

A Study on Conversion Between UML and Source Code Based on RTT(Round-Trip Translator)

Kim Ji Yong[†] · Cho Han Joo^{**} · Kim Young Jong^{***}

ABSTRACT

As programming education becomes more important in recent years, it is necessary to learn how the source code written by students reflects Object-Oriented(OO) concepts. We present a tool called the Round-Trip Translator(RTT) that transforms the Unified Modeling Language(UML) class diagram and Java source code to provide a web-based environment that provides real-time synchronization of UML and source code. RTT was created by improving existing RTE and is a tool for students who are learning OO concepts to understand how their UML or source code reflects the concepts that user intended. This study compares the efficiency and user-friendliness of RTT with the existing Round-Trip Engineering-based tools. The results show that students have improved understanding of OO concepts through UML and source code translation by using the RTT. We also found out that students were satisfied with the use of the RTT, which provides more efficient and convenient user interface than the existing tools.

Keywords : Round-Trip Engineering, UML, Object-Oriented Programming, Education, Software Tools

RTT(Round-Trip Translator) 기반의 UML과 소스코드 변환에 대한 연구

김지용[†] · 조한주^{**} · 김영종^{***}

요약

최근 프로그래밍 교육이 중요해짐에 따라, 객체지향 프로그래밍에 있어 작성한 소스코드가 어떻게 객체지향 개념을 반영하고 있는지를 분석하는 것은 중요하다. 본 논문에서는 UML과 소스코드의 실시간 동기화를 제공하는 웹 기반 환경에서 객체지향 개념을 분석 및 학습하기 위해 Unified Modeling Language(UML) 클래스 다이어그램과 자바 소스코드를 상호 변환하는 Round-Trip Translator(RTT)를 제안한다. RTT는 기존 Round-Trip Engineering(RTE)기반 도구를 개선시켜 구현 하였으며 작성된 UML 혹은 소스코드가 어떻게 객체지향 개념을 반영하는 지에 대한 이해도를 높이기 위한 도구이다. 본 논문은 기존 RTE 기반 도구와 RTT의 효율성과 사용자의 편의성을 비교한 결과 UML과 소스코드 상호 변환을 통해 객체지향에 대한 이해도가 향상되었음을 보여준다. 우리는 또한 학생들이 기존 도구보다 더 효율적이고 편리한 사용자 인터페이스를 제공하는 RTT 사용에 만족한다는 사실을 알아냈다.

키워드 : 라운드트립 엔지니어링, UML, 객체지향 프로그래밍, 교육, 프로그램 도구

1. 서론

Round-Trip Engineering(RTE)은 소스코드, 모델, 구성파일 및 문서와 같은 두 가지 이상의 관련 소프트웨어 산출물을 동기화하는 소프트웨어 개발 도구의 기능이다[1]. 또한 순공학, 역공학 등 전통적인 소프트웨어 엔지니어링 분야와 밀

접한 관련이 있다. 이미 오랜 기간 통합 개발 환경에서 사용되어 왔으며 대표적인 예로는 Visual Paradigm, StarUML, Papyrus 등이 있다.

자바 언어는 객체지향 개념과 원칙 때문에 학습하기에 복잡하고 어렵다[2]. 객체지향 개념을 공부하거나 객체지향 언어를 사용하여 개발하는 경우에 소스코드가 어떤 객체지향적인 개념을 반영하는지 알 수 있어야 한다.

객체지향에 대한 이해도를 향상시키기 위해 Visual Paradigm, StarUML, Papyrus 등의 제품들이 출시되어 있다. 그러나 이러한 제품들을 사용하려면 많은 설정들을 해야 하기 때문에 효율성이 떨어지고 사용자에게 편의성을 제공해주지 못한다. 예를 들어, StarUML 도구는 사용자 인터페이스에

[†] 정회원 : 숭실대학교 융합소프트웨어학과 석사과정

^{**} 비회원 : 숭실대학교 소프트웨어학부 학사과정

^{***} 종신회원 : 숭실대학교 소프트웨어학부 부교수

Manuscript Received : January 14, 2019

First Revision : April 23, 2019

Second Revision : June 18, 2019

Accepted : July 13, 2019

* Corresponding Author : Kim Young Jong(youngjong@ssu.ac.kr)

서 UML을 그리는 기능은 제공하지만 소스코드를 수정할 수 있는 에디터가 내장되어 있지 않아 학습자는 번거로운 과정을 거쳐 산출물을 확인해야 한다[3].

본 논문에서는 사용자 인터페이스에서의 실시간 동기화가 결여된 기존의 RTE 기반 도구보다 효율적이고 편리한 도구를 제안한다. 기존 도구들은 에디터가 내장되어 있지 않아 소스코드를 사용자 인터페이스에서 바로 수정할 수 없다. 그러나 우리가 제안한 도구는 웹 기반의 환경을 제공함으로써 사용자가 실시간으로 UML과 소스코드가 동기화된 산출물을 바로 볼 수 있다. 따라서 사용자는 객체지향 개념을 쉽고 편하게 이해할 수 있다.

2. UML을 활용한 객체지향 학습 도구 RTT

기존 RTE 도구를 개선하여 만든 Unified Modeling Language(UML) 클래스 다이어그램과 자바 소스코드를 상호 변환하는 본 논문의 Round-Trip Translator(RTT)는 객체지향 프로그래밍을 학습하기 위한 웹 기반 교육 프로그래밍 도구로 별도로 설치하지 않고 URL로 접근하여 이용할 수 있다. 만일 학습자가 각기 다른 환경에서 기존 도구를 사용한다면 설치 과정이 필요하지만 RTT를 이용하면 웹을 통해 어느 환경에서나 동일한 학습 환경을 제공받을 수 있다[4].

사용자는 자신이 작성한 소스코드가 어떻게 객체지향적인 개념을 반영하는지 UML을 통해 실시간으로 확인할 수 있으며 UML을 그린 뒤, 이를 변환함으로써 소스코드를 웹에서 바로 확인할 수 있다.

2.1 RTT 사용 프로세스

Fig. 1은 RTT의 프로세스를 보여준다. 소스코드를 UML로 바꾸는 방법은 사용자가 RTT 에디터에 자바 소스코드를 입력한 뒤 저장하고 변환 버튼을 누르면 브라우저 다운로드 경로에 자바 파일이 생성된다. 생성된 파일을 자바 코드 분석을 통해 생성된 JSON 텍스트를 GoJS 라이브러리로 변환한다. 그리고 GoJS 라이브러리를 사용하여 UML을 그린다.

UML을 소스코드로 변환하는 과정은 소스코드를 UML로 바꾸는 방법의 역순이다. 해당 UML의 정보가 GoJS에 의하여 JSON 형태로 바뀐 UML 정보를 토대로 소스코드를 생성하는데 필요한 정보들을 추출하여 자바 파일을 생성한다.

2.2 RTT 기술 아키텍처

RTT 아키텍처는 Fig. 2에서 볼 수 있듯이 크게 3가지 부분으로 나누어져 있다.

1) Front-End

Front-End에서는 UML을 그리거나 자바 소스코드를 작성한다. Javascript와 jQuery를 사용하여 서버와 통신한다.

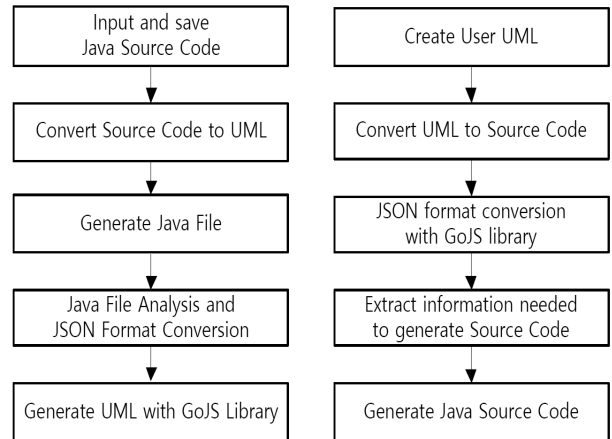


Fig. 1. Process of RTT

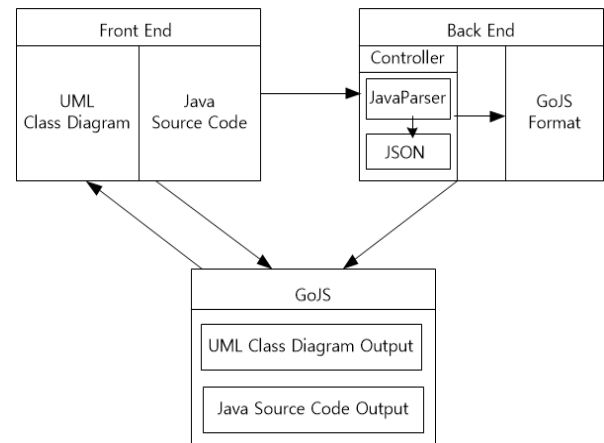


Fig. 2. Architecture of Round-Trip Translator

2) Back-End

Back-End에서는 자바 분석 기법을 사용하여 소스코드를 분석한다. 분석한 내용을 GoJS 라이브러리 형태에 맞는 JSON 형식으로 변환하여 클라이언트에게 보낸다.

3) GoJS Library

GoJS는 UML을 생성하거나 자바 소스코드를 만드는데 사용된다. 서버로부터 응답받은 JSON을 토대로 UML을 그린다. UML 객체를 클릭했을 때는 JSON 정보를 읽어서 이를 분석한다. RTT는 분석한 내용을 자바 파일로 생성하고 이것을 에디터에 보여준다.

3. 기존 도구(StarUML)와 RTT 비교

3.1 StarUML과 RTT에 대한 구조적 비교

Fig. 3과 Fig. 4는 StarUML에서의 UML과 자바 소스코드가 상호 변환된 결과를 보여준다.

학습자가 UML을 그린 뒤 소스코드를 확인하려면 확장 프

로그를 설치해야 하며 'Generate Code' 버튼을 클릭하여 자바 파일을 생성한 뒤 해당 다운로드 경로를 찾아서 파일을 확인해야 한다. 반대로 소스코드로부터 UML을 생성할 때는 해당 소스코드의 폴더 경로를 확인한 다음 'Reverse Code' 버튼을 통해 UML을 생성한다.

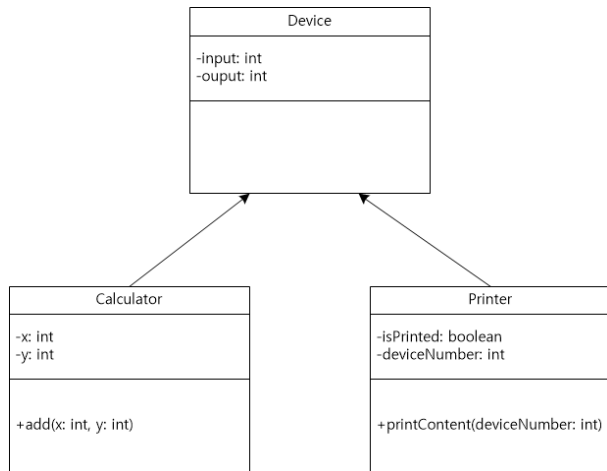


Fig. 3. Example of StarUML UML Class Diagram

```

public class Printer extends Device {
    /**
     * Default constructor
     */
    public Printer() {
    }

    /**
     *
     */
    private boolean isPrinted;

    /**
     *
     */
    private int deviceNumber;

    /**
     * @param deviceNumber
     */
    public void printContent(int deviceNumber)
    {
        // TO DO implement here
    }
}
    
```

Fig. 4. Example of StarUML Java Source Code Output

생성된 UML에 대한 속성과 함수를 보기 위해서는 해당 객체의 설정을 변경해야 상세한 내용을 볼 수 있다. Fig. 3은 StarUML 도구를 사용해서 그린 UML이고 Fig. 4는 Fig. 3의 UML을 소스코드로 변환시킨 결과이다.

Fig. 5, Fig. 6과 동일한 클래스 기반으로 RTT에 의해 생성된 결과를 보여준다.

Class Diagram

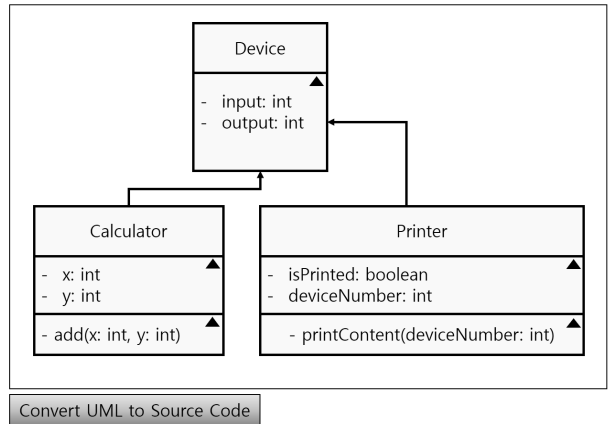


Fig. 5. Example of RTT UML Class Diagram

Source Code

```

Public class Printer extends Device {
    private boolean isPrinted;
    private int deviceNumber;
    public void printContent(int deviceNumber) {
        // TO DO implement here
    }
}
    
```

Replace Diagram

Fig. 6. Example of RTT Java Source Code Output

학습자는 웹에서 에디터의 텍스트를 다운로드하기 때문에 브라우저의 다운로드 경로를 설정해야 한다. UML을 그린 뒤 해당 객체를 클릭하면 에디터에서 객체의 정보가 담긴 자바 소스코드를 확인할 수 있다. 소스코드를 UML로 변환하는 경우, 학습자는 코드를 작성한 뒤 에디터 하단의 'Replace Diagram' 버튼을 통해 소스코드가 반영된 UML을 그릴 수 있다.

Fig. 5는 RTT로 생성한 UML과 해당 UML을 소스코드로 변환시킨 결과이고 Fig. 6은 Fig. 5의 UML을 소스코드로 변환시킨 결과이다.

3.2 StarUML과 RTT에 대한 기능적 비교

1) 단순성

RTT는 사용자 편의성을 위해 직관적인 사용자 인터페이스를 제공한다. 직관적인 인터페이스란 사용자가 암기, 실험, 훈련 없이 제품의 사용법을 이해할 수 있는 것을 의미한다[5, 6]. RTT를 이용하면 별도의 학습과정 없이 직관적인 인터페이스

이스를 통해 객체지향 학습에 집중할 수 있고 UML과 소스 코드를 한눈에 확인할 수 있다.

2) 편집 용이성

StarUML은 드래그 앤드 드롭을 통해 UML에서 다른 UML로 속성과 함수를 옮길 수 없다. 그러나 RTT는 서로 다른 UML 요소끼리 속성과 함수를 쉽게 이동할 수 있다.

Fig. 7의 Account의 'validateUser'함수를 다른 UML로 드래그하여 이동하는 RTT의 드래그 앤드 드롭 프로세스를 보여준다.

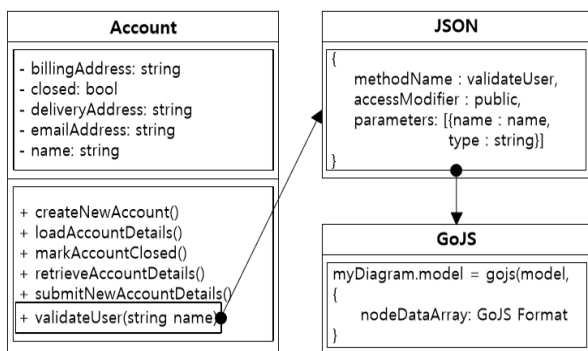


Fig. 7. The Drag and Drop Process of RTT

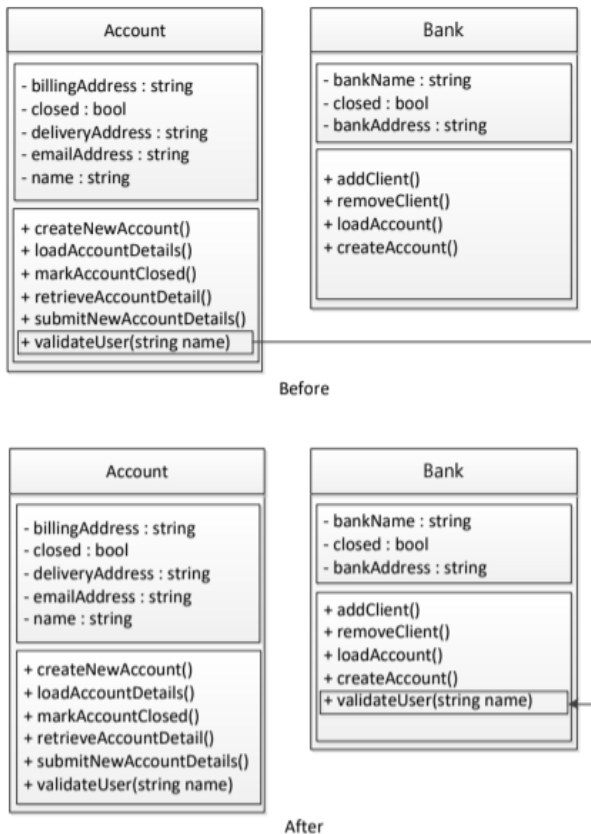


Fig. 8. Before and After Moving Properties in UML

사용자는 UML을 클릭하여 이동하고 싶은 객체의 JSON 정보를 볼 수 있다. JSON에서 GoJS 라이브러리 포맷으로 수정한 후에 UML을 다시 그리면 선택한 속성이나 함수가 다른 UML으로 이동한 것을 볼 수 있다.

Fig. 8은 기존의 Account에 속해 있는 'validateUser'함수의 모습과 드래그 앤드 드롭 기능을 사용하여 Account의 'validateUser'함수를 Bank로 이동한 결과를 보여준다.

드래그 앤드 드롭 기능을 사용함으로써, 사용자는 속성과 함수를 더 직관적으로 수정할 수 있다.

3.3 StarUML과 RTT에 대한 정성적 비교

StarUML은 학습자가 UML 혹은 소스코드를 작성한 뒤 생성된 결과물을 확인하기 위해 미리 설정한 다운로드 경로로 이동해야 해당 소스코드를 볼 수 있다. 소스코드를 UML로 변환하는 경우엔 학습자는 복잡한 설정들을 선택하여 UML을 생성해야 한다. 반면, RTT는 에디터에서 소스코드를 확인 및 수정할 수 있다. 이를 비교하면 Table 1과 같다.

Table 1. Comparison of StarUML and RTT

	Output Check	Settings	UML, Source Code Modification Method
StarUML	On File Path specified in StarUML	Multiple settings required	On File Path With IDE and StarUML
RTT	On Web	Download path required	Directly on Web with Web Editor

학습자가 객체지향 프로그래밍을 학습하는 데 있어 기존의 도구보다 RTT가 편리하다는 것을 알 수 있다. 직관적인 사용자 인터페이스를 사용함으로써 초기 사용자의 진입장벽을 낮추고 학습자가 객체지향 프로그래밍을 학습하는데 집중할 수 있는 환경을 제공한다.

4. 기존 RTE 도구와 RTT의 비교 분석

기존 RTE를 지원하는 도구들에는 Visual Paradigm, StarUML, Papyrus가 있다. 위 도구들은 UML 모델링을 지원하며 UML에서 자바 코드로 변환해 주는 RTE 기능을 제공한다.

1) Visual Paradigm(VP)

Visual Paradigm은 UML2, SysML 및 BPMN(Business Process Modeling Notation)을 지원하는 UML 도구다. 모델링 지원 외에도 코드 생성을 포함하여 보고서 생성 및 코드 엔지니어링 기능을 제공한다.

2) StarUML(SU)

StarUML은 간결한 모델링을 지원하기 위한 정교한 소프트웨어 모델이다. 빠른 모델링, 코드 엔지니어링, 다양한 서드 파티 확장을 지원하며 ERD(Entity Relationship Diagrams), Flowchart Diagram 등을 추가 지원한다.

3) Papyrus(PR)

Papyrus는 UML2, SysML을 지원하는 Eclipse 표준 모델링 도구이다. 주요 기능은 UML 모든 다이어그램을 지원 및 모델링 파일의 분할 및 설치이다.

Table 2는 기존의 RTE 기반 기존 도구들과 RTT를 여러 기준을 가지고 비교한 테이블이다.

Table 2. Round-Trip Engineering Tools Comparison with RTT

Parameters/ Tools	VP	SU	PR	RTT
Support for Editor	X	X	O	O
Download	O	O	O	X
Cost	Not Free	Free	Free	Free
Open source	X	O	X	O

RTT는 에디터를 포함하고 있어 사용자는 소스코드를 바로 수정하여 UML에 반영시킬 수 있다. 그러나 Visual Paradigm은 에디터와 UML을 한눈에 볼 수 없고 StarUML은 에디터가 없다. 학습자가 UML을 그려서 생성한 코드를 수정하려면 직접 자바 파일을 열어서 수정해야 한다.

RTT는 웹 기반 프로그래밍 환경을 제공함으로써 접근의 용이성을 제공한다. 반면 기존 RTE 도구들은 로컬 환경에서만 사용하기 때문에 접근성이 떨어진다.

5. 결 론

본 논문에서는 별도의 다운로드나 설치 없이 사용 할 수 있는 웹 기반의 RTT를 제안했다. 제안된 RTT는 객체지향 개념을 학습할 수 있는 기존의 RTE 기반 도구의 기능을 확장하고, 직관적인 사용자 인터페이스를 설계하고 구현하였으며 UML과 소스코드를 바로 수정하여 반영할 수 있도록 에디터를 내장하고 속성과 함수에 대한 드래그 앤드 드롭을 지원하여 기존 RTE 도구들보다 편집이 용이하다.

기존의 RTE 기반의 도구들과 RTT를 비교함으로써 제안

한 도구를 사용하여 객체지향을 학습하는 것이 기존 도구들을 사용하여 학습하는 것보다 효율적이고 편리하다는 결론을 도출했다. 향후 자바 언어 뿐만 아니라 C++, Python과 같은 객체지향 언어들을 RTT에 포함시켜 학습자가 보다 다양한 언어로 객체지향 개념을 공부할 수 있도록 할 것이다.

References

- [1] Hettel, Thomas, Michael Lawley, and Kerry Raymond, "Model synchronisation: Definitions for round-trip engineering," *International Conference on Theory and Practice of Model Transformations*, Springer, Berlin, Heidelberg, 2008.
- [2] M. Sivasakthi, and R. Rajendran, "Learning difficulties of 'object-oriented programming paradigm using Java': students' perspective," *Indian Journal of Science and Technology*, Vol.4, No.8, pp.983-985, 2011.
- [3] Myers, Brad A., "User interface software tools," *ACM Transactions on Computer-Human Interaction (TOCHI)*, Vol.2, No.1, pp.64-103, 1995
- [4] Mackay, Daniel, James Noble, and Robert Biddle, "A lightweight web-based case tool for UML class diagrams," *Proceedings of the Fourth Australasian User Interface Conference on User Interfaces 2003-Volume 18. Australian Computer Society, Inc.*, 2003.
- [5] Galitz, Wilbert O., "The essential guide to user interface design: an introduction to GUI design principles and techniques," John Wiley & Sons, 2007.
- [6] Everett N. McKay, "UI is Communication : How to Design intuitive, user-centered interfaces by focusing on effective communication," San Francisco, CA: Morgan Kaufmann Publishers, 21-22.



김지용

<https://orcid.org/0000-0002-4260-4043>

e-mail : jyk@soongsil.ac.kr

2019년 숭실대학교 소프트웨어학부(학사)

2019년~현 재 숭실대학교

융합소프트웨어학과 석사과정

관심분야 : Network & Cloud Computing



조 한 주

<https://orcid.org/0000-0002-6327-9825>
e-mail : hanjoo96@soongsil.ac.kr
2019년~현 재 송실대학교
소프트웨어학부 학사과정
관심분야: Network & Cloud Computing



김 영 종

<https://orcid.org/0000-0003-0811-0215>
e-mail : youngjong@ssu.ac.kr
1996년~1998년 한글과컴퓨터 연구원
2005년~2007년 시큐어소프트 기술이사
2006년~2007년 하우리 대표이사
2007년~2009년 오픈소스커뮤니티연구소 소장 및
열린사이버대학교 정보지원처장
2019년~현 재 송실대학교 소프트웨어학부 부교수
관심분야: Security, Blockchain, Network & Cloud Computing