portion of a software system. Proposed diagrams have all a common point, they all have at least one design pattern surrounded by others classes. And more, the engineering task must have a link with the design patterns present in the diagram. We will ask to make change on the diagram and check all children of a class. They did a depth-first search. People without design patterns knowledge, on figure 2, will not prowl through the whole diagram if not necessary. Seem to us, that

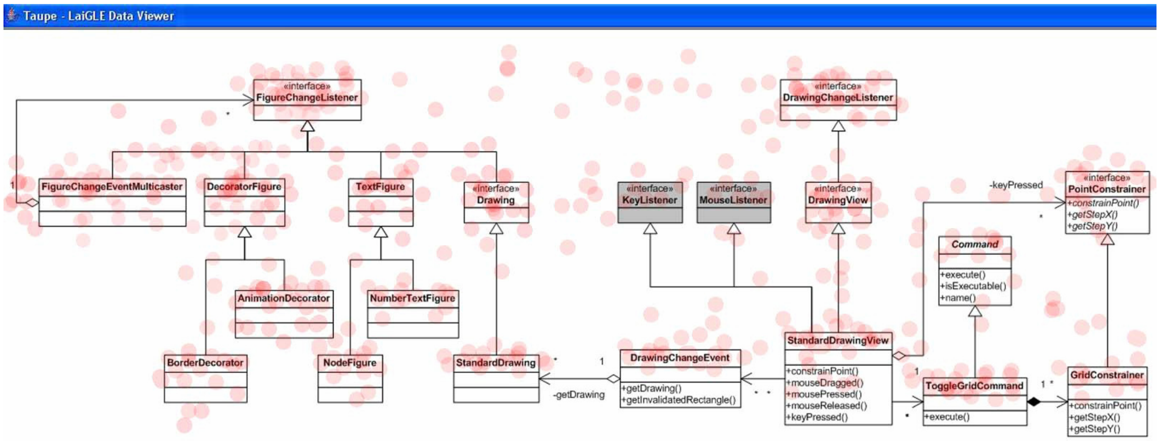

Figure 1 : With design pattern knowledge

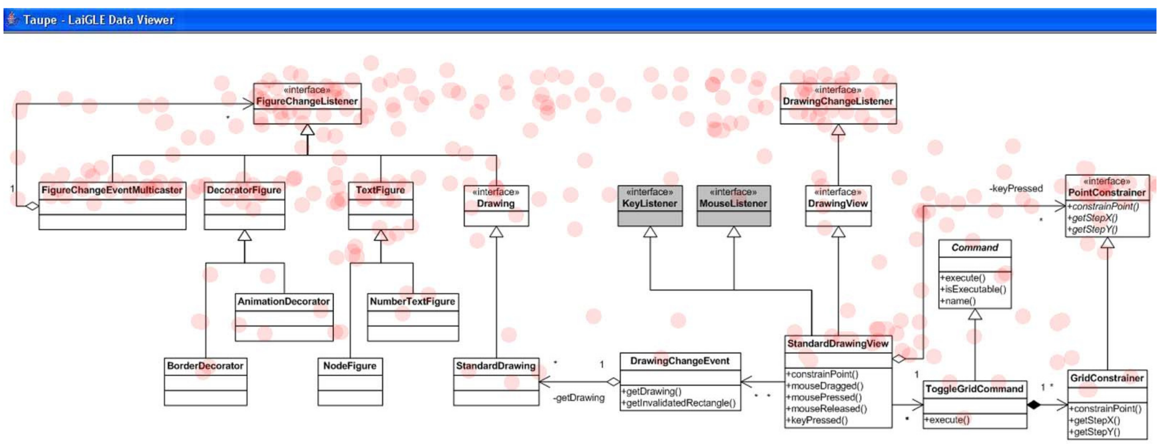

Figure 2 : Without design pattern knowledge

design pattern, to add, remove or modify a feature provide by the pattern. By this way, we will be able to show the usefulness of these structures, by studying the way the subject reach the critical object and considering the time needed.

The experiments are done on two groups. Know and doesn't know design pattern. The engineering task was to add a new figure listener in the system, like a class "circleFigureListener". Only good results are considered. We observe that people with design patterns knowledge, on figure 1, will spend more time on strategic classes like interface classes or abstract classes. Also, they took more time to prowl through the they did a breadth-first search. They are more concern about the name of the object than there structure (interface or abstract classes).

The conclusion of our first experiment was, people without design patterns knowledge perform better in engineering task than people with design pattern. That conclusion invalidates our main premise.

To understand why our assumption was not proved, we investigate again, on our set of subjects to find what they share in common, like the background, the experience, the knowledge of other language ... their feelings... In fact, we make a list of all free variables and try to find a common link between people and their results.

Our conclusion of this investigation was, people in the group that doesn't know design patterns, has more experience about programming and are not much worry if the answer was correct or not. While the test, they were relax and just did what they had to do. Unlike the other group with people that know design pattern, they had less experience about programming and seemed stress by the experiment. Not because they were afraid about the material, but because, they are from our laboratory of software engineering, so they tried to find an inexistent trap or something else. They were more concerned about to give a good or wrong answer than the first group.

So, our first premise failed not because design patterns are an evil thing from hell, but because unconsidered free variables.

3.3 Refined Approach

Looking at various free variables that influence computers experts, we did experiments again. The protocol of the experiment is the same than the naïve approach, that is, we asked to each subject to do some engineering tasks closely link with the design patterns of the diagram. The distinction with the first approach is that we need four groups; two sets of people without design patterns knowledge, but with and without experience, and another two sets of people without design patterns knowledge, with and without experience in programming.

Our new premise is the same than the first, but we consider more in detail our set of subject.

Results are now, more consistent, we can categorize groups; [group A] less efficient, is the group with no knowledge of design patterns and no experience, [group B] second is the group with design patterns knowledge but still not experience, [group C] then the group with experience but no design patterns knowledge, [group D] at last, the most efficient is therefore the with design patterns knowledge and with experience.

We observe that the group A has an erratic way to analyse diagrams, also their average answer or not very efficient. The group B spends more time to analyse understands and makes the engineering task. They try to identify a design pattern before to try to resolve his task. The group C, don't care about design pattern, it do his job in a few time, but don't always know were or how to make the engineering task. The group D, is the more efficient, because it is fast and efficient. It analyse and understand diagrams fast and know where and how to make the change!

4 Conclusion

To conclude, our researches are not like others in computer sciences, we don't study machine, we don't study algorithm, but we study people! And when we speak about people, we speak about a non exact science! That is why our studies are not trivial. With people, we have to consider many factors, many free variables, like experience, the mood of the person at the moment of the experiment... All that information is difficult to harvest, because we have difficulties to define it precisely. To compare something, we have to find a way to quantify it.

Our studies are only at the beginning, we still have a lot of side to find, explore and understand. Only with this experiment, we just scratching the surface, we did not pay attention about the peripheral vision nor the state of mind of subjects.

For now on our new assumption is "Design patterns are use to compensate the lake of background of new computer expert, and help those more experiment to structure their comments and diagrams to allow them to work more effectively in team."

We are currently investigating this new assumption on a larger set of subjects, students of different university and even industrial people.

We believe that our studies will contribute in a valuable contribution about research between computer science and cognitive science.

5 Bibliographies

"Eye Tracking Research & Applications", Duchowski, A., Karn, K. S., and Senders, J. W., Eds., (ETRA) (Palm Beach Gardens, FL, 2000), ACM. URL: http://www.vr.clemson.edu/eyetracking/et-conf/

"Software for Identifying Patterns and Similarities", Julia West, Anne R. Haake, Evelyn P. Rozanski (Rochester Institute of Technology), Keith S. Karn (Xerox Corporation), ETRA, March 27-29 2006