

# 이클립스 기반의 교육용 컴파일러 통합개발환경

성우경<sup>†</sup> · 강현석<sup>††</sup> · 배종민<sup>†††</sup>

## 요 약

컴파일러 교과목에서 수행하는 컴파일러 개발 프로젝트는 많은 경험과 기술을 얻을 수 있다. 그러나 한 학기에 이수하기 부족한 강의 시간과 컴파일러 개발의 높은 난이도 때문에 수업에 어려움을 겪는다. 그리고 컴파일러 목적시스템은 대개 인터프리터로 구현되기 때문에 학생의 흥미를 유도하기도 어렵다. 이에 따라서 컴파일러 교육은 이론 위주의 교육이 되기 쉽다. 이러한 한계를 극복하기 위하여 본 논문에서는 컴파일러의 이론과 실체를 보다 쉽게 습득할 수 있는 통합개발환경을 제시한다. 개발된 통합개발환경에는 마인드스톰 NXT 로봇을 목적시스템으로 하는 레퍼런스 컴파일러와 컴파일러 제작 도구, 목적인어 테스트 도구, 코드생성 시각화 도구가 포함되며 이클립스 플러그인 기반으로 개발되어 편의성과 확장성이 뛰어나다. 개발된 통합개발환경은 학생들이 컴파일러를 보다 쉽게 이해하고 개발할 수 있도록 도와준다.

**주제어** : 교육용 컴파일러, 컴파일러 개발, 통합개발환경, 이클립스, 이클립스 플러그인

## Development of an Eclipse-based IDE for Educational Compilers

U-Kyung Sung<sup>†</sup> · Hyun-Syug Kang<sup>††</sup> · Jong-Min Bae<sup>†††</sup>

## ABSTRACT

Compiler development projects, which are designed and taught in compiler course, allow students to practice and absorb valuable amount of experience and techniques in developing compilers. However, both instructors and students face difficulties as they are often limited by insufficient hands-on time during course of an academic year along with a relatively high level of technologies involved when dealing with compilers. As well, most compiler's target systems use interpreter-based technologies which are rather limited in drawing student's attention. As a result compiler courses often end up being more of a theoretical course than practical. This paper presents a new integrated development environment (IDE) that will help overcome aforementioned difficulties and allow students to obtain both theoretical and practical knowledge more efficiently. The development environment includes a reference compiler with Mindstorms® NXT Robots as the target system, compiler development tool, target language test tool, and code generation visualizer. It is developed as a plug-in for the popular Eclipse IDE which enables easy access and great expandability. This integrated development environment allows students to understand compilers better and start their development faster.

**Keywords** : Educational Compiler, Compiler Development, Eclipse, Eclipse Plug-in

<sup>†</sup> 준 회 원: 경상대학교 대학원 컴퓨터학과

<sup>††</sup> 정 회 원: 경상대학교 컴퓨터학과 교수

<sup>†††</sup> 정 회 원: 경상대학교 컴퓨터학과 교수(교신지자)

논문접수: 2011년 07월 26일, 심사완료: 2011년 09월 23일, 게재확정: 2011년 09월 26일

## 1. 서론

컴파일러 개발은 프로젝트 경험과 종합적인 알고리즘 사용법을 통해 광범위한 지식과 기술을 제공한다. 때문에 컴파일러 교과목은 오래 전부터 ACM Computing Curricula에서 정착된 과목이다.[1] 하지만 컴퓨터과학 교과과정 중에서 컴파일러 교과목은 어려운 교과목에 속한다.

부족한 강의 시간과 컴파일러 개발의 높은 복잡도 때문에 학생과 강사, 모두에게 컴파일러 교과목에서 수행하는 컴파일러 개발 프로젝트는 부담이 된다. 그리고 컴파일러 목적시스템은 대개 인터프리터를 구현하여 사용한다. 이는 컴파일러 동작을 보여주는데 현실적이지 못하여 학생의 흥미를 유도하는데 어려움이 있다. 이러한 이유로 인하여 컴파일러 개발 프로젝트 수행에 한계가 있어, 이론 위주의 교육이 되기 쉽다.

이론 위주의 컴파일러 교육을 벗어나 이론과 실제, 모두 효과적으로 교육하기 위해서는 세 가지 요건이 필요하다.

첫째, 이론중심의 컴파일러 수업 보다는 실험실습 프로젝트 위주의 수업 모형이 필요하다.

둘째, 학생의 흥미를 유발할 수 있는 동기 부여가 필요하다.

셋째, 컴파일러 개발의 어려움을 극복할 수 있는 환경이 필요하다.

이러한 요건을 만족시키기 위해서는 컴파일러 교과목을 강의할 목적의 대상 환경과 개발도구가 필요하다.

컴파일러 개발 프로젝트에 대한 흥미를 유도하기 위해서는 목적시스템을 인터프리터로 실행하는 것 보다는 실제로 구동되는 기계를 대상으로 하는 것이 바람직하다. 레고(Lego)사의 마인드스톱 NXT 로봇은 이러한 의미에서 컴파일러의 목적시스템으로 적합하다고 할 수 있다. 마인드스톱 NXT 로봇은 컴퓨터를 내장한 조립용 브릭으로 로봇 제작을 간단한 조립과 프로그래밍으로 가능하게 한다. 마인드스톱 NXT 로봇은 NBC 중간언어를 처리할 수 있으며 미리 정의된 입출력 API(Application Programming Interface)를 이용하여 다양한 목적의 로봇을 구동할 수가 있어 컴파일러의 대상 환경으로 적합하다. 마인드스톱

NXT 로봇을 대상으로 제작된 컴파일러의 동작은 실제로 구동되는 로봇으로 확인할 수 있기 때문에 컴파일러 개발 프로젝트를 통한 교육 효과를 높일 수 있다.

본 논문에서는 마인드스톱 NXT 로봇을 목적시스템으로 하는 교육용 컴파일러 개발에 적합한 개발도구를 설계하고, 이를 통합개발환경 형태로 제시한다. 통합개발환경(Integrated Development Environment)은 프로그램의 편집과정과 컴파일 과정 및 디버깅 과정을 하나의 환경에서 모두 지원해주는 강력한 개발도구이다. 본 논문에서 제시하는 통합개발환경은 다음과 같은 기능을 통하여 효과적으로 컴파일러 개발 프로젝트를 지원한다.

첫째, 이클립스 플러그인 프레임워크 기반으로 구현한다. 이는 이클립스 통합개발환경이 동작하는 모든 플랫폼에서 사용이 가능하며, 실행 및 배포가 용이하다.

둘째, 통합개발환경을 사용하는 학생의 접근성을 높일 수 있도록 GUI(Graphic User Interface)를 기반으로 설계한다.

셋째, 학생이 수행하는 컴파일러 개발 프로젝트에 참고할 수 있는 레퍼런스 컴파일러를 제공한다. 레퍼런스 컴파일러는 교육목적으로 사용되기 때문에 로봇을 제어하는 C언어 기반의 간단한 원시언어와 이해하기 쉬운 목적언어로 설계하여, 어휘분석기, 구문분석기, 코드생성기를 포함하여 구현한다.

넷째, 마인드스톱 NXT 로봇을 통하여 목적언어를 실행하고 실행결과를 확인할 수 있는 목적언어 테스트 도구를 제공한다.

다섯째, 원시언어와 목적언어의 관계를 시각적으로 표현하여 코드생성 과정의 이해를 돕는 코드생성 시각화 도구를 제공한다.

우리는 위와 같은 기능을 제공하는 교육용 컴파일러 통합개발환경을 통하여 컴파일러 개발 실습 강의를 돕는 역할을 한다. 본 논문의 구성은 다음과 같다. 2장에서 관련 연구를 살펴보고, 3장에서 교육용 컴파일러 통합개발환경의 설계방향을 설명한다. 4장에서는 교육용 컴파일러 통합개발환경을 설계와 구현을 제시하며, 5장에서는 이를 교육에 적용한 사례와 효과를 마지막으로 6장에서는 결론을 내리고 향후과제를 제시한다.

## 2. 관련 연구

Chirp on Crickets는 내장형 로봇 컨트롤러를 사용해서 컴파일러를 교육하는 모델을 제시하여 저학년에서 컴파일러 프로젝트가 가능하도록 유도하였다.[2] 컴파일러의 원시언어로 C 언어 기반 교육용 언어인 Chirp 설계하였고, 교육용 로봇 컨트롤러 Crickets를 목적 언어로 선정하였다. 추가로 Crickets 시뮬레이터도 같이 제공하고 있어 Crickets 로봇 컨트롤러가 없는 환경에도 Chip on Crickets 컴파일러를 사용할 수 있다.

RobotStudio는 이클립스 RCP(Rich Client Platform)로 구현된 로봇 기반의 언어공학 시스템이다.[3] RobotStudio는 대형 소프트웨어 개발을 학습하기 위한 방법으로 RobotStudio를 활용한 통합개발환경 기반 접근법을 제시하였다. 이는 이클립스 RCP 플랫폼을 기반으로 하여 학생들을 RobotStudio 확장점(Extension Point) 명세에 따라서 프로젝트를 수행한다. 제시한 두 시스템은 본 논문의 시스템과 가장 유사한데, NXT 환경에 비하여 학생들의 흥미를 쉽게 유도하기 어렵고, 여러 가지 지원기능이 미약한 편이다.

Bantam 컴파일러는 컴파일러를 교육하기 위한 목적의 컴파일러이다.[4] Bantam Java 언어를 설계하고 교육용 컴파일러 개발을 통하여 컴파일러 과목을 학습하는 방법을 제시하였다. 대상 플랫폼을 MIPS와 X86으로 선정하여 교육용 컴파일러를 개발하였으며, 학생에게 컴파일러 개발 매뉴얼을 제공하여 컴파일러 학습에 활용하였다. 자바 컴파일러 개발을 위한 지원도구라는 점 이외에는 학생들의 어려움을 덜어 줄 수 있는 기능이 미약하다.

한편 학생이 저급언어에 익숙하지 않기 때문에 코드생성 과정에는 많은 어려움이 존재한다. 때문에 Sondag 외 2명은 코드생성의 이해를 돕는 도구인 Frances를 제시하였다.[5] Frances는 원시언어와 목적(저급)언어의 관계를 시각적으로 표현하여 코드생성 과정의 어려움을 해결할 수 있게 도움을 준다. 본 논문은 이러한 시스템들에서 가진 모든 기능을 지원하면서 더욱 강력하고 편리한 기능을 포함한다.

## 3. 설계 방향

교육용 컴파일러 통합개발환경은 아래와 같은 기준을 만족하도록 설계되었다.

첫째, 하드웨어, 운영체제 플랫폼에 독립적이어야 한다. 이를 위하여 본 연구에서는 자바언어로 개발하여 자바 가상 기계(Java Virtual Machine) 기반에서 플랫폼에 독립적으로 실행되도록 한다.

둘째, 기능에 따른 모듈 구현에 대한 확장성과 독립성이 있어야 한다. 이를 위해 이클립스 플러그인 기반으로 컴파일러 개발도구를 구현하고 이에 대한 확장점으로 정의하여 추가되는 기능 간의 독립성과 개발도구의 확장성을 보장한다.

셋째, 학생들에게 친숙한 개발도구의 유저인터페이스를 따른다. 교육용 컴파일러 개발도구는 이클립스 통합개발환경의 유저인터페이스를 그대로 따르게 함으로써 이클립스 통합개발환경을 미리 경험했던 학생들에게 좀 더 익숙한 컴파일러 개발환경을 제공한다.

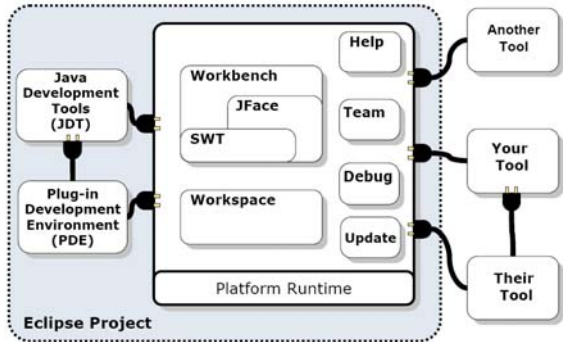
넷째, 파서 제작도구인 LEX와 YACC을 사용해야 한다. 이를 반영하여 컴파일러 개발도구는 LEX와 YACC에 호환성으로 가지며 자바언어를 대상으로 하는 파서 제작도구인 JFLEX와 CUP을 지원한다.[9][10]

다섯째, 목적 언어를 실행하고 테스트할 수 있는 환경을 제공한다. 컴파일러 개발도구를 사용해서 만들어진 컴파일러가 목표로 하는 실행 플랫폼은 마인드스톱 NXT 로봇이다. 컴파일러가 올바르게 코드생성을 하는지 평가하기 위해서는 마인드스톱 NXT 로봇을 통해 생성된 코드를 테스트를 할 수 있는 기능이 요구된다. 마인드스톱 NXT 로봇을 실행 할 수 있는 모듈을 교육용 컴파일러 통합개발환경에 내장하고, 마인드스톱 NXT 로봇이 없어도 생성된 코드를 실행 및 테스트할 수 있는 마인드스톱 NXT 로봇 에뮬레이터도 같이 제공하고 있다.

### 3.1 이클립스 기반의 통합개발환경

이클립스는 자바를 비롯한 다양한 언어를 쉽게 지원하는 통합 개발 환경이다. 이클립스는 1999년에 IBM에서 개발되기 시작하여 2001년에는 이클

립스 컨소시엄을 설립하고 오픈소스 프로젝트로 시작되었다.[6]



<그림 1> 이클립스 플랫폼 아키텍처

<그림 1>은 이클립스 플랫폼이다. 이클립스 플랫폼은 SWT, JFace, 플랫폼 런타임(Platform Runtime), 워크스페이스(Workspace/Resources), 워크벤치(Workbench) 등의 컴포넌트로 구성된다. <그림 1>에서 소켓을 통하여 이클립스 플랫폼과 결합한 컴포넌트를 플러그인(plug-in)이라 한다. 이클립스의 모든 기능은 플러그인에서 수행되며, 다른 플러그인 제공하는 기능을 이용하거나 또 다른 플러그인과 통합되어 기능을 제공한다. 이클립스는 이러한 플러그인 프레임워크를 활용하여 다양한 언어의 개발도구로 확장할 수 있다.

### 3.2 교육용 로봇 플랫폼

#### 3.2.1 마인드스톰 NXT 로봇

만약 컴파일러의 목적코드가 실제 기계어라면 코드생성과정은 매우 복잡하여 교육용으로는 적합하지 않다. 그래서 대부분 가상기계를 설정하고 그 가상기계에 대한 인터프리터를 구현하여 목적코드 생성의 정확성을 확인한다, 이것은 학생들에게 전혀 흥미를 줄 수가 없다.

마인드스톰 NXT 로봇은 레고사에서 개발한 프로그래밍 가능한 인텔리전트 브릭이다.[7] NXT는 NBC[8]라는 저급언어가 지원되고 NXT는 NBC를 수행할 수 있는 도구를 가지고 있기 때문에 가상기계나 인터프리터의 필요 없이 로봇이 그 기능을 대신하므로 생성된 목적코드가 실행되는 과정을 더 실감나게 볼 수 있다. 따라서 본 시스템의 목적코드는 NBC로 정하고 NXT 로봇이 이를 실행시키도록 구현하였다.

NXT 로봇은 <표 1>과 같이 32비트 ARM7 마이크로프로세서를 장착한 소형 컴퓨터이며 주변 환경을 인식하고 인터페이스 할 수 있도록 4개의 입력 센서 포트를 가지고 있다. 그리고 3개의 출력 서보모터 포트를 갖고 있으며, 각각 독립적으로 서보모터의 구동이 가능하다. 통신기능으로 USB 포트와 블루투스 통신 모듈을 내장하고 있어 PC나 브릭 간의 유무선 통신이 가능하다.

<표 1> 마인드스톰 NXT 로봇의 명세

NXT 브릭	32Bit ARM7 MCU(50MHz), 4개의 입력 포트, 3개의 출력 포트, Atmel 8Bit AVR MCU(8MHz),
센서	터치센서, 빛 센서, 회전 센서(모터내장), 소리 센서, 초음파 센서.
모터	서보모터 세트 3개
메모리	64Kbytes SRAM, 256Kbytes Flash ROM

#### 3.2.2 NBC

NBC(Next Byte Codes)는 어셈블러와 유사한 NXT 로봇 제어용 저급언어이다.[8] NBC 언어의 컴파일러는 NXT 로봇이 사용하는 바이트 코드로 변환하여 NXT 로봇을 실행한다. NBC 언어는 NXT 로봇의 하드웨어를 제어할 수 있는 API(Application Programming Interface)를 제공한다.

## 4. 교육용 컴파일러 통합개발환경

### 4.1 시스템 구조



<그림 2> 교육용 컴파일러 통합개발환경의 구조

교육용 컴파일러 통합개발환경은 크게 컴파일러 제작 도구, 레퍼런스 컴파일러, 코드생성 시각화 도구, 목적언어 테스트 도구로 나뉜다. 4개의 모듈은 이클립스 플러그인 프레임워크를 기반으로 구현되어 이클립스 플랫폼 런타임에서 수행된다. <그림 2>는 전체적인 구조를 간략하게 제시한 것이다. 이클립스 플랫폼 런타임은 자바 가상 기계 상에서 실행된다. 교육용컴파일러 통합개발환경은 자바 가상 기계만 있다면 운영체제나 하드웨어에 의존하지 않고 독립적으로 실행된다.

#### 4.1.1 레퍼런스 컴파일러

교육용 컴파일러 통합개발환경은 컴파일러 수업에 교육적으로 활용할 수 있어야 한다. 따라서 컴파일러 수업에 제작할 컴파일러의 레퍼런스 컴파일러를 내장하였다. 교육 목적에 활용할 원시언어(NXT C)와 목적언어(Mini NBC)를 정의 하였으며, 레퍼런스 컴파일러는 원시언어를 컴파일하여 목적언어 코드를 생성한다.

#### 4.1.2 컴파일러 제작 도구

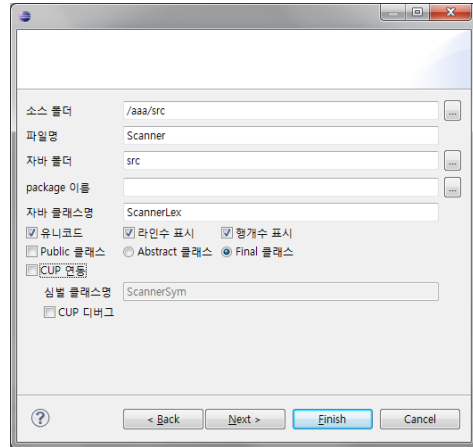
컴파일러 제작 도구는 JFLEX 편집기, CUP 편집기, 구문트리 시각화 도구로 구성한다. 이 도구는 어휘분석기, 구문분석기를 개발하는데 필요한 JFLEX와 CUP 프로그램 등을 제공하며, 기존 CUI(Command User Interface) 기반의 JFLEX / CUP 프로그램을 이클립스 플랫폼에서 수행하는 GUI(Graphic User Interface) 기반으로 확장하여 학생들의 컴파일러 개발 편의성을 높였다.

##### 1) JFLEX 편집기

JFLEX는 Java 환경에서 사용되는 어휘분석기 생성 프로그램이다. LEX와 유사한 문법을 사용하여 어휘분석기에 대한 명세를 작성하며, JFLEX 명세에 따른 어휘분석기는 Java 소스 프로그램 형태로 출력된다. 교육용 컴파일러 통합개발환경에서 제공하는 JFLEX 편집기는 JFLEX 명세 작성과 어휘분석기 생성을 CUI가 아닌 이클립스 기반 GUI 환경에서 수행한다.

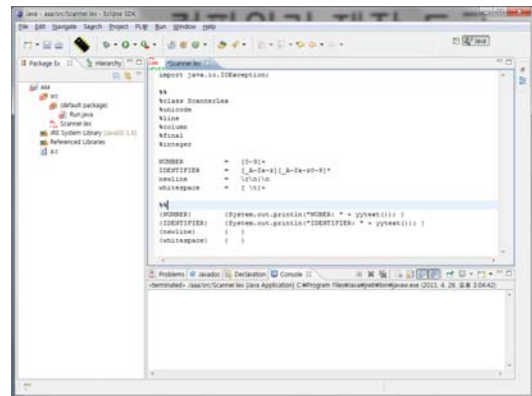
교육용 컴파일러 통합개발환경에서 어휘분석기

를 제작하기 위해서 <그림 3>의 대화상자를 통하여 JFLEX 명세를 생성한다.



<그림 3> JFLEX 명세 생성 대화상자

<그림 3>의 대화상자는 JFLEX 명세가 저장되는 경로, JFLEX 명세에 따른 Java 소스 프로그램 형태의 어휘분석기가 저장되는 경로와 클래스명을 지정한다. 대화상자 중단의 체크리스트를 통하여 JFLEX 명세의 세부설정까지 지정하고 'Finish' 버튼을 누르면 지정된 설정이 적용된 JFLEX 명세가 생성된다.



<그림 4> JFLEX 편집기

<그림 4>의 우측 상단에 위치한 편집기는 JFLEX 편집기이다. 이 편집기를 통하여 JFLEX 명세를 작성하고 어휘분석기를 생성한다. <그림 4>의 JFLEX 편집기에 작성된 코드는 <그림 3>의 대화상자의 설정에 따라 생성된 JFLEX 명세이며, 학생은 생성된 JFLEX 명세를 바탕으로 JFLEX 편집기에서 교육용 컴파일러 어휘분석기의 JFLEX 명세를 작성한다. JFLEX 편집기에서 어휘분석기의 JFLEX 명세 작성이 끝나면 통합개발환경의 컴파일 명령이나 저장 명령을 실행하여

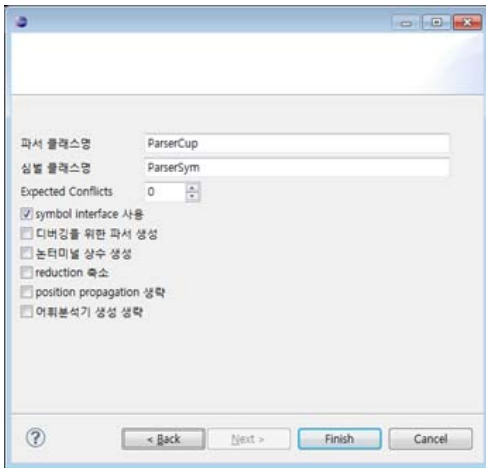
Java 소스 프로그램 형태의 어휘분석기를 생성한다. 생성된 어휘분석기는 교육용 컴파일러 통합개발환경에 관리되며, 통합개발환경에 내장된 Java 컴파일러와 디버거의 지원을 제공받는다.

## 2) CUP 편집기

CUP은 Java 환경의 Bottom-up parser 생성 프로그램이다. YACC과 유사한 문법을 사용하여 구문분석기를 작성하며, 작성된 구문분석기는 Java 소스 프로그램 형태로 출력된다. 교육용 컴파일러 통합개발환경에서 제공하는 CUP 편집기는 CUP 명세 작성과 구문분석기 생성을 CUI가 아닌 이클립스 기반 GUI 환경에서 수행한다.

교육용 컴파일러 통합개발환경에서 구문분석기를 제작하기 위해서 <그림 5>의 대화상자를 통하여 CUP 명세를 생성한다.

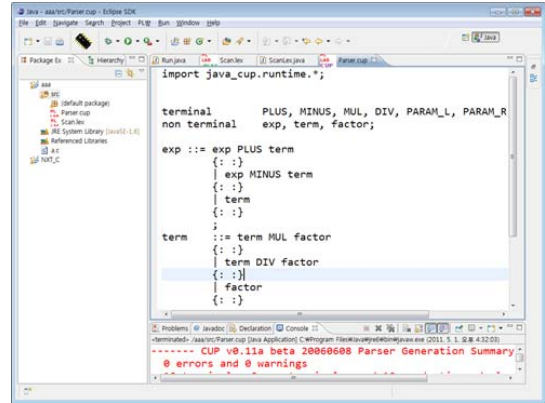
<그림 5>의 대화상자는 CUP 명세에 따른 Java 소스 프로그램 형태의 구문분석기가 저장되는 경로와 클래스명을 지정한다. 대화상자 중단의 체크리스트를 통하여 CUP 명세의 세부설정까지 지정하고 'Finish' 버튼을 누르면 지정된 설정이 적용된 CUP 명세가 생성된다.



<그림 5> CU 명세 생성 대화상자

<그림 6>의 우측에 위치한 편집기는 CUP 편집기이다. 이 편집기를 통하여 CUP 명세를 작성하고 구문분석기를 생성한다. <그림 6>의 CUP 편집기에 작성된 코드는 <그림 5>의 대화상자의 설정에 따라 생성된 CUP 명세이며, 학생은 생성된 CUP 명세를 바탕으로 CUP 편집기에서 교육용 컴파일러 구문분석기의 CUP 명세를 작성한다.

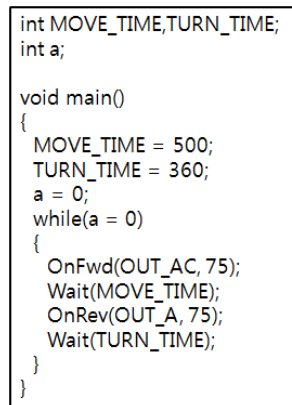
CUP 편집기에서 구문분석기의 CUP 명세 작성이 끝나면 통합개발환경의 컴파일 명령이나 저장 명령을 실행하여 Java 소스 프로그램 형태의 구문분석기를 생성한다. 생성된 구문분석기는 교육용 컴파일러 통합개발환경에 관리되며, 통합개발환경에 내장된 Java 컴파일러와 디버거의 지원을 제공받는다.



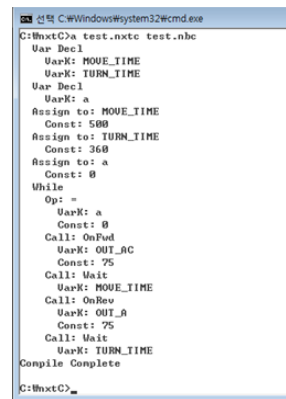
<그림 6> CUP 편집기

## 3) 구문트리 시각화 도구

구문트리 시각화 도구는 컴파일 과정 중에 생성된 구문트리를 GUI 기반으로 시각화하는 도구이다. 학생들이 정의한 트리 노드 자료형으로 생성된 구문트리를 시각화 가능하도록 설계하기 위해서 Java에서 제공하는 Iterator 인터페이스를 적용한다. 이러한 Iterator 인터페이스 구현은 구문트리 시각화 도구가 특정 트리 노드의 자료형에 대해 의존하는 문제를 해결한다.



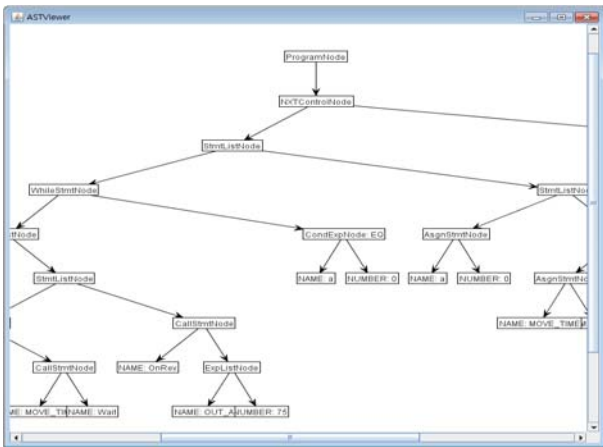
<그림 7> NXT C 프로그램



<그림 8> 텍스트 기반 구문트리 시각화 도구

<그림 8>은 <그림 7>의 NXT C 프로그램의 구문트리를 텍스트 기반으로 구문트리 시각화하는 기존 도구이다. 텍스트 기반 구문트리 시각화 도구는 구문트리를 들여쓰기로 작성된 텍스트로 재구성하여 구문트리 노드정보와 자식 노드정보를 출력한다.

<그림 9>는 GUI 기반으로 확장하여 교육용 컴파일러 통합개발환경에서 제공하는 구문트리 시각화 도구이다. 이 도구는 실제 구문트리를 순회하여 구문트리 노드정보를 글상자 형식으로 표현하며, 자식노드의 위치는 실선과 화살표 형식으로 표현한다. <그림 8>과 <그림 9>와 같은 도구는 구문분석이 올바르게 수행되어 구문트리가 이상 없이 생성되었는지 학생이 직접 확인하는데 도움을 주는 도구이며, 교육용 컴파일러 통합개발환경에 내장된 GUI 기반 구문트리 시각화 도구는 텍스트 기반 도구보다 시인성이 높기 때문에 학생이 더욱 적극적으로 구분분석 결과와 구문트리를 확인하도록 돕는 역할을 한다.



<그림 9> 통합개발환경에서 제공하는 GUI 기반 구문트리 시각화 도구

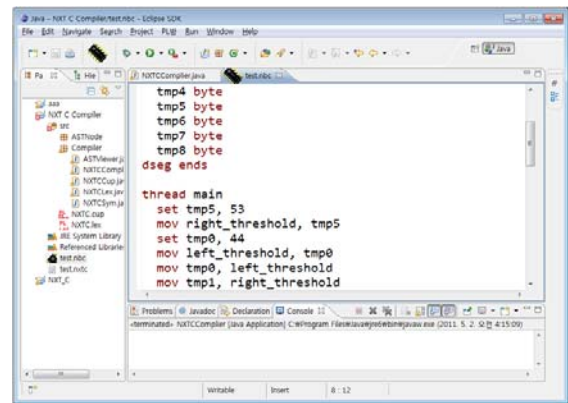
4.1.3 목적언어 테스트 도구

목적언어 테스트 도구는 Mini NBC 코드 뷰어, NXT 제어기, NXT 에뮬레이터로 구성된다. 이 도구는 목적언어(Mini NBC)를 실행하는 환경과 실행 결과를 확인하는 이클립스 통합개발환경 기반 프로그램이다. 이러한 테스트 도구는 목적코드의 생성과 실행을 통해 코드생성 과정을 검증하는 용도로 사용된다.

1) Mini NBC 코드 뷰어

Mini NBC 코드 뷰어는 컴파일 과정에서 생성된 목적언어(Mini NBC) 코드를 보여주는 도구이다. Mini NBC 문법을 따른 구문강조 기능 제공하여 생성된 목적언어 구조를 학생들이 쉽게 이해하도록 지원한다.

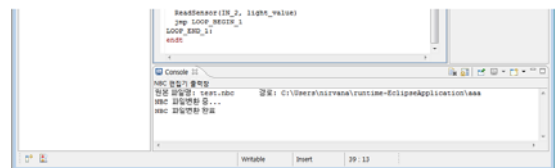
<그림 10>은 Mini NBC 코드뷰어의 실행화면을 나타낸다. <그림 10>에서 보는 바와 같이 구문강조 기능이 적용되어 Mini NBC 명령어와 식별자를 쉽게 구분할 수 있어 학생이 Mini NBC 코드를 보는데 도움을 준다.



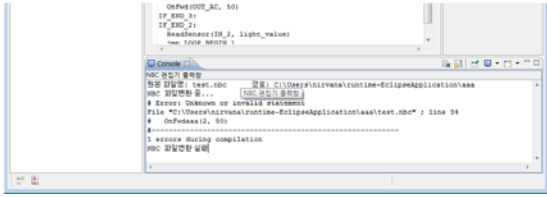
<그림 10> Mini NBC 코드 뷰어

2) NXT 제어기

NXT 제어기는 목적언어를 NXT 로봇이 사용하는 바이트 코드로 변환하며 변환된 바이트코드를 NXT 로봇에 업로드하고 로봇을 실행하는 도구이다. NXT 제어를 통한 NXT 바이트 코드 변환과 NXT 로봇 실행은 이클립스 통합개발환경 기반에서 수행된다. <그림 11>은 NXT 제어기에서 NXT 바이트 코드 변환이 정상적으로 수행된 경우이며, <그림 12>는 변환이 실패한 경우로 오류 종류와 오류 위치를 사용자에게 알려준다.



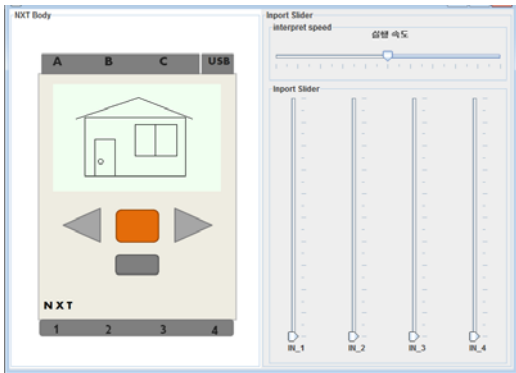
<그림 11> NXT 바이트 코드 변환 성공



<그림 12> NXT 바이트 코드 변환 실패

3) NXT 에뮬레이터

NXT 에뮬레이터는 가상의 NXT 로봇을 통하여 Mini NBC 코드를 실행하는 도구이다. 비용이나 실습 장소 때문에 NXT 로봇을 사용하기 어려운 문제를 NXT 에뮬레이터를 사용하여 해결할 수 있다. NXT 에뮬레이터는 가상 NXT 로봇의 센서 값을 제어할 수 있으며, LCD 출력창과 실행 속도를 제어할 수 있어 다양한 목적언어 실행환경을 제공한다.

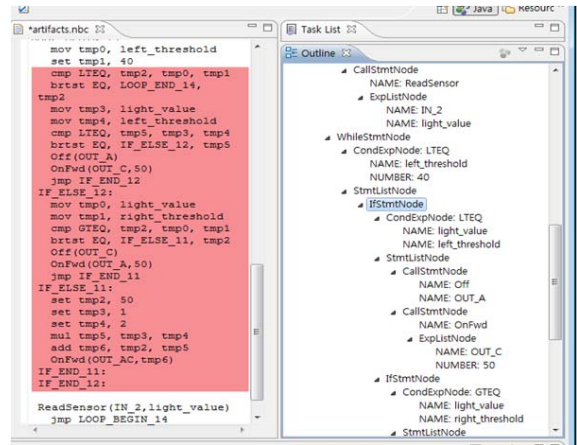


<그림 13> NXT 에뮬레이터

<그림 13>은 NXT 에뮬레이터의 실행화면이다. 왼쪽 화면은 가상의 NXT 로봇 브릭으로 LCD 출력과 모터, 센서의 사용 여부가 표시된다. 오른쪽 화면의 조절 막대로 실행속도와 센서 값을 조절하여 가상 NXT 로봇의 센서 입력 값을 설정한다. Mini NBC 코드를 입력받은 NXT 에뮬레이터는 설정된 센서 값과 Mini NBC 프로그램에 따라 동작되며, 이러한 Mini NBC 프로그램의 실행과정과 실행결과는 가상의 NXT 로봇 브릭을 통하여 사용자에게 전달한다.

4.1.4 코드생성 시각화 도구

컴파일러 수업을 수강하는 학생은 저급언어에 익숙하지 않기 때문에 코드생성 과정은 특히 어렵다. 교육용 컴파일러 통합개발환경에서 코드 생성 과정의 이해를 돕기 위하여 코드 생성 시각화 도구를 제공한다. 코드 생성 시각화 도구는 구문트리 노드에서 생성되는 목적코드를 GUI 기반으로 표시하여 구문트리와 목적언어의 관계를 시각적으로 표현한다. 이러한 표현은 학생이 코드생성 알고리즘을 쉽게 이해하는데 도움을 준다.



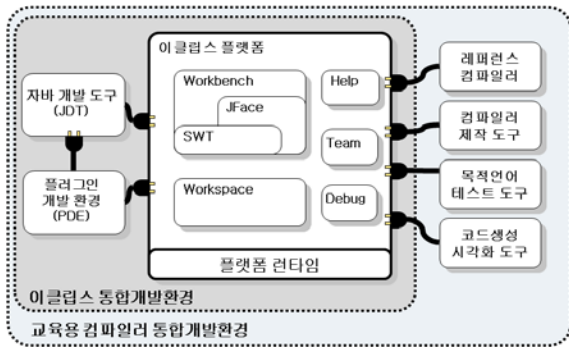
<그림 14> 코드 생성 시각화 도구

<그림 14>는 코드생성 시각화 도구의 실행화면으로 분할된 오른쪽 창은 NXT C 프로그램의 구문트리가 표시되며, 왼쪽 창은 Mini NBC 코드를 표시한다. 현재 <그림 14>의 코드생성 시각화 도구는 구문트리 노드 IfStmntNode가 생성하는 Mini NBC 코드가 강조된 상태이며, 오른쪽 구문트리 화면에서 노드를 선택하면 선택된 노드와 노드의 자손들이 생성한 코드만 왼쪽 Mini NBC 코드 표시 화면에 강조된다. 이러한 구문트리 노드와 노드가 생성한 목적언어 코드의 관계를 표현하기 위하여 구문트리 노드가 생성한 목적언어 코드의 행과 열의 위치를 기억하는 노드 자료구조를 정의한다. 행과 열의 위치가 저장된 구문트리의 노드를 선택하면 선택된 노드부터 순회를 시작하며, 순회된 노드가 가지고 있는 행과 열의 위치를 목적언어 코드 표시 화면에 강조 하여 구문트리 노드와 목적언어 코드의 관계를 표현한다.



### 4.2 이클립스와 통합 및 배포

교육용 컴파일러 통합개발환경은 이클립스 플러그인 프레임워크를 적용하여 개발한다. 교육용 컴파일러 통합개발환경은 Java 모듈 형식인 Jar 파일로 배포하며, 배포된 Jar 파일을 이클립스에 추가하면 교육용 컴파일러 통합개발환경은 이클립스와 통합되어서 하나의 프로그램으로 구성된다. 교육용 컴파일러 통합개발환경의 모듈은 <그림 15>와 같이 이클립스 플러그인 형태로 결합하여 이클립스 통합개발환경과 연동된다.



<그림 15> 이클립스 통합개발환경과 연동되는 교육용 컴파일러 통합개발환경

<표 2> 교육용 컴파일러 통합개발환경 적용 효과

주차	강의 주제	적용 모듈	효과
1	교육용 컴파일러 통합개발환경 설치		이클립스 통합개발환경과 플러그인 개념을 학습
2-3	어휘분석기 제작	JFLEX 편집기	편집기에서 생성된 어휘분석기 템플릿을 사용하여 효율적으로 JFLEX 문서작성
4-5	구문분석기 제작	CUP 편집기	편집기에서 생성된 구문분석기 템플릿을 사용하여 효율적으로 CUP 문서작성
6	구문트리 자료구조 정의	구문트리 시각화 도구	구문분석 결과와 트리 노드의 자료구조를 시각으로 직접 확인
7-9	코드 생성기 제작	코드생성 시각화 도구	코드생성 알고리즘과 구문트리-목적언어 관계를 효과적으로 이해
10	목적코드 실행 및 컴파일러 테스트	Mini NBC 코드 뷰어, NXT 제어기, NXT 에뮬레이터	학생이 개발한 컴파일러의 결과를 확인하고 실행하여 컴파일러의 신뢰도를 평가

### 5. 컴파일러 개발 교육에의 적용

컴파일러 과목 실습 강의에서 10주간 컴파일러 개발 프로젝트 일정을 설계하였고 학생에게 교육용 컴파일러 통합개발환경을 사용하도록 하였다. 학생이 컴파일러를 개발하는 동안 <그림 16>과 같이 교육용 컴파일러 통합개발환경은 컴파일러 개발을 지원하였다. 이러한 개발 지원을 제공받은 학생은 컴파일러 개발과 테스트를 10주간 수행하였고 <표 2>와 같은 효과를 얻었다.



<그림 16> 컴파일러 개발 교육 과정에 제공되는 통합개발환경 기능

### 6. 결론 및 향후 과제

본 논문에서는 마인드스톰 NXT 로봇을 대상으로 하는 C언어 기반의 간단한 원시언어와 이해하기 쉬운 목적언어를 사용하는 교육용 컴파일러를 설계하고, 교육용 컴파일러 개발 프로젝트를 지원하는 통합개발환경을 제시하였다. 본 통합개발환경을 활용하여 컴파일러 제작을 효율적으로 개발할 수 있으며, 목적언어 실행과 테스트도 간단하게 확인할 수 있다.

교육용 컴파일러 통합개발환경은 컴파일러 개발 실습을 수행할 수 있는 환경을 제공하였다. 이는 컴파일러 개발에 더욱 집중할 수 있게 하여 학생이 컴파일러를 제작하는 개발시간을 단축할 수 있으며, 목적언어를 NXT 로봇으로 실행하기 때문에 컴파일러 개발에 대한 흥미를 높인다.

제안한 교육용 컴파일러 통합개발환경은 이클립스 플러그인 프레임워크를 기반으로 하였기 때문에 추가로 요구되는 플러그인을 통한 확장이

가능하고 배포가 용이하며 업데이트 기능을 이용하여 지속적인 유지, 보수가 가능하다.

이러한 특징을 활용하여 추후에는 JFLEX/CUP 이외의 컴파일러 제작도구를 지원하는 유연한 구조로 확장하면 더욱 범용적인 컴파일러 통합개발 환경이 될 것이다. 현재의 교육용 컴파일러 통합 개발환경을 웹 기반으로 확장을 한다면 배포가 필요 없는 플랫폼 독립적인 개발환경으로 개선할 수 있어 학생이 장소와 실행환경에 제약 받지 않고 통합개발환경을 사용할 수 있을 것이다. 추가로 구문분석 과정을 설명하는 도구와 지능적인 에러처리 도구를 개발한다면 컴파일러에 대한 이해를 높일 수 있는 요소가 될 것이다.

### 참고 문헌

- [1] Computing curricula 2008: An interim revision of cs 2001.  
<http://www.acm.org/education/curricula/ComputerScience2008.pdf>
- [2] Li Xu, Fred G. Martin(2006). *Chirp on Crickets: Teaching Compilers Using an Embedded Robot Controller*. SIGCSE '06.
- [3] Li Xu(2007). *RobotStudio: A Modern IDE-based Approach to Reality Computing*. SIGCSE '07.
- [4] Marc L. Corliss, E Christopher Lewis(2008). *Bantam: A Customizable, Java-Based, Classroom Compiler*. SIGCSE '08.
- [5] Tyler Sondag, Kian L. Pokorny, Hridesh Rajan(2010). *Frances: A Tool For Understanding Code Generation*. SIGCSE'10.
- [6] Eclipse Foundation, Inc. About the Eclipse Foundation. <http://www.eclipse.org/org/>
- [7] Lego, Inc. 8547 LEGO® MINDSTORMS® NXT 2.0. <http://mindstorms.lego.com>
- [8] John Hansen. Next Byte Codes(NBC) Programmer's Guide Version 1.0.1 b33
- [9] Gerwin Klein. JFlex - The Fast Scanner Generator for Java. <http://jflex.de/>
- [10] Scott E. Hudson. CUP Parser Generator for Java.

<http://www.cs.princeton.edu/~appel/modern/java/CUP/>

### 성우경



2007 경상대학교  
컴퓨터과학과(학사)  
2009 경상대학교  
컴퓨터과학과(석사)

관심분야: 프로그래밍언어, 정보검색, 병렬처리  
E-Mail: sung\_u\_kyung@hotmail.com

### 강현석



1981 동국대학교  
전자계산학과(학사)  
1983 서울대학교  
계산통계학과(석사)

1989 서울대학교 계산통계학과(박사)  
1984~1993 전북대학교 전자계산학과 부교수  
1984~현재 경상대학교 컴퓨터과학과 교수  
관심분야: 객체지향 데이터베이스, 전자문서 관리, 멀티미디어  
E-Mail: hskang@gnu.ac.kr

### 배종민



1980 서울대학교  
수학교육과(학사)  
1983 서울대학교  
계산통계학과(석사)

1995 서울대학교 계산통계학과(박사)  
1982~1984 한국전자통신연구소 연구원  
1997~1998 Virginia Tech. 방문연구원  
1984~현재 경상대학교 조교수, 부교수, 교수  
관심분야: 컴파일러, 프로그래밍언어, 컴퓨터교육, XML, 데이터베이스 통합,  
E-Mail: jmbae@gnu.ac.kr