UNIVERSITY OF SASKATCHEWAN

Exploring the Influence of Method Dependencies on Code Understanding

You are invited to participate in a research study entitled: **Exploring the Influence of Method Dependencies on Code Understanding**. This survey requires 10-15 minutes to complete.

Student researcher: Md Mustakim Billah, Graduate Student, Department of Computer Science, University of Saskatchewan, Canada. Email: mustakim.billah@usask.ca

Supervisor: Dr. Banani Roy, Assistant Professor, Email: banani.roy@usask.ca, Phone: +13068505630; Department of Computer Science, University of Saskatchewan, Canada.

Funding: The project is funded by the faculty scholarship/stipend from the Department of Computer Science, University of Saskatchewan, Canada.

Purpose and objective of the study: Code comment generation involves automatically creating descriptive comments for code segments, aiding in source code understanding. These comments serve as invaluable documentation, enhancing code readability, facilitating collaboration, and accelerating software development by providing insights into code functionality and design decisions.

In practical software development, source code comprises various methods, often interdependent on each other. In this study, we will try to explore the effects of these method dependencies on code understanding. We also introduce a novel technique for efficiently generating method comments by leveraging the context provided by dependent methods. This approach harnesses the power of large language models to enhance the clarity and comprehensibility of method-level comment generation.

Importance of taking feedback: We seek input from software developers and researchers to validate the study findings and address any potential weaknesses identified through their feedback. Their insights are crucial for refining and strengthening the study's conclusions.

Question Types: Participants will be mostly asked close-ended multiple-choice questions with one open-ended question at the end to take their feedback about the survey.

Potential benefits: Participants will not receive any personal benefits, but they may gain academic insights into the process of designing and participating in research surveys. However, the survey findings may positively impact the software research community in developing better-automated techniques for Code Understanding.

Confidentiality: It's important to note that all data and codes used in this study are strictly confidential and should not be shared with any third party. Additionally, Your participation and data will be anonymous. The collected data will be disseminated in research publications and academic theses in the future. The data will be reported anonymously in an aggregated or summarized form. This survey is hosted by Survey Monkey. The data will be stored in facilities hosted in Canada. Please see the following for more information on the Survey Monkey Privacy Policy.

Risks: There are no known or anticipated risks.

UNIVERSITY OF SASKATCHEWAN

Exploring the Influence of Method Dependencies on Code Understanding

Right to withdraw: Participation in this survey is voluntary. Although participation and the data will be anonymous, you can decide not to participate at any time by closing your browser or choosing not to answer any questions you do not feel comfortable with. However, once the survey is submitted, it cannot be removed.

Storage of data: Data will be stored in table format considering all answers as columns and all responses as rows in the OneDrive account of the principal investigator (Dr. Banani Roy). Professor Banani Roy will have access to the data and can share it with prospective students when required. The data will be stored for five years post-publication and could be used in other research in the future without destroying it if needed. Data will be destroyed using a method that ensures it cannot be recovered five years after publication, provided no further research is being conducted on this topic.

Follow up: To obtain results from the study, participants can email student researcher Md Mustakim Billah or the USASK Interactive Software Engineering and Analytics lab's official website. The aggregated result will be available in a summarized version once it is accepted by one peer-reviewed journal or conference (estimated date: December 31, 2024).

Questions or concerns: Contact the researcher(s) using the information at the top. This research project has been approved on ethical grounds by the University of Saskatchewan's behavioral research ethics board. The participants may print the screen if they wish a copy of the consent form for their records. Any questions regarding your rights as a participant may be addressed to that committee through the research ethics office: ethics.office@usask.ca; 306-966-2975; out-of-town participants may call toll-free 1-888-966-2975.

* **1. Please indicate whether or not you consent to participate.**

- ◯ I consent to participate in the study
- ◯ I do not consent, and hence prefer not to participate

* **2. Do you consent to the potential use of your responses in future research?**

- ◯ Yes
- ◯ No

University of Saskatchewan

Exploring the Influence of Method Dependencies on Code Understanding

Demographic Information

We would appreciate your assistance in providing some basic demographic details. This information will help us understand participants' diverse perspectives and ensure that our analysis considers various demographic factors.

* **3. Which role below describes your current designation best?**

- ◯ Software Developer
- ◯ Researcher
- ◯ Student
- ◯ Other

* **4. How many years of experience do you have in programming?**

- ◯ 0-2
- ◯ 3-5
- ◯ 6-8
- ◯ More than 8

5. In which country do you currently reside?

UNIVERSITY OF SASKATCHEWAN

Exploring the Influence of Method Dependencies on Code Understanding

Task 1

Subtask 1: Understanding the code without any information
Subtask 2: Understanding the code with the help of the dependent methods
Subtask 3: Selecting the comment that best describes the code.

*** 6. Consider the JAVA method given below, and try to understand the logic.**

```
private String consumeCommentTokens(String line) {
        if (line.indexOf(START_COMMENT) == -1 && line.indexOf(END_COMMENT) == -1)
{           return line;
        }
        while ((line = consume(line)) != null) {
            if (!this.inComment && !line.trim().startsWith(START_COMMENT)) {
                return line;
            }
        }
        return line;
    }
```

On a scale of 1 to 10, please indicate how difficult it was for you to understand the code, where a higher rating indicates higher difficulty.

Low Medium High

*** 7. Now, for the same code given above, we give you the dependent methods' implementations.**

```java
//main method
private String consumeCommentTokens(String line) {
        if (line.indexOf(START_COMMENT) == -1 && line.indexOf(END_COMMENT) == -1) {
            return line;
        }
        while ((line = consume(line)) != null) {
            if (!this.inComment && !line.trim().startsWith(START_COMMENT)) {
                return line;
            }
        }
        return line;
    }
//dependent method 1
@Override
    public int indexOf(Object o) {
        return this.backingList.indexOf(o);
    }
// dependent method 2
private String consume(String line) {
        int index = (this.inComment ? endComment(line) : startComment(line));
        return (index == -1 ? null : line.substring(index));
    }
```

On a scale of 1 to 10, please indicate how difficult it was for you to understand the code this time, where a higher rating indicates higher difficulty.

Low Medium High

*** 8. Now for the same JAVA method again, select the comment that best describes the method according to you.**

```java
private String consumeCommentTokens(String line) {
        if (line.indexOf(START_COMMENT) == -1 && line.indexOf(END_COMMENT) == -1)
{           return line;
        }
        while ((line = consume(line)) != null) {
            if (!this.inComment && !line.trim().startsWith(START_COMMENT)) {
                return line;
            }
        }
        return line;
    }
```

- ◯ Consumes all comment tokens, returning the first non-comment token.
- ◯ Consumes comment tokens within a given input line, utilizing helper methods like indexOf and consume to handle comment detection efficiently.
- ◯ Consume comment tokens .

*** 9. If the comment you selected on the previous question, were given to you at first place, would it have been easier for you to understand the code with less effort?**

- ◯ Yes
- ◯ No
- ◯ Neutral

UNIVERSITY OF SASKATCHEWAN

Exploring the Influence of Method Dependencies on Code Understanding

Task 2

Subtask 1: Understanding the code without any information
Subtask 2: Understanding the code with the help of the dependent methods
Subtask 3: Selecting the comment that best describes the code.

*** 10. Consider the JAVA method given below, and try to understand the logic.**

```
private void readParameters(final OContextConfiguration iServerConfig, final
OServerParameterConfiguration[] iParameters) {
    configuration = new OContextConfiguration(iServerConfig);

    if (iParameters != null && iParameters.length > 0) {
      for (OServerParameterConfiguration param : iParameters)
        configuration.setValue(param.name, param.value);
    }

    socketBufferSize = configuration.getValueAsInteger(
      OGlobalConfiguration.NETWORK_SOCKET_BUFFER_SIZE);
  }
```

On a scale of 1 to 10, please indicate how difficult it was for you to understand the code, where a higher rating indicates higher difficulty.

Low Medium High

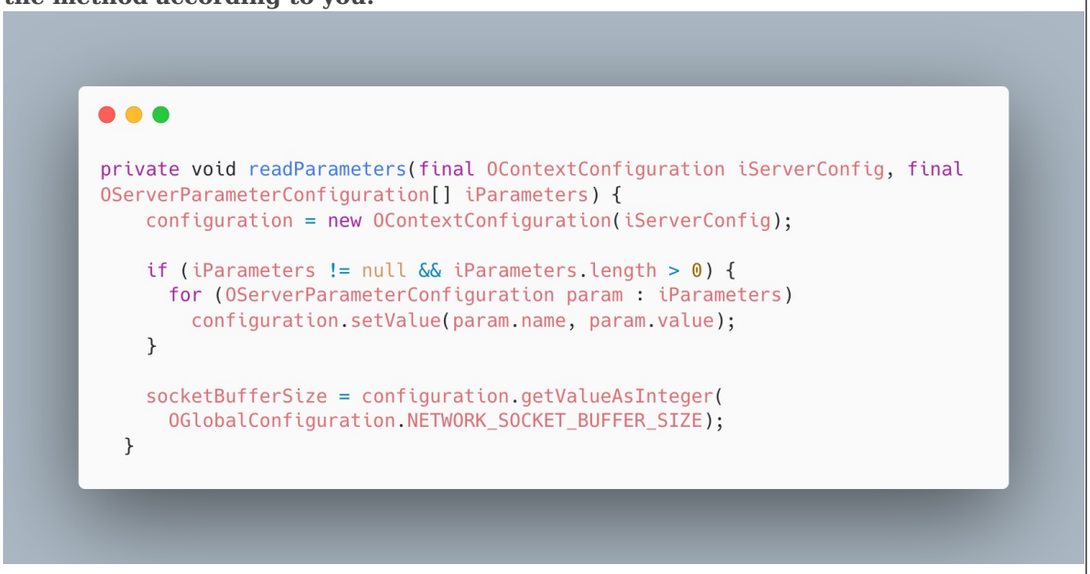
*** 11. Now, for the same code given above, we give you the dependent methods' implementations.**

```
private void readParameters(final OContextConfiguration iServerConfig, final
OServerParameterConfiguration[] iParameters) {
    configuration = new OContextConfiguration(iServerConfig);

    if (iParameters != null && iParameters.length > 0) {
      for (OServerParameterConfiguration param : iParameters)
        configuration.setValue(param.name, param.value);
    }

    socketBufferSize = configuration.getValueAsInteger(
OGlobalConfiguration.NETWORK_SOCKET_BUFFER_SIZE);
  }
//dependent method 1
public Object setValue(final OGlobalConfiguration iConfig, final Object iValue) {
    if (iValue == null) return config.remove(iConfig.getKey());

    return config.put(iConfig.getKey(), iValue);
  }
// dependent method 2
public int getValueAsInteger(final OGlobalConfiguration iConfig) {
    final Object v = getValue(iConfig);
    if (v == null) return 0;
    return v instanceof Integer ? ((Integer) v).intValue() : Integer.parseInt(v.toString());
  }
```

On a scale of 1 to 10, please indicate how difficult it was for you to understand the code this time, where a higher rating indicates higher difficulty.

Low Medium High

*** 12. Now for the same JAVA method again, select the comment that best describes the method according to you.**



```
private void readParameters(final OContextConfiguration iServerConfig, final
OServerParameterConfiguration[] iParameters) {
    configuration = new OContextConfiguration(iServerConfig);

    if (iParameters != null && iParameters.length > 0) {
      for (OServerParameterConfiguration param : iParameters)
        configuration.setValue(param.name, param.value);
    }

    socketBufferSize = configuration.getValueAsInteger(
      OGlobalConfiguration.NETWORK_SOCKET_BUFFER_SIZE);
  }
```

- ◯ Reads parameters and initializes the context configuration using the provided server configuration and parameters, setting the socket buffer size based on the integer value obtained from the configuration.
- ◯ Reads the parameters from the configuration file .
- ◯ Read parameters from server configuration .

*** 13. If the comment you selected on the previous question, were given to you at first place, would it have been easier for you to understand the code with less effort?**

- ◯ Yes
- ◯ No
- ◯ Neutral

UNIVERSITY OF SASKATCHEWAN

Exploring the Influence of Method Dependencies on Code Understanding

Task 3

Subtask 1: Understanding the code without any information
Subtask 2: Understanding the code with the help of the dependent methods
Subtask 3: Selecting the comment that best describes the code.

*** 14. Consider the JAVA method given below, and try to understand the logic.**

```java
public static int getCharacterCountDecimal(long integerValue, int scale)
    {
        boolean isNeg = integerValue < 0;

        int totalDigits = BitHackUtils.getCharacterCountInt64(integerValue);
        int totalLength = totalDigits;

        if (isNeg)
        {
            totalDigits--;
        }

        if (scale > 0)
        {
            totalLength++;

            if (scale >= totalDigits)
            {
                totalLength += (scale - totalDigits) + 1;
            }
        }
        else
        {
            totalLength -= scale;
        }

        return totalLength;
    }
```

On a scale of 1 to 10, please indicate how difficult it was for you to understand the code, where a higher rating indicates higher difficulty.

Low Medium High

*** 15. Now, for the same code given above, we give you the dependent methods' implementations.**

```
public static int getCharacterCountDecimal(long integerValue, int scale)
    {
        boolean isNeg = integerValue < 0;

        int totalDigits = BitHackUtils.getCharacterCountInt64(integerValue);
        int totalLength = totalDigits;

        if (isNeg)
        {
            totalDigits--;
        }

        if (scale > 0)
        {
            totalLength++;

            if (scale >= totalDigits)
            {
                totalLength += (scale - totalDigits) + 1;
            }
        }
        else
        {
            totalLength -= scale;
        }

        return totalLength;
    }
//Dependent method 1
public static int getCharacterCountInt64(long value)
    {
        if (value >= 0)
        {
            return getCharacterCountUInt64(value);
        }
        else if (value == Long.MIN_VALUE)
        {
            return getCharacterCountUInt64(Long.MAX_VALUE) + 1;
        }
        else
        {
            return getCharacterCountUInt64(-value) + 1;
        }
    }
```

On a scale of 1 to 10, please indicate how difficult it was for you to understand the code this time, where a higher rating indicates higher difficulty.

Low Medium High

*** 16. Now for the same JAVA method again, select the comment that best describes the method according to you.**

```java
public static int getCharacterCountDecimal(long integerValue, int scale)
    {
        boolean isNeg = integerValue < 0;

        int totalDigits = BitHackUtils.getCharacterCountInt64(integerValue);
        int totalLength = totalDigits;

        if (isNeg)
        {
            totalDigits--;
        }

        if (scale > 0)
        {
            totalLength++;

            if (scale >= totalDigits)
            {
                totalLength += (scale - totalDigits) + 1;
            }
        }
        else
        {
            totalLength -= scale;
        }

        return totalLength;
    }
```

- ◯ Returns the number of characters needed to represent a decimal number in string form.
- ◯ Returns the number of characters in a decimal value .
- ◯ Calculates the total number of characters needed to represent a decimal value as a string, taking into account the integer value, scale, negative values, decimal point, and leading zeros.

*** 17. If the comment you selected on the previous question, were given to you at first place, would it have been easier for you to understand the code with less effort?**

- ◯ Yes
- ◯ No
- ◯ Neutral

UNIVERSITY OF SASKATCHEWAN

Exploring the Influence of Method Dependencies on Code Understanding

Task Load Index

In this section, you will evaluate the cognitive load associated with this survey.

We gave you three subtasks in each task.

Subtask 1: Understanding the code without any information
Subtask 2: Understanding the code with the help of the dependent methods
Subtask 3: Selecting the comment that best describes the code.

*** 18. How mentally demanding were the subtasks?**

	Low				Medium					High
Subtask 1	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯
Subtask 2	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯
Subtask 3	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯

*** 19. How hard did you have to work for the subtasks to accomplish your level of performance?**

	Low				Medium					High
Subtask 1	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯
Subtask 2	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯
Subtask 3	◯	◯	◯	◯	◯	◯	◯	◯	◯	◯

University of Saskatchewan

Exploring the Influence of Method Dependencies on Code Understanding

Feedback

20. Do you have any concerns or suggestions that might improve the work in the future?

University of Saskatchewan

Exploring the Influence of Method Dependencies on Code Understanding

Really appreciate your effort and time.