

스몰베이직 프로그램 해석기 설계 및 구현에 관한 연구*

김가영 정승완 최광훈
전남대학교 전자컴퓨터공학부
qrkdudq4301@naver.com sjx10@naver.com kwanghoon.choi@jnu.ac.kr

김태진 조영민 김범준
연세대학교 컴퓨터정보 통신공학부
xowls157@naver.com mamiruleess@gmail.com bjkim320@lycos.co.kr

A Study on the Design and Implementation of a Small Basic Program Interpreter

Gayoung Kim Seungwan Jeong Kwanghoon Choi
Dept. Electronics and Computer Engineering, Chonnam National University, Gwangju

Teajin Kim Youngmin Jo Beomjun Kim
Computer & Telecommunications Engineering Division, Yonsei University, Wonju

요 약

본 논문은 다양한 운영체제와 플랫폼에서 스몰베이직 프로그래밍을 할 수 있는 환경을 설계하고 구현하는 목표로 진행 중인 연구 내용을 요약한다. 마이크로소프트 스몰베이직은 처음 컴퓨터 프로그래밍을 배울 때 고려할 수 있는 프로그래밍 언어이다. 단 14개의 키워드로 구성되어 있어 매우 간단하고 배우기 쉽다. 하지만 프로그래밍 환경을 윈도우 운영체제에서만 사용해야 하는 단점이 있다. 소스 코드가 공개되어 있지 않아 새로운 요구사항을 맞추기도 어렵다. 예를 들어, 초보자의 흥미를 높이는데 필요한 새로운 라이브러리를 쉽게 추가할 수 없다. 이러한 문제를 해결하기 위해 다양한 운영체제에서 사용 가능하고 쉽게 라이브러리를 추가할 수 있는 스몰베이직 환경을 설계하고 구현하고자 한다. 윈도우, 맥, 리눅스 운영체제에서 스몰베이직 프로그램을 작성하고 PC 뿐만 아니라 안드로이드 기반 스마트폰에서 실행할 수 있다.

1. 서 론

모바일 시대를 넘어서 다가오는 사물인터넷 시대에는 모든 사물에 컴퓨터 기능이 탑재되고 이를 운용하기 위한 소프트웨어가 필요하다. 그만큼 사회 전반에 소프트웨어의 역할이 커지게 되어 소프트웨어에 대한 이해가 매우 중요하다. 대학에서 컴퓨터 이외의 전공자에게도 코딩 교육을 시키고 있고, 초중고교에서 일정시간의 코딩 교육이 의무화되고 있는 것이 세계적인 추세이다.

마이크로소프트의 비제에 라지(Vijaye Raji)가 개발한 스몰베이직(Small Basic)[1,2]은 처음 코딩을 접하는 사람을 위한 윈도우 기반 프로그래밍언어 및 코딩 환경이다. 코딩을 전혀 배우지 않은 사람이 처음 배우기에 적합하도

록 이해하기 쉬운 최소한의 언어 요소만 도입하여 만들었다. 코딩 환경이 매우 단순하여 적응하기 쉽고, 라이브러리를 활용하여 텍스트 및 그래픽스 프로그램을 쉽게 작성할 수 있고, C#이나 VB.Net으로 새로운 라이브러리를 만들어 추가할 수 있다. 또한 소스 프로그램을 공유할 수 있는 일종의 앱스토어를 제공한다.

스크래치나 앱인벤터와 같은 교육용 코딩 환경을 통해서 코딩의 기초 개념을 배울 수 있지만 이러한 환경에서는 코드가 없는(그림으로 작성된) 프로그램을 통해서 코딩을 배운다. 스몰베이직은 이러한 점에서 다르게 접근한다. 파이썬이나 C/C++/Java는 초보자가 처음 코딩 개념을 배우기에 너무 많은 특징을 보유하고 있어 장벽이 될 수 있다. 이러한 점에서 스몰베이직은 처음 코딩을 배우기에 적합한 프로그래밍언어다.

그럼에도 불구하고, 스몰베이직은 마이크로소프트의 닷넷프레임워크에 종속되어 리눅스, 맥 운영체제, 웹, 안드로이드

* 이 논문은 2016년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. NRF-2014R1A1A2053446)

로이드 스마트폰에서 실행하기에 불편하고, 스몰베이직의 확장 역시 닷넷프레임워크와 연관된 도구를 통해 개발해야 한다.

본 연구에서는 스몰베이직을 더욱 다양한 운영체제와 플랫폼에서 활용할 수 있도록 오픈소스 기반으로 코딩 환경을 개발하여 국내 코딩 교육의 커뮤니티를 양성하는 것을 궁극적인 목표로 삼고 있다. 이 논문에서 이러한 연구의 초기 진행 사항을 요약하는 것이다.

논문의 구성은 다음과 같다. 2절에서 스몰베이직에 대해 요약해서 소개한 후 초보자를 위한 다른 코딩 언어와 비교하고, 3절에서 스몰베이직 코딩 환경 설계 및 개발의 진행 사항을 요약한다. 4절에서 연구 진행에서 어려운 점을 논의하고, 5절에서 결론을 맺고 향후 연구를 논의한다.

2. 관련 연구

스몰베이직을 이용해 플리커 웹사이트에서 지정한 주제의 사진들을 찾아 컴퓨터 바탕 화면에 슬라이드 쇼하는 프로그램[2]을 [그림 1]과 같이 간단하게 작성할 수 있다.

```
For I=1 To 20
  pic = Flickr.GetRandomPicture("Korea")
  Desktop.SetWallPaper(pic)
EndFor
```

Fig 1 A Small Basic Program for Wall Paper Slide Show

스몰베이직의 특징을 다음과 같이 요약할 수 있다[3].

- 스몰베이직의 키워드는 전체 14개, IF, Then, Else, EndIf, While, EndWhile, Elseif, For, To, Step, EndFor, Goto, Sub, EndSub이다.
- 특별한 선언 없이 변수를 사용하고, 스코프(scope) 개념이 없이 모든 변수를 전역 변수로 사용한다.
- 타입 개념 역시 없다. 문자열 상수와 숫자 상수를 프로그램에서 사용할 수 있으나 내부적으로 모두 문자열로 관리하고 연산을 수행할 때 적절히 해석한다. 배열을 지원한다.
- 직관적으로 이해하기 쉬운 Goto문과 서브루틴 호출을 통한 제어 흐름 방식을 지원한다. IF, For, While을 지원한다.
- Flickr나 Desktop과 같이 간단한 형태의 객체 개념을 통해 라이브러리 함수와 설정 변수를 제공한다.

스몰베이직과 코딩 교육에 사용되는 다른 언어나 환경을 다음 표와 같이 간략하게 비교할 수 있다.

	Simplicity	Accessibility	Library
Scratch	Simple	Web/PC	Limited
AppInventor	(Codeless)		
Small Basic	Simple	Windows	Limited
Python	Complex	OS-neutral	Rich
C/C++/Java			

배우기 쉬운 장점이 있음에도 불구하고 스몰베이직 환경이 윈도우로 국한되는 단점이 있고, 처음 코딩을 배우는 사람의 흥미를 끌기 위해 모바일이나 사물인터넷 기기와 연동하기 위한 라이브러리가 제한적인 단점도 있다. 이 연구에서 스몰베이직 코딩 환경을 Java기반으로 다시 개발하여 특정 운영체제의 의존성을 없애고 (Java로 작성된) 모바일과 사물인터넷 기기와 보다 쉽게 연동할 수 있는 환경을 만들고자 한다.

3. 스몰베이직 프로그램 해석기

2절에서 설명한 동기로 Java 기반 스몰베이직 언어의 해석기를 작성하였다. [그림 2]와 같이 파싱 단계, 기본 블록으로 변환하는 단계, 실행하는 단계로 나누어 구현하였다.

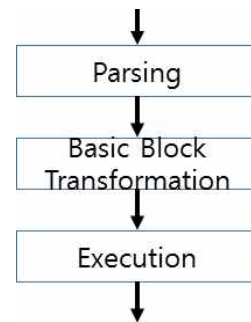


Fig 2 The Internal Stages of the Small Basic Interpreter

파싱 단계에서는 스몰베이직 예제 프로그램들을 기반으로 LALR(1) 문법을 역으로 유추하여 작성하여 구문 분석기를 구현하였다. 마이크로소프트에서 제공하는 스몰베이직 교재의 예제 프로그램을 모두 제대로 파싱함을 확인하였다.

기본 블록 변환 단계에서는 입력으로 주어진 스몰베이직 프로그램의 추상구문트리(AST, Abstract Syntax Tree)를 따라가며 기본 블록으로 나눈다.

- 기본 블록은 일련의 문장들로 중간 문장에 라벨이 있어 기본 블록의 시작을 거치지 않고 점프해서 들어올 수 없고 중간 문장에 Goto 문이 있어 끝까지 실행하지 않고 점프해서 나가는 경우가 없도록 만든 것이다.
- 보통 기본 블록의 마지막 문장은 Goto 문으로 끝난다. 만일 Goto 문이 없으면 더 이상 실행할 문장이 없는 것으로 판단한다.
- 기본 블록은 특별한 확장 없이 스몰베이직 프로그램의 추상구문트리로 표현할 수 있다.

[그림 3]에서는 [그림 1]의 예제 프로그램을 입력 받아 기본 블록 변환 단계 이후에 얻은 기본 블록들을 보여준다. 각 기본 블록에는 라벨을 붙여 구분한다. 그리고 스몰베이직 프로그램을 시작하면 \$main 라벨의 기본 블록부터 실행한다.

```
$main:
  Goto $L0
$L0:
  I = 1
  Goto $L1
$L1:
  If I <= 20 Then
    pic = Flickr.GetRandomPicture("Korea")
    Desktop.SetWallPaper(pic)
    I = I + 1
    Goto $L1
  EndIf
```

Fig 3 Basic Blocks for the Example in Fig 1

실행 단계에서는 도메인 Env의 변수 환경을 준비하고 각 문장을 실행한다.

- Env ::= VarName -> Value
- Value ::= StrValue | ArrayValue
- StrValue ::= String
- ArrayValue ::= Index -> Value
- Index ::= String

도메인 Value는 문자열 형태의 값(StrValue)와 배열(ArrayValue)이다. 배열은 문자열로 표현된 첨자를 배열 원소의 값으로 매핑하는 것으로 표현한다. 다차원 배열은 배열 원소를 배열로 두어 표현할 수 있다.

If와 Sub문은 Java의 조건문과 함수 호출로 구현하고 Goto문은 지정한 라벨의 블록을 실행한다. 더 이상 실행할 문장이 없으면 종료한다.

식에 포함된 객체의 메소드 호출이나 필드 읽기 쓰기의 경우 해당 객체에 대한 라이브러리를 활용하는 방식으로 구현한다. 라이브러리 호출은 Java의 리플렉션(Reflection)으로 구현하여, 추가로 라이브러리를 쉽게 확장할 수 있도록 구성하였다.

4. 논의 사항

스몰베이직 언어에 대한 구문과 의미를 명확히 설명하는 문서가 없어 구문 분석기와 해석기 구현이 잘 동작하는지 검증하는데 어려움이 있다. 현재 마이크로소프트 스몰베이직 환경은 바이너리 실행 파일로 제공하므로 샘플 프로그램을 작성하여 실행해봄으로써 역으로 유추하는 방식에 의존할 수밖에 없다.

스몰베이직 라이브러리를 구현할 때도 동일한 문제가 있다. 이 문제 역시 많은 스몰베이직 샘플 프로그램을 모아 마이크로소프트 스몰베이직 환경의 실행 결과와 비교함으로써 최대한 동일하게 동작할 수 있도록 구현할 계획이다.

5. 결론 및 향후 연구

이 논문에서 Java로 스몰베이직 언어 해석기를 개발한 내용을 리포트 했다. 스몰베이직은 처음 코딩에 접하는 사람에게 매우 유용한 간단한 프로그래밍 언어의 장점을 유지하면서, 현재 마이크로소프트 스몰베이직 환경이 윈도우에서만 사용가능하고 라이브러리 확장이 자유롭지 못한 단점을 보완할 수 있을 것이다.

향후 스몰베이직 언어의 기본 라이브러리를 모두 구현하고 최소한의 코딩 환경을 갖춘 다음 전체 소스 코드를 오픈할 예정이다. 이 오픈소스를 기반으로 국내 코딩 교육의 커뮤니티를 양성하는데 기여할 수 있을 것이다.

참고 문헌

[1] Microsoft Small Basic, <http://smallbasic.com>.
 [2] Vijaye Raji, Erik Meijer, Expert to Expert: The Basics of SmallBasic, <https://channel9.msdn.com/blogs/charles/expert-to-expert-the-basics-of-smallbasic>, 2009.
 [3] Philip Conrod, Lou Tylee, The Developer's Reference Guide To Microsoft Small Basic, Kidware Software LLC, 2010.