Quantifying programmers' mental workload during program comprehension based on cerebral blood flow measurement: A controlled experiment

Conference Paper · June 2014

DOI: 10.1145/2591062.2591098

CITATIONS
48

READS
328

6 authors, including:

Takao Nakagawa
Nara Institute of Science and Technology
6 PUBLICATIONS **62** CITATIONS

SEE PROFILE

Yasutaka Kamei
Kyushu University
137 PUBLICATIONS **2,819** CITATIONS

SEE PROFILE

Hidetake Uwano
Nara Institute of Science and Technology
41 PUBLICATIONS **385** CITATIONS

SEE PROFILE

Akito Monden
Okayama University
207 PUBLICATIONS **2,516** CITATIONS

SEE PROFILE

Quantifying Programmers' Mental Workload during Program Comprehension Based on Cerebral Blood Flow Measurement: A Controlled Experiment

Takao Nakagawa
Nara Institute of Science and Technology
Nara, Japan
takao-n@is.naist.jp

Yasutaka Kamei
Kyushu University
Fukuoka, Japan
kamei@ait.kyushu-u.ac.jp

Hidetake Uwano
Nara National College of Technology
Nara, Japan
uwano@info.nara-k.ac.jp

Akito Monden
Nara Institute of Science and Technology
Nara, Japan
akito-m@is.naist.jp

Kenichi Matsumoto
Nara Institute of Science and Technology
Nara, Japan
matumoto@is.naist.jp

Daniel M. German
University of Victoria
BC, Canada
dmg@uvic.ca

ABSTRACT

Program comprehension is a fundamental activity in software development that cannot be easily measured, as it is performed inside the human brain. Using a wearable Near Infra-red Spectroscopy (NIRS) device to measure cerebral blood flow, this paper tries to answer the question: Can the measurement of brain blood-flow quantify programmers' mental workload during program comprehension activities? We performed a controlled experiment with 10 subjects; 8 of them showed high cerebral blood flow while understanding strongly obfuscated programs (requiring high mental workload). This suggests the possibility of using NIRS to measure the mental workload of a person during software development activities.

Categories and Subject Descriptors

D.2.5 [**Software Engineering**]: Testing and Debugging; D.2.8 [**Software Engineering**]: Metrics

General Terms

Measurement

Keywords

Program comprehension, mental workload, cerebral blood flow measurement

1. INTRODUCTION

Program comprehension is a fundamental activity required in today's software development processes such as coding, code review, debugging, code reuse and maintenance. Its measurement is difficult as it is a mental (cognitive) process performed inside the human brain.

To measure such mental activities, recent neuroscience and cognitive science studies try to directly measure brain activity using sensors such as EEG, fMRI and NIRS [1]. Also in the software engineering domain, Siegmund *et al.* [6] pointed out (at the FSE2012 New Idea Track) the necessity of analysis of brain activities in program comprehension. They proposed an experiment design using fMRI (functional magnetic resonance imaging) measurement; however, no result has been reported so far, and research progress in this area is strongly demanded.

In this paper, we focus on the measurement of programmers' mental workload during program comprehension to answer the question: *Can brain measurement quantify programmers' metal workload in program comprehension?* If the measurement could identify programmer's very high workload, which may imply the work is beyond his/her capacity, timely help by an expert or a manager needs to be considered.

This paper presents an experiment design using a wearable NIRS (Near Infra-red Spectroscopy) to observe the cerebral blood flow of the prefrontal cortex (PFC), which has been considered to govern planning of complex cognitive behaviour and decision making [7]; therefore, we believe PFC activity is vital in program comprehension. In the experiment, we asked each subject to perform two tasks: 1) non obfuscated C programs, and 2) strongly obfuscated C programs that should require higher mental workload of PFC.

As a result of a controlled experiment with 10 graduate students in computer science, 8 students showed higher cerebral blood flow during reading of obfuscated versions. This suggests the possibility of measuring mental workload in program comprehension using NIRS, while we also came up with several improvements needed in future experiments to clarify the feasibility and limitation of our approach.

2. RELATED WORK

To understand the human aspect of program comprehension (such as understanding level, developers' behaviour,

ICSE Companion'14, May 31 – June 7, 2014, Hyderabad, India
Copyright 2014 ACM 978-1-4503-2768-8/14/05...$15.00
http://dx.doi.org/10.1145/2591062.2591098

```
 1: int func(int ***A, int N, int M, int L){
 2:   int i,j,k;
 3:   int p;
 4:   p=A[0][0][0];(1)
 5:   for(i = 0; i < N; i++){
 6:     for(j = 0; j < M; j++){
 7:       for(k = 0; k < L; k++){
 8:         if(A[i][j][k] < p) p = A[i][j][k];(2)
 9:       }
10:     }
11:   }
12:   return p;(3)
13: }
```

```
preconditions:
 A[0][0][0] = 97, A[0][0][1] = 48
 A[0][1][0] = 52, A[0][1][1] = 71
 A[1][0][0] = 17, A[1][0][1] = 64
 A[1][1][0] = 11, A[1][1][1] = 32
 A[2][0][0] = 20, A[2][0][1] = 22
 A[2][1][0] = 48, A[2][1][1] = 86

 N = 3, M = 2, L = 2
```

Figure 1: The sample of the program text and preconditions given to subjects

comprehension strategy), researchers have used indirect measurement such as interview, questionnaire or 'think-aloud' protocol (let subjects speak their content of thinking during experiment [2]).

Parnin [5] analysed both short-term and long-term memory retention of developers who are working in parallel programming tasks from the viewpoint of cognitive neuroscience. Also, Nakamura *et al.* [4] focused on the remembering, recalling and forgetting of variables in source code to develop a model of program comprehension. Their experiment showed that the time required to complete a comprehension task well matched the difficulty of recalling a variable.

Siegmund *et al.* [6] applied a neuroscientific approach for the program comprehension process and proposed an experiment plan for identifying cortical regions related to the program comprehension in the FSE'12. They pointed out the necessity of analysis of brain activities to answer the question such as *What distinguishes good programmers from bad programmers?* or *What makes a good programmer?* However, they only mentioned about their interim report of progress of the experiment and its design, thus, they have not published results yet. Siegmund *et al.* predict that prefrontal cortex (related to the memory operation or complex intellectual activity) will be activated when developers try to understand the program.

These studies suggest that cognitive process related to the human memory exist during program comprehension. However, there are no experimental results about brain activation during program comprehension, or programming. Thus, little is known about how actually brain works during program comprehension tasks.

3. EXPERIMENT

3.1 Subjects

Ten students of Nara Institute of Science and Technology participated in the experiment as subjects. All subjects are male, 22-26 years old, and have experience using C-language for at least 3 years.

3.2 Programs and Assignment

Six programs (three algorithms and two difficulty levels) of 17-32 lines of code, all written in C language, are used. 3 algorithms are searching a keyword, calculating total values, and seeking the maximum value in an array. One of the algorithms is used in an exercise task before a main experiment task.

To prepare two difficulty-level programs for each algorithm, we use an obfuscation technique to make a 'hard' version program from an 'easy' (non-obfuscated) version. We used a loop obfuscation so that loop counters and sentinel values are updated frequently and irregularly without changing the functionality of a program [3]. Figure 1 shows a non-obfuscated program that seeks the minimum value in an array.

Two different level/functionality tasks are assigned to 1 subject. To reduce learning effect, half of the subjects perform the easy task first, and the others perform the hard one first. All subjects perform an exercise task before the main experimental task. The exercise task has two complexity levels similar to the main experimental task.

3.3 Task

To standardize the strategy of program comprehension among all subjects, they read and simulate the execution of a program using *mental simulation* strategy (also known as hand simulation). It is one of the bottom-up program comprehension strategy to simulate the program's execution process (e.g.,, control flow and variable assignment). To properly trace the program, subjects have to remember the current position of loop-flow and variables name/value in their short-term memory during mental simulation.

During the mental simulation, when the subjects reach to a checkpoint marked in the program like (1) of Line 4 and (2) of Line 8 in Figure 1, they write down the value of each variable at the checkpoint to an answer sheet. After writing down these values, they raise their hand. An experimenter (one of the authors) checks the values on their answer sheet and tell the subjects whether or not the answer is correct.

If their answer is correct, the subjects continue to perform the comprehension from the current checkpoint to the

Figure 2: WOT-200(Hitachi medical Co.)

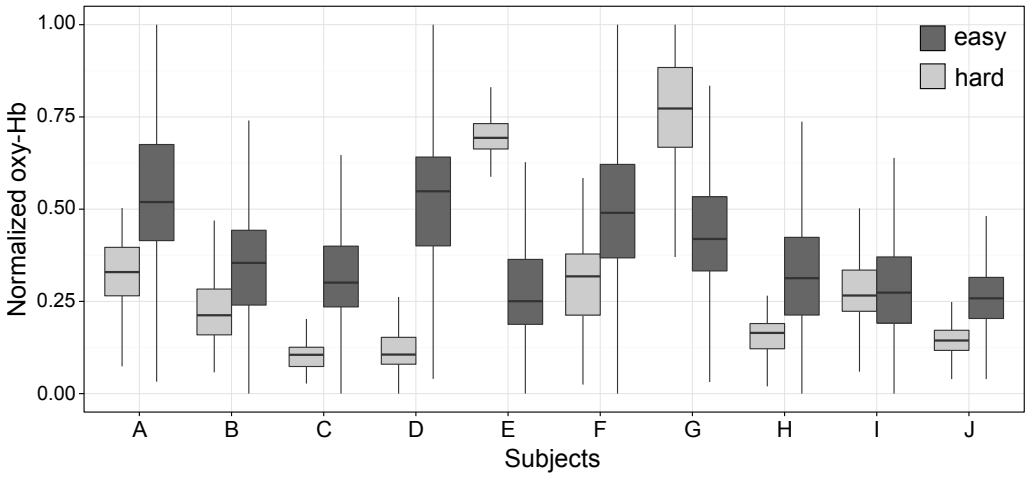

Figure 3: Distribution of normalized oxy-Hb

next one. If not, they go back to a previous checkpoint and restart the comprehension task. When they correctly answer the last checkpoint marked at the *return* statement in the program (like (3) of Line 12 in Figure 1), they have completed the task.

3.4 Equipment and environment

We use the NIRS (Wearable Hikari Topography WOT-200, made by HITACHI MEDICO). Figure 2 shows the appearance of the device.

NIRS assumes that higher brain activity requires more oxygen to be transported by the blood flow. Therefore, to quantify the brain activity, NIRS measures the amount of oxygenated haemoglobin (oxy-Hb) in the cerebral blood flow.

We consider this device suitable for measurement of program comprehension under the condition similar to the real environment. Because it is lightweight, can be easily set on the subject's head, and does not keep subjects' body in a fixed position during an experiment in contrast to the fMRI, MEG and PET that has a finer and wider spatial resolution than NIRS.

Subjects sit down during the experiment. Experiments are performed in a quiet room where only the subject and the experimenter are.

To avoid the noise in the measurements, an experimenter (one of the authors) asked the subjects not to lower and raise their head. The subjects adjust the position and the height of the chair before beginning of the experiment. Program text and an answer sheet are put in front of the subjects.

3.5 Metrics

Since the amount of oxy-Hb can be measured only as a *relative* value from the beginning of the measurement, we used a normalized value based on the following equation:

$$Normalized\ oxyHb = \frac{oxyHb - min(s)}{max(s) - min(s)}$$

where *max(s)* and *min(s)* are the maximum and minimum value through all tasks of each subject *s*. The range of the normalized oxy-Hb is [0,1]. We measured the normalized oxy-Hb every 200ms.

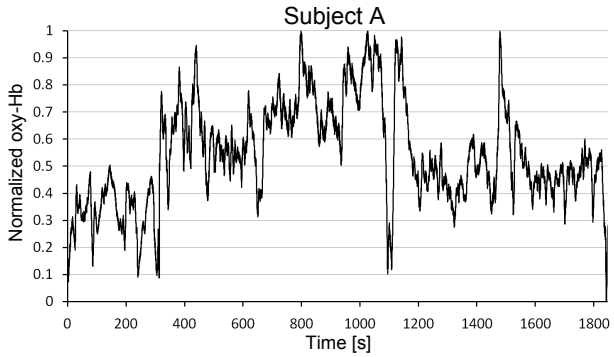

Figure 4: Chronological changes of brain activation

4. RESULTS AND DISCUSSION

Figure 3 shows the distribution of normalized oxy-Hb of each subject/task. Labels A to J represents each subject (the left box shows the distribution of 'easy', and right one shows the 'hard'). The y-axis corresponds to the normalized oxy-Hb (i.e., how much the brain works actively).

We found that the normalized oxy-Hb of hard tasks is larger than easy tasks among all subjects except E and G. This result suggests that the complexity of the program induces the activation of the prefrontal cortex, thus, we consider that mental workload could be quantified using cerebral blood flow measurement.

Another finding is that the variance of normalized oxy-Hb of hard tasks is larger than easy tasks among all subjects except E. This suggests that even in a hard task, mental workload is often very low. Figure 4 shows the time-course changes of subject A's data during performing 'hard' task, which indicates the amount of oxy-Hb continues to change throughout the experiment. Therefore, additional measurement, such as PC operation history and eye-gaze tracking, is needed in future study to observe subjects' external behaviours.

The result also indicates that some subjects (E and G in our case) may show the counter-trend tendency to others. This could happen by several reasons, e.g., 1) measurement

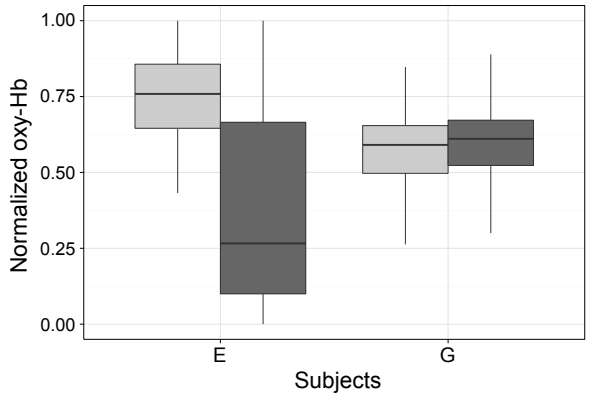

Figure 5: Normalized oxy-Hb in an exercise task

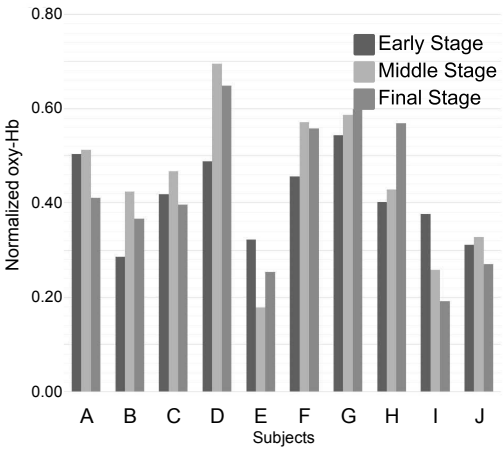

Figure 6: Chronological changes of brain activation

error (the sensor may not fit well to some subjects' forehead), 2) subject's skill (high skill subjects may not feel any difficulty in hard tasks), 3) subject's natural property (some subjects may not require high oxy-Hb in mental simulation), etc.

For further analysis, Figure 5 shows the distribution of normalized oxy-Hb of E and G in the exercise tasks. Interestingly, subject E showed the same counter-trend reaction (oxy-Hb during 'easy' higher than 'hard') in the exercise task (Figure 5, left graph). Further experiments are required in future to analyse why and how often this would happen. At least, an interview to subjects after the experiment is needed to clarify if all subjects felt the 'hard' task more difficult than the 'easy' task.

Figure 6 shows the result of time-series analysis. We equally divided the task completion time into three parts, the early stage, the middle stage, and the final stage. Each bar in Figure 6 shows the median of the normalized oxy-Hb of each stage.

Figure 6 indicates that normalized oxy-Hb is higher in the middle stage than the early stage (8 out of 10 subjects) and higher in the middle stage than the final stage (7 out of 10 subjects). This may happened because most wrong answers occurred in the middle stage, which implies that high workload is required to correct the answers. This result suggests the possibility to quantify the time-course change of mental workload using NIRS.

Threats to validity: To generalize our result, we need to consider top-down comprehension strategy and its difficulty because our experiment lets subjects use only bottom-up strategy (i.e., mental simulation). However, we believe that our method can be applied to another strategy if difficulty levels of the program are well defined, because PFC has a strong relation with complicated intellectual activity and the top-down strategy is as complicated as bottom-up strategy.

5. CONCLUSIONS

In this paper, we aimed to investigate whether or not developers' workload can be quantified using cerebral blood flow measurement of the prefrontal cortex. In our experiment, we measured the amount of oxygenated haemoglobin (oxy-Hb) during comprehension of two different types of programs, 'hard' (high complexity) and 'easy' (low complexity). The result showed the tendency that oxy-Hb becomes higher in 'hard' programs than 'easy' programs, which suggests the possibility of measuring mental workload by oxy-Hb.

In the future, we are planning to compare program comprehension tasks with other cognitive tasks such as reading a natural language text or doing a mathematical calculation. Also, we are planning to conduct very-easy/very-hard tasks as baseline tasks, e.g., doing (nothing) with eye-closed as very-easy, and doing extremely-difficult mathematical calculation as very-hard. We also plan to use other measurement sources, e.g., history of PC operations, eye-tracking and interview to subjects.

6. REFERENCES

[1] R. Cabeza and L. Nyberg. Imaging cognition II: An empirical review of 275 PET and fMRI studies. *Journal of cognitive neuroscience*, 12(1):1–47, 2000.

[2] K. A. Ericsson and H. A. Simon. Verbal reports as data. *Psychological review*, 87(3):215, 1980.

[3] A. Monden, Y. Takada, and K. Torii. Method for scrambling programs containing loops. *IEICE Trans. on Information and Systems*, 80(7):644–652, 1997. (in Japanese).

[4] M. Nakamura, A. Monden, T. Itoh, K. Matsumoto, Y. Kanzaki, and H. Satoh. Queue-based cost evaluation of mental simulation process in program comprehension. In *Proc. of 9th IEEE International Software Metrics Symposium (METRICS'03)*, pages 351–360, 2003.

[5] C. Parnin. A cognitive neuroscience perspective on memory for programming tasks. In *Proc. of 22nd Annual Meeting of the Psychology of Programming Interest Group (PPIG)*, 2010.

[6] J. Siegmund, A. Brechmann, S. Apel, C. Kästner, J. Liebig, T. Leich, and G. Saake. Toward measuring program comprehension with functional magnetic resonance imaging. In *Proc. of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering*, (FSE '12), pages 24:1–24:4, 2012.

[7] Y. Yang and A. Raine. Prefrontal structural and functional brain imaging findings in antisocial, violent, and psychopathic individuals: A meta-analysis. *Psychiatry Research: Neuroimaging*, 174(2):81 – 88, 2009.