

스크립트 언어 비교: 프로그래밍 편의성 관점에서

장한일⁰, 이동주^{*}, 지정훈^{*}, 우균^{*}, 김원영^{**}, 최완^{**}

^{*}부산대학교 컴퓨터공학과

^{**}한국전자통신연구원 디지털융연구단

{daystar⁰, mrlee, jhji, woogyun}@pusan.ac.kr, {wykim, wchoi}@etri.re.kr

Comparing Scripting Languages: From the Viewpoint of Programming Facilities

Hanil Jang⁰, Dongju Lee^{*}, JungHoon Ji^{*}, Gyun Woo^{*}, Won-Young Kim^{**}, Wan Choi^{**}

^{*}Dept of Computer Engineering, Pusan National University

^{**}Digital Home Research Division, Electronics and Telecommunications Research Institute

요약

스크립트 언어는 모듈의 결합이나 조정 등의 여러 목적으로 쓰이고 있는 언어이다. 스크립트 언어는 사용의 편리함과 높은 생산성으로 인해 널리 사용되고 있다. 그러나 현재 너무나 많은 스크립트 언어가 개발·제시되고 있어 프로그래머가 스크립트 언어를 선택해야 할 때 어떤 언어를 선택해야 할지 판단하기 힘든 실정이다. 이에 본 논문에서는 프로그래밍 편의성 관점에서의 비교 기준을 제시하고 대표적인 스크립트 언어(VB script, Ruby, Perl, Python, Tcl, Lua, Groovy)를 이 기준에 따라 비교하였다. 비교 결과 각 비교 분야에 강점을 보이는 언어를 발견하였다. 본 논문의 결과는 스크립트 언어를 선택하는데 도움이 될 것이다.

1. 배경

스크립트 언어는 일괄처리 작업을 효율적으로 처리하기 위해 제시된 언어로서, 예전엔 일괄처리 언어(batch languages) 또는 작업 관리 언어(job control languages)로 불렸다. 현재는 사용의 편리함과 높은 생산성 때문에 넓은 응용 분야에서 사용되고 있으며 지금도 많은 스크립트 언어가 개발되고 있다.

하지만 그 결과 너무 다양한 스크립트 언어가 제시되고 있다. 현재 명령 처리나 텍스트 처리 또는 범용의 목적으로 사용되고 있는 언어는 수십여 가지나 있다. 여기에 특정 응용 프로그램에 내장되는 형태의 스크립트 언어나 특정 목적으로 개발되는 스크립트 언어까지 포함하자면 그 수는 헤아릴 수가 없을 정도이다[1].

그 결과, 많은 스크립트 언어 중 다양한 응용 영역에 따라 어떤 스크립트 언어를 선택해야 할지 판단하기가 힘든 실정이다. 스크립트 언어 비교에 관한 관련 연구로는 Pascal Rigaux가 조사한 결과[2]가 있다. 그러나 Pascal Rigaux는 각 언어로 구현한 소스코드의 길이만을 비교했을 뿐, 각 언어가 제공하는 요소나 프로그래밍의 용이성에 대해서는 고려하지 않았다.

이 논문에서는 프로그래밍 편의성을 제고하기 위한 몇 가지 기준을 제시하고 이 기준에 따라 몇 가지 스크립트 언어를 비교한다. 이 논문의 목적은 스크립트 언어의 특징을 비교함으로써 스크립트 언어 선택에 도움을 주고자 함이다. 특정 스크립트 언어를 선전하거나 스크립트 언어 각각을 평가함으로써 경쟁을 유도하고자 하는 목적이 아니다.

2. 대상 스크립트 언어

스크립트 언어의 응용 분야에 따라 대표적인 스크립트 언어를 몇 가지 선정하였다. 여기에 선정된 언어는 현재

제시된 스크립트 언어 중에서 꾸준한 개발이 이루어지고 사용자층이 많은 언어들이다. 물론, 비교대상을 선정하는 과정에서 본 논문 작성자의 주관이 개입되었다는 것은 부인할 수 없으나 최대한 객관적으로 영향력 있는 스크립트 언어를 선정하기 위해 오픈소스 사이트[3]에서 많이 사용되고 있는 언어를 조사하여 선정하였다.

2.1. VB script

Windows 환경에서 Active X 컨트롤을 효과적으로 처리하기 위해 1991년 Microsoft에서 배포된 스크립트 언어이다. 원래 Visual Basic 언어는 범용 프로그래밍 언어로 제작되었으나 Microsoft 사의 적극적인 지원으로 스크립트 언어로서의 역할을 공고히 하고 있다.

2.2. Ruby

Yukihiro Matz Matsumoto가 1995년에 배포한 언어이다. 프로그래머의 고충을 덜어 줄 수 있는 요소를 제공하는 방향으로 설계된 언어이며 현재 Perl, Python과 함께 범용언어로 널리 사용되고 있다[4].

2.3. Perl

1980년대 후반 Larry Wall이 개발한 스크립트 언어이다. Perl은 Practical Extraction and Report Language를 의미한다. 이름이 나타내는 의미처럼 텍스트 생성과 처리를 목적으로 만들어진 언어이다. 하지만 현재에는 언어에 많은 요소가 보장되어 범용언어로 사용되고 있다. 현재 Perl은 Unix 서버의 CGI 작성에 널리 사용되고 있으며, 초기에는 Unix 플랫폼을 위해 제작되었지만 현재는 Windows 환경에서도 지원되고 있다[5].

2.4. Python

1990년에 Guido van Rossum이 개발한 언어이다. Python은 객체지향, 구조적 프로그래밍, 함수형 프로그래밍

밍 등의 많은 패러다임을 지원하려는 의도에서 설계된 언어이다. 구글(Google)같은 큰 프로젝트에도 사용되었던 언어이다. 역시 언어에 많은 요소가 포함되어 있어 범용 언어로 사용되고 있으며 Unix 환경에서 Perl과 경쟁하고 있다[6,7].

2.5. Tcl

1993년에 John Ousterhout가 개발한 언어이다. 셸 스크립트에서 발전된 형태의 언어이며, C와의 인터페이스를 제공하는 확장형/내장형(extension/embeddable) 언어이다. GUI 작성에 탁월한 성능을 보이며 파일 및 디렉토리, 문자열 처리 등을 쉽게 수행할 수 있도록 한다. 현재 다양한 패키지들이 개발되어 있다[8].

2.6. Lua

1993년에 Roberto Ierusalimschy, Luiz Henrique de Figueiredo, Waldemar Celes가 개발한 경량의 명령형 언어이다. C와의 인터페이스를 자연스럽게 지원한다. 컴파일된 바이트코드를 C로 작성된 인터프리터를 통해 수행할 수도 있고, 관련 C 라이브러리를 통해 C 코드에 내장된 형태로 독립적으로 실행될 수 있는 프로그램을 개발할 수 있다. 대부분의 요소가 메타테이블 같은 타입을 이용하는 등의 방법으로 동적으로 구현되어 있다[9,10].

2.7. Groovy

2003년에 James Strachan이 Python, Ruby, Smalltalk의 영향을 받아 Java 기반으로 개발한 언어이다. Java와 유사한 문법구조를 가지고 있으며 Java와의 인터페이스를 제공한다[11].

3. 비교 기준

여기서는 프로그래밍 편의성을 위해 필요한 몇 가지 기준을 제시한다. 이들 기준은 비록 저자의 주관에 따라 선정된 것이긴 하지만, 효과적인 프로그래밍을 위해 필수적인 기능들을 포함하고 있다. 3.10. 절에서 설명하겠지만, 필수적이라고 보기 힘든 고수준의 개념은 배제하였다. 특히 일반적인 스크립트 언어에서 지원하고 있지 않은 타입 관련 부분은 제외하였다.

3.1. 함수적 추상화

함수 및 프로시저를 지원하는가?

3.2. 객체기반 스크립팅

객체를 지원하는가? 별도의 선언문 없이 바로 객체를 생성하는 것이 가능한가?

3.3. 파일처리

파일 및 디렉토리 연산이 제공되는가? 미디어 파일(그림파일, 음악파일, 동영상파일 등) 처리를 지원하는가?

3.4. 외부 컴포넌트 접근성

개발된 라이브러리카나 컴포넌트의 사용이 가능한가? DB에 접근할 수 있는 방법을 제공하는가? 마크업 문서 처리를 지원하는 방법을 제공하는가?

3.5. GUI(Graphical User Interface)

윈도우, 버튼 등의 생성이 쉬운가? 이들 컨트롤에 대한

이벤트 처리가 쉽게 표현되는가?

3.6. 문자열 처리

문자열 처리 연산을 제공하는가? 정규식을 지원하는가?

3.7. 한글지원

한글 글자 단위로 문자열 처리가 가능한가? GUI 컨트롤 등에도 한글 처리가 잘 되는가? 한글 자소를 기반으로 정규식 작성이 가능한가?

3.8. 개발환경

통합 개발도구를 지원하는가? 유용한 라이브러리가 얼마나 많은가?

3.9. 다른 언어와의 연동

다른 언어에 내장되는 것이 가능한가? 다른 언어로 작성된 루틴이나 모듈을 사용하는 것이 가능한가? 언어에서 다른 언어와의 인터페이스를 지원하는 요소가 존재하는가?

3.10. 고려하지 않은 비교 기준

몇 가지 고급 언어의 기능은 위 기준에서 제외하였다. 제외된 기능들은 예외처리, 클래스, 상속, 타입 변환 등의 기능이다. 이들 기능은 고급 기능이긴 하지만 오히려 프로그래밍 복잡도를 높이는 측면이 있으므로 스크립트 언어만을 대상으로 한 이상의 비교 기준에서 제외하였다.

4. 스크립트 언어 비교

2절에서 제시한 각 언어에 대해 3절의 비교기준을 적용하여 각 조건에 대한 상대적 만족도를 표 1에 점수로 표현하였다. 표 1의 1열에는 비교 대상인 스크립트 언어를, 2열부터는 각 비교기준에 의한 평가점수를 기재하였다. 각 비교기준에 의한 점수는 각각 모두 10점 만점이며 기준을 만족시키는 정도에 따라 점수를 책정하였으며 상대적 만족도에 따라 점수를 차등 배분하였다.

표 1의 2열에 나타낸 기준 3.1 함수적 추상화에 관한 점수는 모두 10점으로 책정하였다. 비교대상 모두 함수 및 프로시저를 지원하기 때문이다. 점수에는 반영하지 않았지만 비교대상 중에서 Python, Lua, Groovy의 함수를 값(first-class value)으로 취급할 수 있으므로 다른 비교대상 보다 더 높은 추상화를 제공한다고 볼 수 있다. 특히, Lua의 클로저(closure)를 통해 경우 이러한 특징을 지원해준다.

기준 3.2와 관련하여 Lua를 제외한 다른 비교대상들은 객체를 바로 생성하는 것을 제공해주지 않았으며 나머지 비교 대상 중 Tcl은 객체를 지원하지 않았다.

기준 3.3에 대해서는 모두 8점을 책정하였다. 비교대상 모두 기본적인 디렉토리 및 파일 연산을 제공했으며 미디어 파일 처리는 모두 별도의 라이브러리를 이용해서 해결 가능하였기 때문이다.

기준 3.4 컴포넌트 부분에서는 Visual Basic에 가장 높은 점수를 책정하였다. 이는 Visual Basic에서 사용할 수 있는 컴포넌트들이 아주 방대하고 많으며 비교 조건을 모두 만족하기 때문이다. 나머지 비교대상들 모두 비교 조건을 모두 만족하나 Visual Basic에 비하면 상대적으로 사용할 수 있는 컴포넌트의 양이 적기 때문에 9점으로 책

스크립트 언어	비교기준									
	3.1	3.2	3.3	3.4	3.5	3.6	3.7	3.8	3.9	총계
VB Script	10	7	8	10	10	5	5	10	5	70
Ruby	10	7	8	9	8	10	5	7	7	72
Perl	10	7	8	9	8	10	5	8	7	72
Python	10	7	8	9	8	10	5	8	7	72
Tcl	10	0	8	9	9	10	5	8	7	66
Lua	10	10	8	9	8	7	5	8	10	75
Groovy	10	7	8	9	8	10	8	7	10	76

표 1. 비교기준에 따른 스크립트 언어 비교 결과

정하였다.

기준 3.5 GUI 관련 부분에서는 가장 다양한 컨트롤들을 제공했고 이벤트 처리 또한 간단했으며 Resource Editor를 이용하여 WYSIWYG 방식으로 GUI 컨트롤을 구성할 수 있는 Visual Basic이 가장 우수한 것으로 조사되었다. 그 다음 Tcl의 GUI 구성과 이벤트 처리가 간단히 표현되어 다른 비교대상들 보다 약간의 우위를 점하고 있다.

기준 3.6 문자열 처리 부분에서는 정규식을 지원하며 많은 스트링 처리 연산을 기본 제공하는 Ruby, Perl, Python, Tcl에 가장 많은 점수를 책정하였다. Groovy 역시 정규식을 지원하고 스트링 처리 연산을 제공하지만 Ruby, Perl, Python, Tcl에 비해서는 다양한 스트링 처리 연산을 제공하지 못하므로 차등을 두었다.

기준 3.7 한글지원 부분에서는 기본 문자열을 유니코드로 처리하는 Groovy에 가장 많은 점수를 책정하였다. 하지만 Groovy를 포함한 비교 대상들 모두 한글 자소를 기반으로 정규식을 작성하는 것이 불가능 하였다.

기준 3.8 개발환경 부분에서는 대부분 통합개발환경을 자체적인 툴이나 다른 툴의 플러그인 형태로 지원해주고 있었고, 이미 개발된 라이브러리가 다수 있었다. 하지만 Visual Basic은 가장 강력한 개발환경과 많은 컴포넌트를 제공해주고 있어 다른 언어와 차등을 두었고 Ruby와 Groovy의 경우에는 다른 비교 대상들에 비해 개발된 전용 라이브러리가 툴이 상대적으로 적었다.

기준 3.9와 관련하여, 다른 언어와의 연동이 가장 잘 되는 언어는 Lua와 Groovy로서 다른 언어에 내장이 용이하고 다른 언어의 루틴을 호출하는 것 또한 가능하다. 따라서 Lua와 Groovy에 가장 많은 점수를 책정하였다. 다른 비교대상들 또한 다른 언어와의 인터페이스를 지원하고 있으나 Lua, Groovy와는 차등을 두었다. Visual Basic의 경우에는 다른 언어로 작성된 컴포넌트를 사용할 수 있는 것에 그쳐 다른 비교대상과 차등을 두게 되었다.

각 비교기준에 의한 평가 점수의 합은 비교대상 중 Groovy가 가장 높았다. 이는 Groovy가 각 비교기준에서 다른 비교대상과 큰 차이가 없는 기능을 제공하면서 한글 처리와 다른 언어와의 인터페이스 등에서 우수한 특성을 보였기 때문이다. Groovy는 신생 스크립트 언어로서 다른 언어의 좋은 특징을 많이 반영하고 있다. 위와 같은 평가 결과의 원인은 부분적으로는 Groovy의 이런 특징 때문이 아닌가 생각된다.

5. 결론

스크립트 언어의 우열을 판단하기는 어렵다. 그렇지만 각 언어 간의 비교를 통해 특정 분야에 있어서 강점을 보

이는 스크립트 언어를 발견할 수 있었다. 문자열을 처리하는 작업에는 Perl, Python, Tcl이 유리하고 CGI 작성에는 Perl이나 Python이 유리하며, 짧은 코드로 간단히 프로그래밍을 하기에는 Lua가, 한글 글자 처리나 표현, 전반적으로 쉬운 프로그래밍에는 Groovy를 사용하는 것이 유리하다는 것으로 조사되었다.

쉬운 프로그래밍을 위해서는 기본적으로 쉬운 문자열 처리 연산, 사용하기 쉬운 객체, 다국어 처리 지원이 언어에서 필수적으로 지원되어야 할 것으로 생각되며, 미디어 파일 처리, 쉬운 GUI 구성과 처리와 같은 고급 기능도 라이브러리가 기타 방법으로 지원되어야 할 것이라고 생각한다.

참고문헌

- [1] Scripting Languages, Wikipedia (http://en.wikipedia.org/wiki/Scripting_language)
- [2] 스크립트언어 비교 페이지 (<http://merd.sourceforge.net/pixel/language-study/scripting-language/>)
- [3] Source Forge (www.sourceforge.net)
- [4] David Thomas, Dave Thomas, Chad Fowler, Programming Ruby: The Pragmatic Programmers' Guide 2nd Ed., Pragmatic Bookshelf, 2004
- [5] Larry Wall, Jon Orwant, Tom Christiansen, Programming Perl, 3rd Ed., O'REILLY, 2000
- [6] M. F. Sanner. "Python: A programming languages for software integration and development.", The Scripps Research Institute, 1999
- [7] Mark Lutz, PROGRAMMING PYTHON, 2nd Ed., O'REILLY, 2001
- [8] John K. Ousterhout, Tcl and the Tk Toolkit, Addison-Wesley, 1994
- [9] R. Ierusalimschy, L. H. de Figueiredo, W. Celes, "Lua—an extensible extension language", Software: Practice & Experience 26 #6, pp.635-652, 1996
- [10] R. Ierusalimschy, L. H. de Figueiredo, W. Celes, "Lua 5.0 Reference Manual", Technical Report MCC-14/03, PUC-Rio, 2003
- [11] groovy 공식페이지(<http://groovy.codehaus.org>)
- [12] Terrence W. Pratt, Marvin V. Zelkowitz, Programming Languages: Design and Implementation, 4th Ed., Prentice Hall, 2001
- [13] John K. Ousterhout, "Scripting: Higher Level Programming for the 21st Century", 1997