

한글 프로그래밍 언어 '새싹'의 코드 시각화 방안 연구

A Research on Source Code Visualization of 'Saesark,' a Korean Programming Language

천준석*, 우균**

부산대학교*, 부산대학교/LG전자 스마트제어센터**

Junseok Cheon*, Gyun Woo**

Pusan National Univ.*, Pusan National Univ.,
Smart Control Center of LG Electronics,**

요약

오늘날 사용되고 있는 소프트웨어의 규모는 옛날과는 비교가 안 될 정도로 커졌다. 또한, 소프트웨어는 여러 개발자의 협업을 통해 만들어지게 된다. 이러한 개발 구조는 코드 이해를 도와주는 '방법'의 필요성을 요구하게 되었다. 코드 시각화는 그 방법의 하나로, 커다란 코드를 도형 등을 이용해 쉽게 이해할 수 있도록 한다. 이러한 시각화 방법은 널리 사용되고 있는 언어에 대해서는 많은 방법이 제공되고 있지만, 한글 프로그래밍 언어에 대해서는 제공되고 있는 방법이 없다. 이 논문에서는 한글 프로그래밍 언어인 '새싹'을 대상으로 하는 코드 시각화 방법을 제안한다.

I. 서론

개발자가 커다란 소프트웨어를 개발할 때에는 어떠한 방법으로 해야 할까? 작은 규모의 프로그램을 개발할 경우에는 개발자의 머릿속의 설계도를 토대로 소프트웨어를 개발하면 된다. 하지만 소프트웨어의 규모가 커지면, 여러 개발자가 팀을 구성해서 개발해야 하므로 다른 사람과의 의사소통이 필요하다.

이러한 의사소통이 쉬워지도록 하는 대표적인 방법이 코드 시각화이다. 코드 시각화는 소스 코드를 도형 등의 직관적인 방법으로 표현하는 것을 말한다. 대표적인 코드 시각화 방법에는 흐름도, UML 등이 있다.

한편, 최근에는 세계적으로 소프트웨어 교육 열풍이 불고 있다. 이에 따라 우리나라도 소프트웨어 조기 교육을 시행하고 있다. 하지만 프로그래밍 언어 대부분이 영어로 되어있다는 점은 초보자의 진입 문턱을 높이는 가장 큰 원인이다.

지난 연구에서 우리는 한글 객체지향 프로그래밍 언어 '새싹'을 개발하였다. '새싹'은 초보자가 쉽게 프로그래밍을 시작할 수 있도록 한글 어순과 비슷한 문법을 사용한다. 또한, Java 언어를 기초로 하여 개발되었기 때문에 나중에 C++이나 Java와 같은 주요 언어를 사용할 때에 쉽게 적응할 수 있다.

본 논문에서는 새싹 언어를 UML을 사용하여 시각화하는 방법을 제안한다. 초보자가 프로그래밍을 통해 문제 해결을 할 때도 시각화를 통해 프로그램의 전체 구조를 쉽게 파악할 수 있다. 또한, 나중에 협업을 통해 대규모 소프트웨어 개발에도 도움을 줄 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 새싹 언어와 UML에 대해 설명한다. 3장에서는 제안하는 시스템에 대해 설명한다. 이후 4장에서 결론을 맺는다.

II. 관련 연구

1. 한글 프로그래밍 언어 '새싹'

새싹은 부산대학교 프로그래밍언어연구실에서 개발한 한글 프로그래밍 언어이다[1]. 새싹은 처음에 C를 참고하여 절차형 프로그래밍 언어 형태로 개발되었다. 새싹은 JVM(Java Virtual Machine) 상에서 동작하기 때문에 JVM이 동작하는 환경이라면 어디에서나 동작하는 장점이 있다. 또한, 한국인이 사용하기에 어색하지 않도록 문법을 적용하였다.

새싹은 이후 절차형 언어에서 객체지향 언어로 바뀐 버전으로 개발되었다[2]. 새싹은 한글로 배울 수 있는 객체지향 언어이기 때문에 객체지향 개념을 쉽게 이해할 수 있는 장점이 있다. 또한, 오류 메시지를 한글로 출력하게 하여 코딩 초보자도 오류를 쉽게 수정할 수 있도록 도와준다. 새싹은 또한 람다 식을 지원하여 코드 길이가 줄어들고 더 명확하게 할 수 있도록 돕는다.

2. UML

소프트웨어의 시각화 방법은 흐름도(flowchart)나 ER 다이어그램을 시작으로 하여 여러 방법론이 소개되었다. 객체지향 언어가 주목받기 시작한 90년대 전후로는 객체지향 언어의 분석 및 설계에 대한 여러 방법론이 등장하였다. Grady Booch가 개발한 Booch method와 James Rumbaugh가 개발한 OMT가 유명하다[3].

하지만 객체 지향 시각화 방법론마다 그림을 그리는 방법이 서로 달라서 실제 사용 시 혼란을 유발하였다. 클래스의 경우, OMT에서는 직사각형으로 그려진다. 이에 반해, Booch method에서는 구름 형태로 그려진다.

이러한 방법론의 통일을 위해 규정된 언어가 UML(Unified Modeling Language)이다. 1994년에 Grady Booch와 James Rumbaugh가 손을 잡고 통일을 위한 협력을 개시하였다. 이후 1995년에 Ivar Jacobson이 여기에 가세하였고, 1997년에 UML 1.0이 발표되었다. 이후 발전을 거듭하여 2005년 11월에는 UML 2.0이 발표되었다.

우리는 UML 기본 표현 중의 하나인 클래스 다이어그램을 이용하여 새싹 코드를 시각화하려고 한다. 클래스 다이어그램은 UML의 기본 표현 중의 하나로, 시스템을 구성하는 클래스와 그 상호관계를 나타낸다. 클래스 간의 관계는 도형과 화살표를 이용해서 표현한다.

III. 시스템 구현

본 논문에서 제안하는 새싹 언어의 UML 형태 코드 시각화 전체 구조는 그림 1과 같다. 먼저 새싹 소스코드를 파싱하여 중간 표현인 Java 소스코드로 변환한다. 파싱된 Java 소스코드는 ObjectAid UML Explorer를 이용하여 UML로 출력한다⁴⁾. 이 과정을 통해 새싹 코드를 UML로 출력하게 된다.



▶▶ 그림 1. 새싹 언어 시각화 시스템의 전체 구조

새싹 소스코드를 파싱하기 위해 파서를 개발하였는데, 파서는 JavaCC를 이용하여 구현하였다⁵⁾. 이 과정은 새싹 문법을 Java 문법에 맞게 변환한다. 이를 위해 JavaCC를 이용해 AST를 생성하였고, 방문자 패턴을 이용해서 Java 소스코드로 출력하였다.

ObjectAid UML Explorer는 Eclipse 플러그인 형태로 개발된 Java 소스코드 시각화 도구이다. Java 소스코드를 입력받아 클래스 다이어그램 또는 시퀀스 다이어그램 형태의 출력물을 생성한다. 결과물은 드래그&드롭을 이용하여 적절한 배치가 가능하다.

제안하는 시스템을 이용하여 간단한 새싹코드를 UML로 출력해보았으며, 그 결과는 그림 2와 같다. 그림 2의 코드는 점점(Java의 인터페이스) 가와 나의 정수형(Java의 int형) 변수 ㄱ의 값을 출력하는 예제이다. 이 코드를 파싱하여 Java 코드를 생성하게 된다.

```

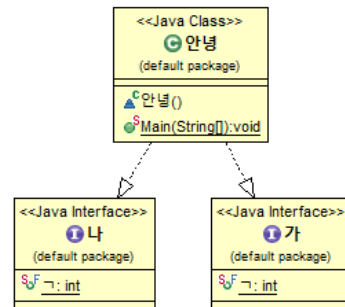
점점 가 {
    정수 ㄱ = 10;
}

점점 나 {
    정수 ㄱ = 100;
}

클래스 안녕 시행 가, 나 {
    시작(문자열[] 값) {
        출력("%수\값", ㄱ.ㄱ);
        출력("%수\값", 나.ㄱ);
    }
}
  
```

▶▶ 그림 2. UML 생성을 위한 간단한 새싹 코드

그림 3은 그림 2의 새싹 코드를 클래스 다이어그램 형태로 시각화한 결과이다. 안녕 클래스와 가, 나 점점 사이의 관계를 선과 삼각형을 사용해서 나타낸 것을 확인할 수 있다. 이는 클래스와 인터페이스 간의 관계이므로 점선으로 표현되었으며, 상속 관계이기 때문에 안이 비어있는 삼각형으로 나타난 것을 확인할 수 있다. 하지만 Java 언어 형태로 출력되는 한계점이 있다.



▶▶ 그림 3. 제안한 시스템을 통한 새싹 코드 시각화 결과

IV. 결론

본 논문에서는 새싹 코드를 UML 형태로 시각화하는 시스템 구현을 목표로 하였다. 구현한 시스템은 새싹 코드를 파싱하여 중간 표현인 Java 코드로 변환한다. 이후 ObjectAid UML Explorer를 통해 UML 형태로의 시각화를 수행한다. 초보자가 새싹을 이용해서 프로그래밍할 때, UML 형태의 시각화를 통하여 자신이 개발한 프로그램을 한눈에 파악할 수 있으므로, 문제 해결 능력이 향상될 것으로 기대된다.

향후 연구로 UML 형태의 시각화 결과를 Java 형식이 아닌 새싹 형식으로 출력하는 것에 관해 연구하고자 한다. 현재 개발된 시스템은 Java 형태로 출력되기 때문에 Java 언어를 잘 모르는 초보자가 적용하기 어려운 한계가 있다. 따라서 ObjectAid UML Explorer를 대신할 수 있는 Eclipse 플러그인 형태의 UML 생성 도구를 개발하여 좀 더 보기 쉬운 시각화 결과를 얻을 수 있도록 하겠다.

ACKNOWLEDGMENT

이 논문은 2013년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(NRF-2013R1A1A4A01006704)

* 교신 저자 : 우균(부산대학교, woogyun@pusan.ac.kr)

■ 참고 문헌 ■

- [1] 천준석, 강도훈, 김건우, 우균, "간결한 한글 프로그래밍 언어 '새싹'", 한국정보과학회논문지, 제42권, 제4호, pp.496-503, 2015.
- [2] 천준석, 우균, "새싹: 초보자를 위한 한글 객체 지향 프로그래밍 언어," 한국콘텐츠학회논문지, 제16권, 제3호, pp.288-295, 2016.
- [3] P. Graessle, H. Baumann and P. Baumann, UML 2.0 in Action: A project-based tutorial, PACKT Books, 2005.
- [4] ObjectAid, <http://www.objectaid.com/> (16.04.27 방문)
- [5] 아오키 미네로우, 컴파일러 구조와 원리, 한빛미디어, 2010.