Toward Conjoint Analysis of Simultaneous Eye-Tracking and fMRI Data for Program-Comprehension Studies

Norman Peitek
Leibniz Institute for Neurobiology
Magdeburg, Germany

Janet Siegmund
University of Passau
Passau, Germany

Chris Parnin
NC State University
Raleigh, North Carolina, USA

Sven Apel
University of Passau
Passau, Germany

André Brechmann
Leibniz Institute for Neurobiology
Magdeburg, Germany

ABSTRACT

After decades of research, there is still no comprehensive, validated model of program comprehension. Recently, researchers have been applying psycho-physiological measures to expand our understanding of program comprehension. In this position paper, we argue that measuring program comprehension simultaneously with functional magnetic resonance imaging (fMRI) and eye tracking is promising. However, due to the different nature of both measures in terms of response delay and temporal resolution, we need to develop suitable tools. We describe the challenges of conjoint analysis of fMRI and eye-tracking data, and we also outline possible solution strategies.

CCS CONCEPTS

• **Human-centered computing → HCI design and evaluation methods**; **Empirical studies in HCI**;

KEYWORDS

program comprehension, functional magnetic resonance imaging, eye tracking

ACM Reference Format:
Norman Peitek, Janet Siegmund, Chris Parnin, Sven Apel, and André Brechmann. 2018. Toward Conjoint Analysis of Simultaneous Eye-Tracking and fMRI Data for Program-Comprehension Studies. In *EMIS '18: Symposium on Eye Movements in Programming, June 14–17, 2018, Warsaw, Poland*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3216723.3216725

1 INTRODUCTION

Understanding program comprehension is important, as programmers spend most of their time comprehending existing source code [LaToza et al. 2006; Standish 1984; Tiarks 2011]. For decades, researchers have been trying to understand how programmers comprehend code, which is challenging due to the complexity of the underlying cognitive processes. Program comprehension involves many cognitive subprocesses, including language comprehension, attention, and problem solving [Siegmund et al. 2014, 2017]. Past research observed program comprehension with think-aloud protocols, interviews, and comprehension summaries. However, these conventional methods are limited in accurately and holistically capturing internal cognitive subprocesses of program comprehension [Siegmund 2016]. Hence, researchers have been applying psycho-physiological measures to observe program comprehension from a new perspective [Crk et al. 2015; Floyd et al. 2017; Fritz et al. 2014; Nakagawa et al. 2014; Sharif and Maletic 2010; Siegmund et al. 2014].

1.1 Eye Tracking

One wide-spread measurement tool for observing program comprehension is eye tracking [Bednarik and Tukiainen 2006; Busjahn et al. 2015; Sharif and Maletic 2010; Yusuf et al. 2007]. Eye tracking offers insight into the visuo-spatial attention of programmers [Holmqvist et al. 2011]. For example, Sharif and Maletic studied the difference in program comprehension between camelCase and under_score identifier styles [Sharif and Maletic 2010]. Busjahn et al. distinguished reading styles between novices and experts, also using an eye tracker [Busjahn et al. 2015]. So, eye tracking is an effective measure to gain insight into the behavior and strategy during program comprehension.

However, research with eye tracking often focuses on eye gaze (i.e., fixations and saccades), which allows one to make only limited claims on higher level cognitive processes (e.g., language comprehension, decision making, memory). While other eye-tracking properties, such as pupil dilation or spontaneous blink rates, have been shown to be an effective measure of cognitive load [Beatty and Lucero-Wagoner 2000; Graham 1992], these measures have not progressed beyond the planning stage in program-comprehension studies [Behroozi et al. 2018; Ford et al. 2015; Nolan et al. 2015].

To increase the predictive power of eye-tracking experiments, researchers have combined eye tracking with dedicated measures to capture programmers' cognitive load. For example, Fakhoury et al. used simultaneously eye tracking and functional near-infrared spectroscopy (fNIRS) to observe the effect of the quality of identifier names on programmers' cognitive load [Fakhoury et al. 2018]. In the same vein, there have been studies with simultaneous eye tracking and electroencephalography (EEG) to predict task difficulty and programmer expertise [Fritz et al. 2014; Lee et al. 2017]. These studies illustrate how a combination of methods strengthens an

EMIS '18, June 14–17, 2018, Warsaw, Poland

ACM ISBN 978-1-4503-5792-0/18/06…$15.00
https://doi.org/10.1145/3216723.3216725

experiment's predictive power to understand program comprehension. We previously proposed a study with simultaneous functional magnetic resonance imaging (fMRI) and eye tracking [Peitek et al. 2017]. Overall, the combination of behavior-capturing eye tracking with a cognitive-load capturing measure (e.g., EEG, fNIRS, fMRI) appears to be promising.

1.2 fMRI

In our work, we study fMRI as a way to understand the underlying cognitive subprocesses of program comprehension. fMRI allows us to observe brain activation and thus identify neural correlates of cognitive processes [Gazzaniga et al. 2013].

In the first fMRI study of program comprehension, we have found five dedicated activated brain areas during bottom-up comprehension [Siegmund et al. 2012, 2014]. In a follow-up study, we collected empirical evidence showing a difference in neural efficiency between bottom-up and top-down comprehension [Siegmund et al. 2017]. However, we failed to observe an expected effect of rich identifier names (i.e., *beacons* [Brooks 1983]), which may be due to the cognitive complexity of program comprehension and limitations of the fMRI analysis, in which we averaged activation across the whole comprehension process without considering the phases specifically related to the beacons. Program comprehension consists of multiple phases. For example, when programmers understand source code, they have to decide on how to solve presented task (strategy finding). Next, they have to extract and integrate the meaning of individual code lines (semantic chunking), mentally execute code, and eventually conclude their understanding. Throughout these tasks, the cognitive subprocesses vary in their intensities. While fMRI can capture these differences, we currently lack the knowledge of how to accurately determine the timing of the phases to identify brain activation more specifically. Thus, we integrated simultaneous eye tracking during our fMRI experiments with the goal to identify the comprehension phases based on the observed visual attention.

1.3 Simultaneous fMRI and Eye Tracking

Simultaneous measurement with fMRI and eye tracking may offer detailed insight toward a more holistic understanding of program comprehension [Peitek et al. 2017]. Integrating simultaneous and precise eye tracking into our fMRI study framework is challenging on a technical level. We resolved the method synchronization details and successfully conducted a full study, which revealed difficulties regarding the quality of the eye-tracking data (cf. Section 2).

However, the simultaneous recording of eye gaze and brain activation is only half the problem: The next challenge is to actually apply fruitful analysis to the two data streams. They could be viewed independently (e.g., observe top-down and bottom-up comprehension and separately compare brain activation strength and fixation count). However, in our view, the true value lies in integrating observations from the two separate data streams (i.e., a conjoint analysis of both data streams).

Duraes et al. studied debugging with simultaneous measurement via fMRI and eye tracking, but do not report how successful and precise the recorded eye-tracking data were in their experiment. Furthermore, they did not appear to use the eye-movement data [Duraes et al. 2016].

An analysis of simultaneous fMRI and eye tracking is challenging, because the two have fundamentally distinct characteristics. They are not just different in temporal resolution, but fMRI relies on the haemodynamic response, which means observed brain-activation data are delayed by around five seconds [Chance et al. 1993]. A conjoint analysis of instant eye-tracking data and delayed fMRI data will need to take this into account.

1.4 Multi-Modal Experiments

In our view, the future of program-comprehension research will exemplify this struggle of creating valuable analysis as we are moving toward multi-modality. While this paper focuses on the combination of fMRI and eye-movement data, we are investigating further expansions of our fMRI study framework. Specifically, we are experimenting with recording pupil dilation, spontaneous blink rates, and physiological data (i.e., heart rate, respiration, and electrodermal activity), which also acts as an indicator of cognitive load [Boucsein 2012]. Thus, in the upcoming future, we will have numerous characteristically different measures observing the same cognitive process.

In the next section, we describe the conducted experiment. In Section 3, we outline possible conjoint analyses of simultaneous fMRI and eye-tracking data. We argue for the need of a proper tool support in Section 4. We conclude the paper in Section 5.

2 EXPERIMENT

In a pilot study with 22 student participants, we investigated observing program comprehension with simultaneous fMRI and eye tracking. For this purpose, we replicated our previous fMRI study on contrasting bottom-up and top-down comprehension [Siegmund et al. 2017]. The study was conducted on a 3-Tesla scanner,[1] equipped with a 32-channel head coil at the Leibniz Institute for Neurobiology in Magdeburg, Germany. We used an MRI-compatible EyeLink 1000[2] eye-tracker for simultaneous measurement of eye movements. The EyeLink eye tracker offers 1000 Hz temporal resolution, <0.5° average accuracy, and 0.01° root mean square (RMS).

The eye tracker collected eye gazes, events (i.e., fixations, blinks, saccades), and pupil dilation. Details on participant demographics, method synchronization and analysis procedure can be found on the project's Web site.[3]

2.1 Eye-Tracking Quality

A preliminary analysis of the eye-tracking data that we obtained revealed some problems. In short, a major issue is the low success rate of eye tracking (only 8 out of 22 participants resulted in an eye-tracking dataset with, at least, 85% of recorded frames). For successful participants, the spatial error is around 20 - 60 pixels, which results at a used line height of 40 pixels in an uncertainty regarding fixation lines. We observed a minimal drift, but in contrast to the general spatial error, it is not a factor. Overall, the pilot study demonstrated that the simultaneous measurement and individual evaluation of each data stream is possible, but we still need to refine

[1] Philips Achieva dStream, Best, The Netherlands
[2] SR Research Ltd, Ottawa, Ontario, Canada, http://www.sr-research.com
[3] https://github.com/brains-on-code/simultaneous-fmri-and-eyetracking

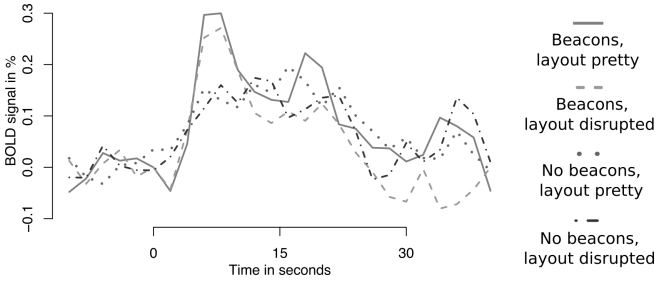

Figure 1: BOLD Response in BA21 from Our Top-Down Comprehension Study [Siegmund et al. 2017]

our setup. For complete fMRI and eye-tracking datasets, we can analyze a comprehensive perspective of participants comprehending code.

3 STRATEGIES FOR DATA ANALYSIS

As far as we are aware of, there are no established procedures to conjointly analyze simultaneous fMRI and eye-tracking data. In this section, we present three ideas on how we may extract knowledge from simultaneous fMRI and eye-tracking data.

3.1 Hypothesis Generation

We can use the two data streams to make observations of a programmer's behavior and generate new hypotheses. Investigating programmers individually with fMRI or eye tracking already offers an interesting perspective on programmers' minds. If we are able to simultaneously explore both data sets, for example, by a real-time video replay, we may be able to generate new hypotheses of program comprehension.

An intuitive exploration of fMRI and eye tracking (and in the future possibly with behavior data, pupil dilation, and physiological data) would allow us to delve into many aspects of program comprehension.

3.2 Hypothesis Testing

A further possibility is to analyze the data to test specific hypotheses and cover another step in the scientific method. This is critical, as each measure offers unique insights into program comprehension.

For example, we found no significant effect of using rich identifier names in our study on top-down comprehension when analyzing the averaged brain activation strength across the entire 30 second task length [Siegmund et al. 2017]. However, Figure 1 illustrates a stronger activation in a language-processing brain area between 5 and 10 seconds after task onset, which fits the typical delay of the haemodynamic response [Chance et al. 1993]. Our hypothesis is that, at the beginning of presenting source code, programmers fixate on a meaningful identifier and recall a matching programming plan. However, since we cannot observe the exact participant behavior in the fMRI scanner, this is speculation. With simultaneous fMRI and eye tracking, we could objectively assess this hypothesis. We can evaluate whether programmers indeed fixate on a rich identifier, which then triggers the increase on brain activation.

Generally, we may detect specific events with eye tracking indicating particular behavior (e.g., mentally executing a loop) and use that as starting point for the resulting haemodynamic response of brain activation. Such individual analysis would allow us to answer our question on the effect of rich identifier names. Similarly, other analyses of particular phenomena are also possible.

3.3 Informed fMRI Analysis

In the long term, we have the vision of an eye-tracking-informed fMRI analysis for studies of program comprehension. Currently, the entire task of 30 to 60 seconds is viewed as one black box of program comprehension. However, it actually consists of many smaller phases and cognitive subprocesses with varying intensity. In such eye-tracking-informed fMRI analysis, we would like to use eye tracking to gather information on the programmers' behavior and then feed it into a general linear model (GLM) analysis of the fMRI data. This way, instead of having one large black box of program comprehension, fMRI data analysis could be more fine-grained and thus allow us to separately evaluate phases of, for example, identifier recognition, loop execution, and result computation. Such detailed analysis may help us to understand the brain activation for each phase in more detail, and in the long-term, make fMRI experiments more valuable to our community.

4 PROPOSED TOOL SUPPORT

After presenting our vision for multi-modal experiments and, more specifically, our ideas for conjoint analyses of fMRI and eye-tracking data, we would like to stress a clear-cut need for proper tool support for each of the three analyses. Ogama is an established eye-tracking analysis tool [Voßkühler et al. 2008], which provides capabilities for replaying recorded data, heat maps, area-of-interest (AOI) analysis, and statistical analysis (e.g., fixation counts, saccade lengths, regressions). We see the need for a dedicated open-source tool that similarly sets the base for simultaneous fMRI and eye-tracking experiments and, in the long term, for multi-modal experiments of program comprehension.

This may be implemented as an extension to an existing tool. For example, we may extend Ogama and add the necessary brain-related features and customize the existing eye-tracking functions for our needs. Alternatively, we could extend an fMRI analysis tool, such as BrainVoyager, with eye-tracking functionalities. Finally, the tool could be a dedicated implementation. In either case, we see the need for two different modes, data exploration and data analysis.

4.1 Data Exploration

To support hypothesis generation based on exploring observations of fMRI and eye tracking, the envisioned tool needs to provide a compelling exploration view. To implement such a view, the tool should be able to import the brain activation data, either as raw data or pre-processed, and the eye-tracking data. The tool would need to take the haemodynamic response, which delays the brain activation by several seconds, into account. We cannot directly map the instant eye-tracking observation to the delayed brain activation.

Moreover, the exploration view should be configurable and provide different views. For example, on the eye-tracking side, it may offer a scanpath or fixation view. On the brain-activation side, it may

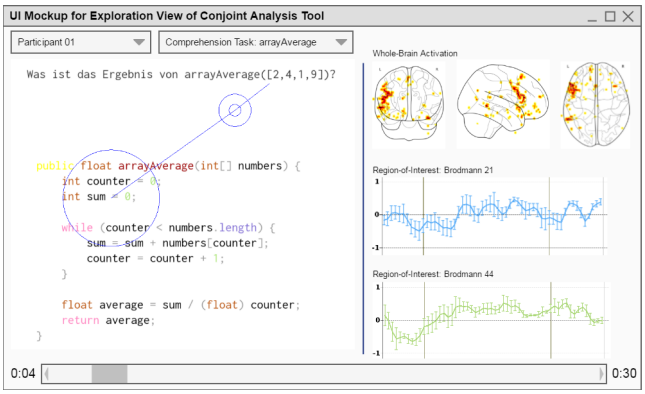

Figure 2: Mockup of Tool UI for Data Exploration

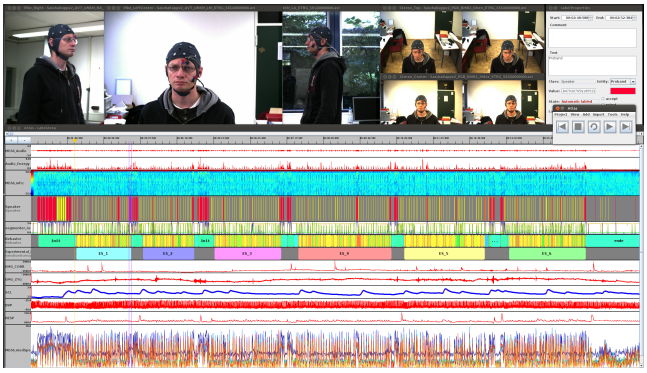

Figure 3: Screen of Multi-Modal Annotation Tool ATLAS [Meudt et al. 2012]

offer a whole-brain activation view or a focus on specific regions-of-interest. We created a mock-up for our vision in Figure 2, which presents eye-tracking replay (left side), whole-brain activation (top-right), and a selected subset of brain areas (bottom-right).

While we focused in this paper on eye-tracking and brain-activation data, future studies will move toward multi-modal experiments. Currently, we are experimenting with numerous other data streams: pupil dilation, spontaneous blink rates, and physiological data (heart rate, respiration, electrodermal activity). We potentially may integrate simultaneous EEG as well. Thus, in the long-term, the tool should be further generalized beyond fMRI and eye tracking. This way, it would be flexible and support any multi-modal view of the data, depending on the collected data set.

4.2 Data Analysis

The second mode of the tool should cover hypotheses testing and informed fMRI analysis. To support these analyses, we split the requirements in two parts: (manual or automated) data annotation and informed fMRI analysis.

Manual Data Annotation. To conduct an eye-tracking-informed fMRI data analysis, we would need to annotate eye-tracking events, such as a fixation on a rich identifier or mental loop execution. In a first stage, we could manually detect eye-tracking events. The proposed tool should allow us to manually annotate data streams. We envision a UI similar to ATLAS, which is a tool for annotating multi-modal data streams of human-computer interaction experiments [Meudt et al. 2012]. Figure 3 shows ATLAS with multiple data streams. Initially, the tool may only support fMRI and eye tracking, but eventually could be extended to support multi-modal annotation (e.g., behavioral or physiological data).

Automated Eye-Tracking Event Detection. The manual detection of events in the first stage introduces human error and inconsistencies to the data annotation. Thus, in the long term, we would prefer to automatically detect eye-tracking events. We would need to describe the event criteria in a flexible manner and store them in a database. Similarly, much like Ogama allows researchers to define and store AOIs, we would need to describe complex eye-tracking events (e.g., fixation for at least half a second on a rich-identifier AOI). As our current experiments do not permit scrolling, we can use static AOIs. Once we extent our studies to allow scrolling, and thus the code display is dynamic, we may need to implement an automated detection of AOIs [Barik et al. 2017].

Informed fMRI Analysis. An important feature of the envisioned tool would be connecting the annotated eye-tracking data to an fMRI data analysis tool. For example, the tool could feed the insights as a parameter input into *nipype*, a Python-based neuro-imaging data processing tool [Gorgolewski et al. 2011]. Nipype could, with the event-enriched data, create an individualized GLM model and enable an thorough analysis of the fMRI data.

5 CONCLUSION

In this paper, we have discussed why simultaneous fMRI and eye tracking can be insightful, but also challenging. Once we can conjointly analyze both data streams, we may generate new hypotheses for program comprehension, test existing hypotheses, and eventually create a theory of program comprehension. To properly support all these goals, we see the need for tool support. This includes a way of annotating eye-tracking data with events, which should be customizable for each experiment and stimuli and dealing with the haemodynamic delay of fMRI data. Such tool support would facilitate a more fine-grained analysis of brain activation and increase the value of fMRI studies on program comprehension.

In the long term, multi-modal experiments like simultaneous fMRI and eye tracking may help to considerably move our understanding of program comprehension forward. While the future work is a long road ahead, it offers a revolutionizing perspective and, in our mind, is worth to be pursued.

ACKNOWLEDGMENTS

We thank all participants of the fMRI study. Furthermore, we thank Andreas Fügner, Anke Michalsky, and Jörg Stadler for their technical support during pilot studies and fMRI data acquisition.

Siegmund's and Brechmann's work is supported by DFG grants SI 2045/2-1 and BR 2267/7-1. Siegmund's work is further funded by the Bavarian State Ministry of Education, Science and the Arts in the framework of the Centre Digitisation.Bavaria (ZD.B). Parnin's work is supported by the National Science Foundation under grant number 1755762. Apel's work has been supported by the German Research Foundation (AP 206/6).

REFERENCES

Titus Barik, Justin Smith, Kevin Lubick, Elisabeth Holmes, Jing Feng, Emerson Murphy-Hill, and Chris Parnin. 2017. Do Developers Read Compiler Error Messages?. In *Proc. Int'l Conf. Software Engineering (ICSE)*. IEEE, 575–585.

Jackson Beatty and Brennis Lucero-Wagoner. 2000. The Pupillary System, Handbook of Psychophysiology, Cacioppo, Tassinary & Berntson. (2000).

Roman Bednarik and Markku Tukiainen. 2006. An Eye-Tracking Methodology for Characterizing Program Comprehension Processes. In *Proc. Symposium on Eye Tracking Research & Applications (ETRA)*. ACM, 125–132.

Mahnaz Behroozi, Alison Lui, Ian Moore, Denae Ford, and Chris Parnin. 2018. Dazed: Measuring the Cognitive Load of Solving Technical Interview Problems at the Whiteboard. In *Proc. Int'l Conf. Software Engineering (ICSE)*. IEEE, 4 pages.

Wolfram Boucsein. 2012. *Electrodermal Activity*. Springer Science & Business Media.

Ruven Brooks. 1983. Towards a Theory of the Comprehension of Computer Programs. *Int'l Journal of Man-Machine Studies* 18, 6 (1983), 543–554.

Teresa Busjahn, Roman Bednarik, Andrew Begel, Martha Crosby, James H. Paterson, Carsten Schulte, Bonita Sharif, and Sascha Tamm. 2015. Eye Movements in Code Reading: Relaxing the Linear Order. In *Proc. Int'l Conf. Program Comprehension (ICPC)*. IEEE, 255–265.

B. Chance, Z. Zhuang, C. UnAh, C. Alter, and Lipton L. 1993. Cognition-Activated Low-Frequency Modulation of Light Absorption in Human Brain. *Proc. Nat'l Academy Sciences of the United States of America (PNAS)* 90, 8 (1993), 3770–3774.

Igor Crk, Timothy Kluthe, and Andreas Stefik. 2015. Understanding Programming Expertise: An Empirical Study of Phasic Brain Wave Changes. *ACM Trans. Comput.-Hum. Interact.* 23, 1 (2015), 2:1–2:29.

João Duraes, Henrique Madeira, J Castelhano, C Duarte, and M Castelo Branco. 2016. WAP: Understanding the Brain at Software Debugging. In *Proc. Int'l Symposium Software Reliability Engineering (ISSRE)*. IEEE, 87–92.

Sarah Fakhoury, Yuzhan Ma, Venera Arnaoudova, and Olusola Adesope. 2018. The Effect of Poor Source Code Lexicon and Readability on Developers' Cognitive Load. In *Proc. Int'l Conf. Program Comprehension (ICPC)*. IEEE, 11 pages.

Benjamin Floyd, Tyler Santander, and Westley Weimer. 2017. Decoding the Representation of Code in the Brain: An fMRI Study of Code Review and Expertise. In *Proc. Int'l Conf. Software Engineering (ICSE)*. IEEE, 175–186.

Denae Ford, Titus Barik, and Chris Parnin. 2015. Studying Sustained Attention and Cognitive States with Eye Tracking in Remote Technical Interviews. *Eye Movements in Programming: Models to Data* (2015), 5 pages.

Thomas Fritz, Andrew Begel, Sebastian C. Müller, Serap Yigit-Elliott, and Manuela Züger. 2014. Using Psycho-Physiological Measures to Assess Task Difficulty in Software Development. In *Proc. Int'l Conf. Software Engineering (ICSE)*. ACM, 402–413.

Michael S. Gazzaniga, Richard B. Ivry, and George R. Mangun. 2013. *Cognitive Neuroscience: The Biology of the Mind*. Norton & Company.

Krzysztof Gorgolewski, Christopher D. Burns, Cindee Madison, Dav Clark, Yaroslav O. Halchenko, Michael L. Waskom, and Satrajit S. Ghosh. 2011. Nipype: A Flexible, Lightweight and Extensible Neuroimaging Data Processing Framework in Python. *Frontiers in Neuroinformatics* 5 (2011), 13 pages.

Frances K. Graham. 1992. Attention: The Heartbeat, the Blink, and the Brain. *Attention and Information Processing in Infants and Adults: Perspectives from Human and Animal Research* 8 (1992), 3–29.

Kenneth Holmqvist, Marcus Nyström, Richard Andersson, Richard Dewhurst, Halszka Jarodzka, and Joost Van de Weijer. 2011. *Eye Tracking: A Comprehensive Guide to Methods and Measures*. OUP Oxford.

Thomas D. LaToza, Gina Venolia, and Robert DeLine. 2006. Maintaining Mental Models: A Study of Developer Work Habits. In *Proc. Int'l Conf. Software Engineering (ICSE)*. ACM, 492–501.

Seolhwa Lee, Danial Hooshyar, Hyesung Ji, Kichun Nam, and Heuiseok Lim. 2017. Mining Biometric Data to Predict Programmer Expertise and Task Difficulty. *Cluster Computing* (2017), 1–11.

Sascha Meudt, Lutz Bigalke, and Friedhelm Schwenker. 2012. Atlas - Annotation Tool Using Partially Supervised Learning and Multi-View Co-Learning in Human-Computer-Interaction Scenarios. In *Int'l Conf. Information Science, Signal Processing and their Applications (ISSPA)*. IEEE, 1309–1312.

Takao Nakagawa, Yasutaka Kamei, Hidetake Uwano, Akito Monden, Kenichi Matsumoto, and Daniel M. German. 2014. Quantifying Programmers' Mental Workload During Program Comprehension Based on Cerebral Blood Flow Measurement: A Controlled Experiment. In *Proc. Int'l Conf. Software Engineering (ICSE)*. ACM, 448–451.

Keith Nolan, Aidan Mooney, and Susan Bergin. 2015. Examining the Role of Cognitive Load When Learning to Program Program. January (2015), 2–4.

Norman Peitek, Janet Siegmund, and André Brechmann. 2017. Enhancing fMRI Studies of Program Comprehension with Eye-Tracking. In *Proc. Int'l Workshop on Eye Movements in Programming*. Freie Universität Berlin, 22–23.

Bonita Sharif and Johnathon Maletic. 2010. An Eye Tracking Study on camelCase and under_score Identifier Styles. In *Proc. Int'l Conf. Program Comprehension (ICPC)*. IEEE, 196–205.

Janet Siegmund. 2016. Program Comprehension: Past, Present, and Future. In *Int'l Conf. Software Analysis, Evolution, and Reengineering (SANER)*. IEEE, 13–20.

Janet Siegmund, André Brechmann, Sven Apel, Christian Kästner, Jörg Liebig, Thomas Leich, and Gunter Saake. 2012. Toward Measuring Program Comprehension with Functional Magnetic Resonance Imaging. In *Proc. Int'l Symposium Foundations of Software Engineering–New Ideas Track (FSE-NIER)*. ACM, 24:1–24:4.

Janet Siegmund, Christian Kästner, Sven Apel, Chris Parnin, Anja Bethmann, Thomas Leich, Gunter Saake, and André Brechmann. 2014. Understanding Understanding Source Code with Functional Magnetic Resonance Imaging. In *Proc. Int'l Conf. Software Engineering (ICSE)*. ACM, 378–389.

Janet Siegmund, Norman Peitek, Chris Parnin, Sven Apel, Johannes Hofmeister, Christian Kästner, Andrew Begel, Anja Bethmann, and André Brechmann. 2017. Measuring Neural Efficiency of Program Comprehension. In *Proc. of the 2017 11th Joint Meeting on Foundations of Software Engineering (ESEC/FSE)*. ACM, 140–150.

Thomas Standish. 1984. An Essay on Software Reuse. *IEEE Trans. Softw. Eng.* SE–10, 5 (1984), 494–497.

Rebecca Tiarks. 2011. What Programmers Really Do: An Observational Study. *Softwaretechnik-Trends* 31, 2 (2011), 36–37.

Adrian Voßkühler, Volkhard Nordmeier, Lars Kuchinke, and Arthur M. Jacobs. 2008. OGAMA (Open Gaze and Mouse Analyzer): Open-Source Software Designed to Analyze Eye and Mouse Movements in Slideshow Study Designs. *Behavior Research Methods* 40, 4 (2008), 1150–1162.

Shehnaaz Yusuf, Huzefa Kagdi, and Jonathan I. Maletic. 2007. Assessing the Comprehension of UML Class Diagrams via Eye Tracking. In *Proc. Int'l Conf. Program Comprehension (ICPC)*. IEEE, 113–122.