# Implementation of RETUSS to Ensure Traceability between Class Diagram in UML and Java Source Code in Real Time

**Keisuke Mori*, Tetsuro Katayama*, Yoshihiro Kita†,**
**Hisaaki Yamaba*, Kentaro Aburada*, and Naonobu Okazaki***
**University of Miyazaki, 1-1 Gakuen-kibanadai nishi, Miyazaki, 889-2192, Japan*
*†Tokyo University of Technology, 1404-1 Katakura, Hachioji, 192-0914, Japan*
*E-mail: mori@earth.cs.miyazaki-u.ac.jp, kat@cs.miyazaki-u.ac.jp, kitayshr@stf.teu.ac.jp, yamaba@cs.miyazaki-u.ac.jp, aburada@cs.miyazaki-u.ac.jp, oka@cs.miyazaki-u.ac.jp*

**Abstract**

Ensuring of the traceability of deliverables is one of effective methods to secure the quality of software. It can verify that the requirements are reflected in the programs. But it has two problems: taking much labor and time, and causing mistakes by human handling. This paper has implemented RETUSS (Real-time Ensure Traceability between UML and Source-code System) in order to solve the above two problems. RETUSS can ensure traceability between Class diagram in UML and Java source code in real time.

*Keywords*: Software Quality, Traceability, UML, Java

## 1. Introduction

It's increasing the importance of software in society, and it's becoming more important to secure the quality of software. Ensuring of the traceability of deliverables is one of effective methods to secure the quality of software[1]. It can verify that the requirements are reflected in the programs, specify the scope of the impact due to the modification in the requirements, and remove the gap between the documents and the source code. But it has the following two problems:

- Taking much labor and time to modify similarly other related deliverables by modifying some in a deliverable
- Having a risk that you cannot ensure traceability because of causing mistakes to ensure traceability by human handling

This paper has implemented RETUSS (Real-time Ensure Traceability between UML and Source-code System) in order to solve the above two problems. RETUSS can ensure traceability between Class diagram in UML and Java source code in real time.

UML[2] (Unified Modeling Language) is a visual language for expressing the design and pattern of software. It is used for requirement specification and system design documents[3]. RETUSS targets Class diagram, which is an important diagram expressing the static structure of the system in UML. It also targets source codes in Java language.

## 2. RETUSS

Figure 1 shows an overview of RETUSS implemented in this paper. RETUSS has the following five areas:

- Menu Bar
- File List Area
- Drawing Item Selection Area
- UML Description Area
- Source Code Description Area

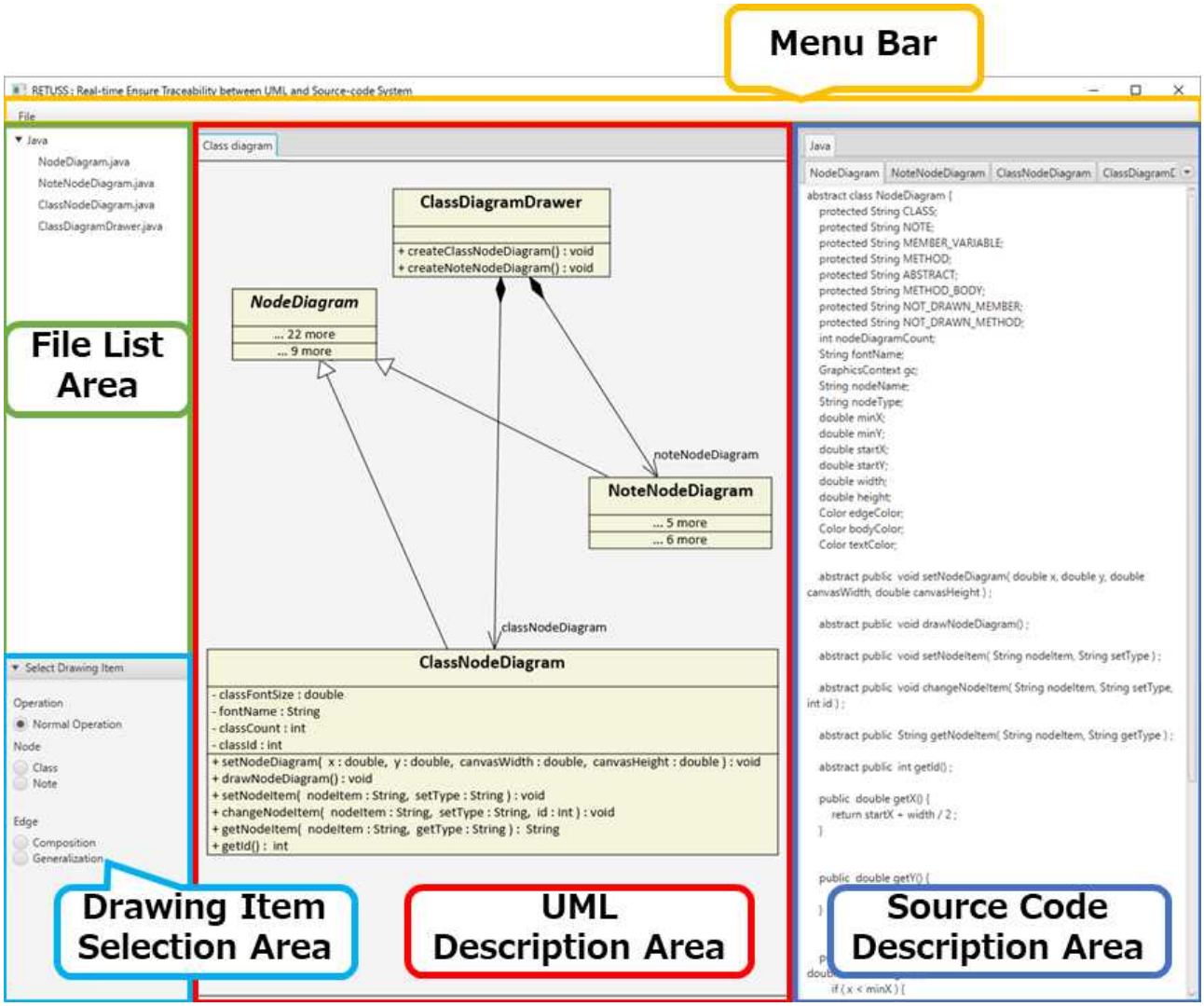


Fig. 1. Overview of RETUSS

In addition, RETUSS has the following three functions:

- Open java a file
- Description of Class diagram
- Description of Java source code

A user can describe Class diagrams in the UML Description Area in the "Description of Class diagram" function. Items and functions that can be described are as follows:

- Description of Classes and Notes
- Moving Classes and Notes
- Modifying Class name and Note content
- Deleting Classes and Notes
- Adding Attributes and Operations in Class
- Modifying the Attributes and Operations in Class
- Deleting Attributes and Operations in Class
- Hiding Attributes and Operations in Class
- Description of Composition and Generalization between Classes

A user can describe Java source code in the Source Code Description Area in the "Description of Java source code" function. When the user describes Java source code, RETUSS draws Class diagram corresponding to source code's contents in UML Description Area. RETUSS

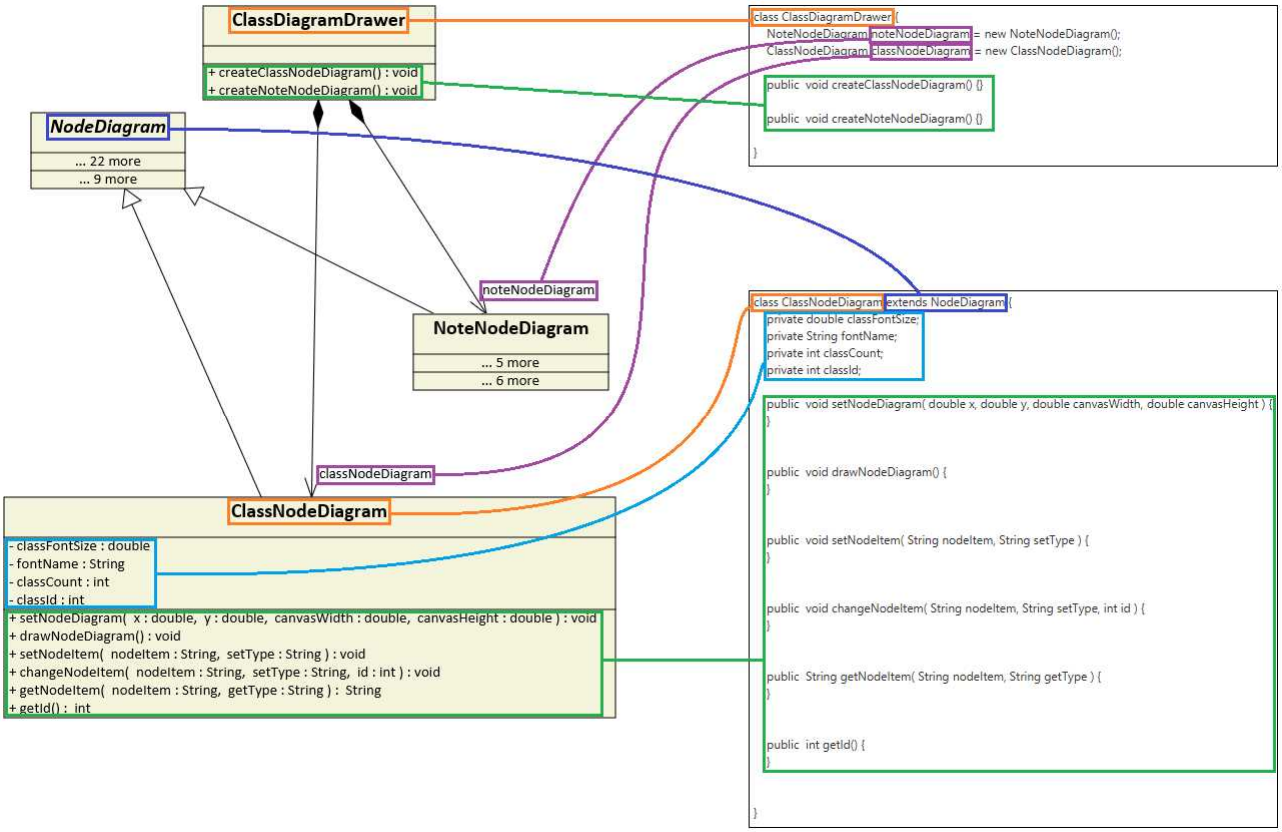


Fig. 2. The correspondence between the Class diagram and Java source code in executing the application example

draws multiple Classes corresponding to Java source code in the Source Code Description Area. Because one tab corresponds to one Class, a user draws Class of the corresponding the Java source code in the text area of each tab.

## 3. Application Example

As an application example, a part of Class diagram showing the structure of RETUSS itself is adopted to verify that the functions of it are operated correctly. Figure 1 shows a screenshot of RETUSS in executing the application example, and Figure 2 shows the correspondence between the Class diagram and Java source code at this time.

By looking at the ClassNodeDiagram class of the Class diagram and the Java source code in Fig. 2, the Class name, the Class name of the Generalization destination, the Attributes and Operations in the Class diagram correspond to the class name, the class name of the inherited destination, the field and the method in the Java source code, respectively. Similarly, by looking at the ClassDiagramDrawer class, the Class name, the Related end name of the Composition destination, Operation in the Class diagram correspond to the class name, field, and method in the Java source code, respectively.

Hence, it is clear that RETUSS can ensure traceability between Class diagram in UML and Java source code.

## 4. Evaluation of Usefulness

To consider RETUSS usefulness, we experiment with experimental participants. As an experimental method, the experimental participants modify Class diagrams and Java source code the ensured traceability. Specifically, the experimental participants modify Class diagram and the Java source code to ensure traceability for requirements change when the Class diagram and Java source code already exist. As a modification method, two cases are prepared: case A using RETUSS, and case B

Table 1. The time required by three experimental participants to ensure traceability in each case (seconds).

| Trials count | Case A | Case B |
|---|---|---|
| One | 113 | 292 |
| Two | 76 | 148 |
| Three | 129 | 229 |
| Average | 106 | 223 |

without using. For each case, we measure and compare the time required for the experimental participants to ensure traceability between the modified Class diagram and Java source code.

Table 1 shows the time required by three experimental participants to ensure traceability in each case.

The average time required for modifying in case A using RETUSS was 106 seconds. On the other hand, the average time required for modifying in case B without using RETUSS was 223 seconds. Case A could be reduced by 52.7% compared to case B.

Further, in case B without using RETUSS, some mistakes were caused because of ensuring traceability by human handling. During the modification, for example, mismatch between Visibility of Class and access modifiers of Java source code, and deletion of curly braces of a method body of Java source code. On the other hand, the above mentioned errors did not occur in case A using RETUSS, because it can ensure traceability automatically.

Hence, it's possible to reduce labor and time and eliminate causing for human handling to ensure traceability by using RETUSS.

## 5. Related Research

EA[4] (Enterprise Architect) and Astah[5] are modeling tools that describe UML. Both EA and Astah can generate source code from UML and UML from source code automatically. EA and Astah can ensure traceability of source code are language such as Java, C# and C++.

EA, Astah and RETUSS support the design phase and maintenance phase of software development in using UML. EA and Astah can generate skeleton Java source code from Class diagram and Class diagrams from implemented Java source code in methods automatically. But EA and Astah cannot ensure traceability between Class diagrams and Java source code in real time. On the other hand, RETUSS can ensure traceability between Class diagrams and Java source code in real time.

Therefore, RETUSS takes less time to modify similarly other related deliverables by modifying some in a deliverable, compared to the other two tools.

## 6. Conclusion

In this paper, we have implemented RETUSS, that ensures traceability between UML and source code in real time with the aim of solving two problems in ensuring traceability.

We applied RETUSS to a part of Class diagram of the structure of itself, and our tool ensured traceability between Class diagram in UML and Java source code. Finally, as a result of experiments with the experimental participants, the time could be reduced by 52.47% to ensure traceability between the Class diagram and the Java source code, and elimination causing for human handling by using RETUSS. Therefore, RETUSS can solve two problems in ensuring traceability.

Consequently, RETUSS is expected to support to secure the quality of software.

Future works are as follows:

- Correspond to the same Attribute and same Operation
- Correspond to incompatible syntax in Java source code
- Extension of Class diagram description function

## References

1. SQuBOK Sakutei Bukai, *Guide to the Software Quality Body of Knowledge*, 2nd edn. Ohmsha, 2014 (in Japanese).
2. *Welcome To UML Web Site!*, http://www.uml.org/, (accessed December 7).
3. Dan Pilone, Neil Pitman, *UML 2.0 in a Nutshell.* O'Reilly Media, 2009.
4. *UML tools for software development and modelling - Enterprise Architect UML modeling tool*, https://www.sparxsystems.com/, (accessed December 7).
5. *Astah - Software Design Tools for Agile teams with UML, ER Diagram, Flowchart, Mindmap and More | Astah.net*, http://astah.net/, (accessed December 7).