Position Paper

Can the software design activity be quantified?

Jonatan Hernandez,[†1] Hironori Washizaki[†1]
and Yoshiaki Fukazawa[†1]

The software design activity is a creative human activity. very software designer has his own process for doing this activity. ecause of this it is difficult to analyze everyone in the same way. We propose a view of the design activity that allows us to extract quantifiable data from the design process. We have applied our view in the creation of UM class diagrams and we were able to obtain promising results of several aspects of the creation of UM class diagrams, such as which is the most common target object, or how are distributed the actions of a designer.

1. Introduction

Software design is an important activity that always takes place when developing a piece of software. There are always questions about the software to be built: How are we going to divide this problem? How are we going to represent this problem? How are we going to connect all these pieces of the problem? There are many questions that need an answer before we have a complete piece of software.

Answering these kind of questions is the design activity. But how many answers are for these questions? There are as many as there are designers. Every designer has its own way of solving his design problems. There are many solutions, and every problem has its own context. This makes designing a very personal activity.

2. Our proposal

The view we propose is a simplified representation of the design activity. We take into account only individual design activity, without the help of other designers. Our target is the creation of the UML class diagrams, as this kind diagrams are the most used of the UML specification. In Fig. 1 we show the big picture of our analysis process.

The assumption of our proposal is that an important part of the design activity, is related to the creation UML class diagrams. The data generated during this creation can be used to quantify the design activity. To obtain data we log the interaction of the designer with its tool, in this case a UML modeling tool. We have used several of these tools, such as ArgoUML[⋆1], Astah[⋆2], and Enterprise Architect[⋆3]. We capture the events of the UML modeling tool, and then transform the data from events into actions of the user by selecting the relevant events.

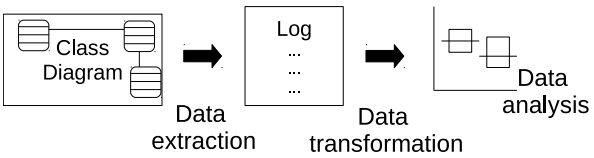

Fig. 1 Data extraction and analysis

Thus the design activity it is represented as a sequence of actions from the user. In Fig. 2 we show the details of these actions and events.

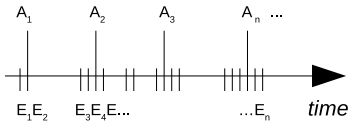

Fig. 2 vents and actions from the user

The final log of the actions after being processed have three parts: 1) The type of action: create, delete, or modify; 2) The target element of the UML diagram: e.g. class, attribute, etc.; and 3) the time stamp.

3. Quantifying the design activity

We have applied our view[2)] and[3)] in two case studies. For example we show two views of the same session by one designer: in Fig. 3 we show a view of the activity with the target elements, e.g. class, attributes, elements, etc. In Fig. 4

†1 Faculty of Fundamental Science and ngineering, Waseda University

⋆1 http: argouml.tigris.org
⋆2 http: astah.net
⋆3 http: www.sparxsystems.com products ea

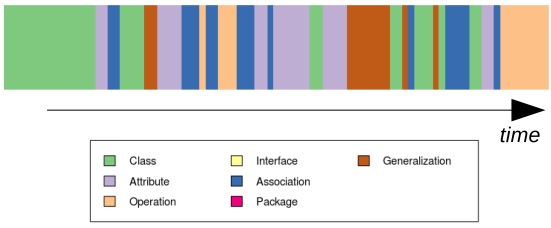

Fig. 3 Session of elements

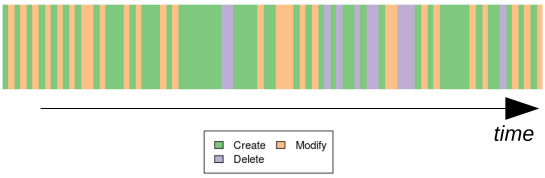

Fig. 4 Session of actions

we show a different view of the same activity focusing only in the actions, e.g. create, delete, and modify.

Using these different views we have been able to identify —in some sessions— strategies such as top-down, bottom-up, and opportunistic[8]. Generally in short sessions it was easier to identify a particular strategy. In longer sessions, such as the ones that extends for several weeks, there was not a clear strategy used. Also we were able to identify the most common element, for example, creation and modification of relationships was very common in some sessions. Using this information we can calculate some ratios, e.g. $\frac{TotalNumberOfCreations}{TotalNumberOfDeletions}$, which can give us a hint of how many changes of the design had the developer.

3.1 Related studies

The design activity usually has been researched using verbal protocols[5),6),7)] and[1)]. Verbal protocols take long time to analyze, and are more difficult to quantify.

Interaction data has been used to analyze other aspects of the software development, for example, program understanding[4)]. In our approach we are using this data to analyze the design activity. Nevertheless, these two aspects can be complementary.

4. Conclusions and future work

Using our approach we were able to obtain some quantifiable data from the design activity, such as most common element or action. In some cases we identified strategies using the distribution of the actions of the user. We have obtained this information in the design activity, and our next step is to relate these results to the quality of the design. Evaluating the design is also a difficult problem because there is not one and best solution. Every design problem usually has many different solutions. There also has been proposals to evaluate the design, such as UML metrics. But still there is no universal agreement with these metrics. We believe that our proposal offers a faster and more systematic process towards quantifying the design activity. Our future work is to offer the designer a tool for an easy way to evaluate its past work, to identify good and bad patterns in his own designing in activity, thus allowing him to improve his own design process. Our goal is always to provide information about his design activity in an unobtrusive and easy to measure way.

References

1) F.Détienne. Design strategies and knowledge in object-oriented programming: effects of experience. *Human–Computer Interaction*, 10(2-3):129–169, 1995.
2) Jonatan Hernandez, Hironori Washizaki, and Yoshiaki Fukazawa. Macro and micro patterns in the software design activity. In *JCKBSE*, pages 298–302, 2012.
3) Jonatan Hernandez, Hironori Washizaki, and Yoshiaki Fukazawa. Using an automatic collection method to identify patterns during design activity. In *Knowledge-Based Software Engineering*, pages 491–504. Springer, 2014.
4) Roberto Minelli, Andrea Mocci, Michele Lanza, and Takashi Kobayashi. Quantifying program comprehension with interaction data. In *Quality Software (QSIC), 2014 14th International Conference on*, pages 276–285. IEEE, 2014.
5) V.Popovic and B.J. Kraal. Expertise in software design: Novice and expert models. *Proceedings of Studying Professional Software Design*, 2010.
6) S.Sonnentag. Expertise in professional software design: A process study. *Journal of Applied Psychology*, 83(5):703, 1998.
7) W. Visser. Designers' activities examined at three levels: organization, strategies and problem-solving processes. *Knowledge-Based Systems*, 5(1):92–104, 1992.
8) W.Visser and J.M. Hoc. Expert software design strategies. In *Psychology of Programming*, pages 235–249. Academic Press, 1990.