

차세대 객체 지향형 프로그램 언어들을 중심으로 한 자유 소프트웨어 운동

한서대학교 이재용*
청강문화산업대학 홍종준

1. 서론

자유 소프트웨어 운동이 성공의 결실을 맺기 위해서는 공개 소스 코드들이 널리 활용되고 배포될 수 있도록, 번역하고 실행할 수 있는 공개된 언어가 필수적이다[1]. 리치와 커니건이 만든 C와 호환되는 gcc가 존재하는 상황에서 자유소프트웨어 운동에 기폭제가 될 수 있는 언어들은 3세대 객체지향형 스크립트 언어들이라고 할 수 있을 것이다[2,3].

전통적으로 인터프리터 방식은 컴파일러 방식 언어에 비해 언어가 단순하고 수행 속도가 떨어져 크게 주목받지 못했다. 하지만 1990년대 후반에 하드웨어의 성능이 놀라울 정도로 향상됐고, 기존 자원을 빠른 속도로 효율적으로 묶어내는 것이 중요해졌다. 또, 스크립트 언어 자체가 개념적, 기능적으로 향상된 것도 주요한 원인이다[4,5].

대표적인 3세대 객체지향형 스크립트 언어로 파이썬(Python), 루비(Ruby) 등이 있고, 펄이 3세대적 특성을 구현하기 위해 개발되고 있다[6,7,8]. 이들 언어의 공통적인 특징은 객체지향 스크립트 언어이고, 간결한 문법 구조를 채택하여 언어의 코어가 상대적으로 작은 대신 풍부한 라이브러리로 재사용성과 개발 생산성을 높였으며, 다른 언어와 잘 붙는다(gule)는 것이다. 또 소스코드가 상대적으로 적으며, 따라서 프로그램의 개발속도는 빠르고, 개발 후 테스트 버전을 실행하는 기간에 실행속도가 문제가 되는 부분을 C나 C++ 같은 언어로 재작성함으로써 개발을 성공적으로 할 수 있도록 돕는다는 것이다[6,9,10]. 즉, 컴퓨터에게는 복잡한 일을, 프로그래머에게는 즐거움을 돌려준다는 본연의 목적에 충실하게 되는 것이다.

2. 3세대 스크립트 언어들의 자유소프트웨어 운동

2.1 파이썬(Python)

암스텔담의 귀도 반 로섬(Guido van Rossum)에 의해 개발된 인터프리터 언어인 파이썬은 자유소프트웨어들에 걸맞게 1989년 크리스마스 연휴를 즐기 위한 취미 프로젝트로 시작하였다. 파이썬은 차세대 스크립트 언어 중에서 가장 주도적으로 자유 소프트웨어 운동을 주도하는 언어이다. 개발 단계부터 문제해결 그 자체에 더 많은 노력을 집중할 수 있도록 설계되었고 문제해결의 흐름을 한눈에 볼 수 있도록 해주는 명료한 형식을 가진다. C/C++를 연계하여 상대적으로 속도가 빨라야 하는 부분을 보완할 수 있다. 코딩을 하는 시간을 C/C++ 보다 1/5~1/10로 단축 가능하므로 개발 시간을 단축해야 하는 작업에 적합하다. 이와 같이 다른 언어와 접착이 가능한 것은 분리된 소프트웨어 컴포넌트들을 아주 쉽고 유연한 방법으로 연결하는 접착제 언어이기 때문이다[4,5,6].

즉, 여러 다른 프로그래밍 언어들과 쉽게 결합이 가능하고, CGI나 애플리케이션 프로그래밍 등 여러 분야에 응용이 쉬운 새로운 프로그래밍 언어이다. 이들 모두는 파이썬이 시스템 언어와 스크립트 언어의 절묘한 중간에 위치하기 때문에 가능한 일이다[4,5,6].

파이썬의 자유소프트웨어 운동은 1994년에 USENET 뉴스 그룹을 통하여 전파되기 시작하였다. 2002년 4월말 한국 파이썬 사용자 모임이 탄생하여 국내 자유소프트웨어 운동에 활력소가 되었다.

이러한 파이썬은 다른 언어들과 결합하여 다양한 프로그램 언어들을 양산하고, 이것들이 파이썬의 자

* 중신회원

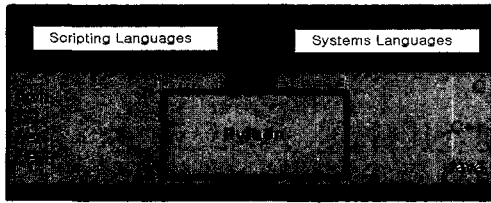


그림 1 파이썬과 다른 언어들

유소소프트웨어 운동에 활력을 불어 넣고 있다. 그것들을 가능하게 만드는 것이 표 1에서 보이고 있는 관련 도구들이다.

표 1 파이썬 관련 도구들

윈도우 관련 도구들	wxPython, Pmw, PyQt, PyKDE
애플리케이션 서버	zope, medusa
C/C++ 인터페이스 모듈	swing

Pmw(Python mega widget)은 파이썬-tkinter 모듈을 이용해서 고 수준의 합성된 위젯을 만들기 위한 툴킷이다[11]. 초보자에게도 다가가기 쉬운 프로그래밍 언어이다. 원래, Tkinter는 Tcl/tk를 파이썬에서 사용하기 위한 인터페이스이다. 파이썬과 Tkinter를 이용하면 매우 빠르게 GUI 프로그램을 작성할 수 있다. 속도가 그다지 빠르지는 못하다는 단점도 있으나 프로그래머의 부담을 줄인다.

wxPython은 파이썬에서 GUI 프로그래밍할 수 있게 해 주는 모듈이다. C++에서 사용하는 wx Windows라는 라이브러리를 파이썬에서 쓸 수 있도록 새롭게 만든 것이다[6]. wxPython이 Tkinter 보다 좋은 점은 실행속도가 Tkinter보다 빠르다는 점과 화면이 미려하다는 점이다. 몇 가지 버그가 있어 Tkinter보다 안정성이 떨어진다는 점이다. Tkinter와 마찬가지로 대부분의 플랫폼에서 사용할 수 있다[4,5].

조프(zope)는 고성능 동적 웹 사이트를 구축하기 위해 파이썬으로 작성된 공개 애플리케이션 서버이다. 플랫폼에 독립적이고 대부분의 작업을 웹을 통해서 처리할 수 있게 해준다. 막강한 획득(acquisition) 기능을 통해 객체 재사용 및 확장을 용이하게 해주고 HTTP, FTP, WebDAV 등 다양한 프로토콜을 지원한다[4].

Medusa는 인터넷 서버 프레임워크를 제공하는 고성능 TCP/IP 서버 아키텍처이다. 대화형 인터넷 서버와 클라이언트의 프레임워크로 적합하다. HTTP를 포함한 대부분의 서버와 클라이언트로 사용 가능하다. 단일 프로세스로 처리되며 가상메모리로 각 접속별로 메모리를 효과적으로 공유, 관리하여 메모리 낭비와 페이지징을 줄여주고 동기화의 시간을 허비하게 되는 것을 막을 수 있다[12].

Swing은 C나 C++로 만들어진 프로그램을 파이썬에서 사용할 수 있게 해주는 프로그램이다. Swing은 C로 만든 모듈을 파이썬에서 쓸 수 있게 하는 과정을 대신 처리해 준다.

파이썬을 이용하여 만든 상업적 제품들은 대단히 많다. 특히, 인터넷 White Page로 유명한 Four11, URL 서버와 검색엔진으로 유명한 Google과 Infor seek에서 사용하는 우편서비스, 레드햇 리눅스의 설치 프로그램인 아나콘다 등이 대표적이다. 그밖에 NASA의 존슨 스페이스 센터에서는 현재 구현된 모든 셸과 펄 프로그램을 파이썬으로 바꾸고 있다. 뿐만 아니라 고수준의 파이썬 제어 모듈들이 다른 언어에 영향을 준 서브루틴 라이브러리에 의해 구현되어 활용성이 높다. 그 예로는 Grail extensible 인터넷 브라우저, GNU Mailing list manager(mailman) 등이 있다[13,14].

2.2 루비(Ruby)

루비는 일본의 마츠모토 유키히로가 만든 공개 소프트웨어이다. 인터프리터형 언어이므로, 수정 및 수행을 빠르고, 쉽게 할 수 있다[15]. 일상적인 처리를 간단하게 프로그래밍할 수 있다. 따라서, 개발 싸이클이 짧다[16].

루비의 변수나 식에는, 형 선언이 없다. 컴파일 할 때 형 체크가 생략되며, 프로그램 실행 시에 형을 검사한다. 이것으로 인해, 프로그램이 간결하고 유연하게 되는 경향이 있다. 또한, 형이 없는 것은 식이나 변수이고, 객체에는 각각의 형이 확실히 있다. 이것을 '동적형'이라고 한다. 루비의 변형에는 어떠한 형의 값에도 대입할 수 있다[8].

루비의 정규 표현 장치는 GNU awk(gawk)에서 사용되고 있던 정규 표현 루팅을 개조한 것이다. 이 정규표현은 원래 GNU Emacs용으로 개발된 것이다 [17].

루비는 처음부터 객체 지향 언어로 설계되어 삽입 기능도 클래스 라이브러리로 구성되어 있다. 언어 자체가 하나의 개념에 기초해서 설계되어 있기 때문에, 사양에 통일감이 있다[15]. 객체 지향이라는 하나의 패러다임으로 통일된 언어 구조로 구성된다[16].

루비에는 사용하지 못하게 된 객체는 자동으로 루비가 회수한다. 이 기능으로 프로그래머가 어느 객체는 언제까지 효력이 있는 지를 관리할 필요가 없고, 실수로 그 사용불가능하게 되거나 사라지거나, 필요 없는 것을 잘못해서 방치하는 등의 예러가 없어진다[15]. 변수가 참조한 문자열 객체는 다음의 내용을 처리하기 위해 할당되면서 어디에서도 참조되지 않게 되면 자동으로 정리한다[16].

표 2 루비 관련도구들

애플리케이션 서버	Petit+Server, Roach
윈도우 관련 도구들	Locana GUI builder, Riverstone, SpecRuby, Ruby/Tk widget

루비에는 스래드 기능이 있다. 루비의 스래드 기능은 유저 레벨 스래드이기 때문에, 어느 OS에서도 동작한다. 어디에서도 동일한 기능을 수행하므로, 안심하고 가볍게 스래드를 사용할 수 있다[15].

루비의 개발은 주로 리눅스에서 이루어지고 있다. 그러나, FreeBSD를 비롯한 각종 유닉스계 OS, 윈도우 NT/98/95 등에서도 작동한다. 더욱이 동일 스크립트가 모든 플랫폼에서 거의 같이 동작한다[15].

Ruby의 자유 소프트웨어 운동은 ruby-list라는 메일링 리스트에서 행해진다. Ruby의 메일링 리스트는 5개가 있다[18].

- (1) ruby-list@ruby-lang.org : 메인 메일링 리스트
- (2) ruby-dev@ruby-lang.org : 개발자용 메일링 리스트
- (3) ruby-ext@ruby-lang.org : 확장 라이브러리 제작자용의 메일링 리스트
- (4) ruby-math@ruby-lang.org : “Ruby와 수학”에 대한 메일링 리스트
- (5) ruby-talk@ruby-lang.org : 영어로 된 메일링 리스트

실제로 널리 사용되고 있는 루비로 제작된 도구로는 FreeBSD의 패키지 관리 도구 중 하나인 port upgrade가 루비를 사용하였다.

3. 차세대 자유소프트웨어 운동

3.1 Pandora 엔진

판도라 엔진은 다목적 객체지향 소프트웨어 개발 도구이다. 이 엔진은 MOO(Modular Object Oriented) 기술에 기반을 둔 공개된 소프트웨어이다. 이 소프트웨어는 애플리케이션과 게임 개발 모두를 적용된다. 상호 플랫폼 개발을 위한 MS 윈도우와 리눅스 모두를 지원한다. 뿐만 아니라, 쉬운 디버깅을 위한 실시간 디버그 출력, Integrated Zip 압축, PNG, PCX, JPEG, IFF 그림 파일을 지원한다. 또, 스트리밍 용량을 포함하는 Sample Playback을 위한 오디오를 지원한다. 디지털 뮤직 지원을 위하여 MOD, XM, IT S3M 형식을 처리한다[19].

다른 멀티 플랫폼과의 차이점은 Modular Object Orientation(MOO)이다. 언어는 아니지만 많은 양의 바이너리 파일과 공유 라이브러리들의 모임이다.

판도라 개발자들은 그림 2의 다이어그램에서 보이고 있는 개발자 공간 안에서 홀로 작업하며 필요한 모듈을 만들거나 객체지향 프로그램을 생성하는 스크립트나 클래스를 만들 수 있고 실행 프로그램을 생성할 수 있다.

판도라 엔진에서 설계된 MOO 설계 모델은 최근의 객체지향 구현의 모든 특성들을 포함하고 있다. 클래스, 객체, 다형태성(polymorphism), 상속, 메소드들을 포함한다. 다른 많은 특성들이 제공되며 다음과 같은 특성을 가진다.

- (1) Action 지원
- (2) 공유와 통신
- (3) 지원적 상속

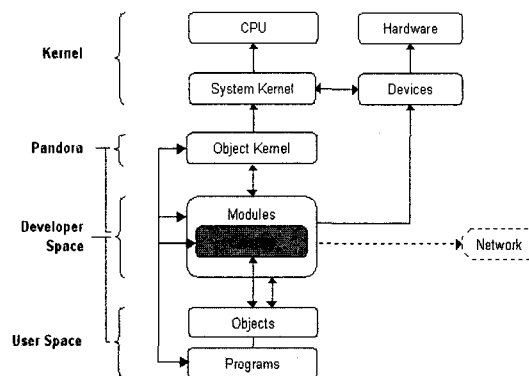


그림 2 판도라 엔진의 개발공간

- (4) 다형태성과 요약
- (5) 동적 자원 추적

Action은 실세계의 구문론과 훨씬 유사한 방법의 함수를 호출하는 방법을 만든다. 객체들은 private, shared, public으로 선언될 수 있다. 객체를 공유하면 다른 프로그램은 프로그램과 결합하는 프로그램과 상호 작용할 수 있다. 객체 네트워킹이 지원되어서 네트워크 상에서 객체들은 공유되고 분배되고 상호 교류된다. 클래스 상속은 외부 자료 파일과 특별한 특성을 존재하는 임의의 클래스에 더할 수 있다. 다형태성은 시스템 내부에서 다항체 필드 동작뿐만 아니라 action 서비스를 통해서 지원된다.

다음의 그림 3은 판도라에 의해서 정의되는 모듈들과 클래스들의 일부를 서술한 것이다. 시스템의 앞부분에는 객체 커널이 놓여져 있다. 커널에 의해서 제공되는 함수 옆에는 많은 내부 클래스들이 시스템이 기동하여 실행을 할 수 있는 필요성이 정의된다. 하드웨어 모듈은 다음 우선순위의 레벨에 위에 있으며 시스템 위에서 놓여진 추상 계층을 표현한다. 비록 하드웨어 모듈이 낮은 레벨의 디바이스들과 거의 배타적으로 통신한다고 해도, 새로운 모듈을 요구가 있을 때 직접적으로 하드웨어를 액세스할 수 있는 새로운 모듈을 만드는 것이 가능하다. 이것은 항상 보다 빠른 코드를 생성할 수 있지만, 만일 디바이스가 시스템 위에서 사용한다면 개발 시간이라는 면에서는 빠르지 않다.

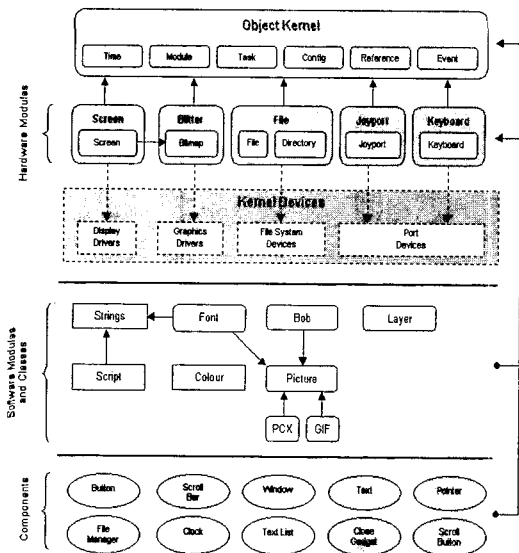


그림 3 판도라 엔진의 모듈과 클래스

3.2 Atilla

Atilla는 “XML과 파이썬 기반의 소프트웨어 실행 환경”이라고 정의할 수 있다. Atilla 프로젝트는 국내에서 개발자들이 주체가 되는 자유소프트웨어이다. 즉, 인터넷 기술의 다양한 변화들에 대한 새로운 방향을 제시한 것이다. Atilla는 파이썬과 XML 그리고 인터넷을 통한 방향제시라고 볼 수 있다. 이 자유소프트웨어 운동의 목표는 “파이썬과 XML을 이용한 클라이언트/서버 프로그램처럼 웹 개발을 편리하게 해줄 수 있는 실행환경을 구축하고, 이를 오픈 소스 프로젝트로 세계에 공개한다”라는 것이다.

Atilla 프로젝트는 다음과 같이 세 개의 개발 범위를 가진다.

- (1) 코아 프레임워크 개발
- (2) Atilla 브라우저 개발
- (3) Atilla IDE 개발

이를 위하여 개발된 코아 프레임워크는 그림 5와 같다.

Atilla 코아 프레임워크 개발을 위하여 XPL 규약의 정의, XPL 프로세서 개발, 번역 프로세스 개발, GUI/DB 객체 라이브러리 개발, Error Handler 개발로 나뉘어 진다. XML(eXtensible markup on Python Language)은 Atilla 환경에서 새로이 구성된 스크립트형 언어이다. 전체 구조 중 데이터 부분은 XML을 적용하고, 프로세스 부분은 파이썬을 적용하는 것이다. XML은 문서의 데이터화를 실현할 수 있는 언어로써 인터넷에서의 문서 교환의 표준이라고 할 수 있으며, 객체의 구조, 속성, 메소드의 표현이 쉽고, 처리하기 쉽다. 또, 객체 지향형 개발도구의 작성에 최적의 표현방식이다. 파이썬은 선택한 이유는 객체 지향

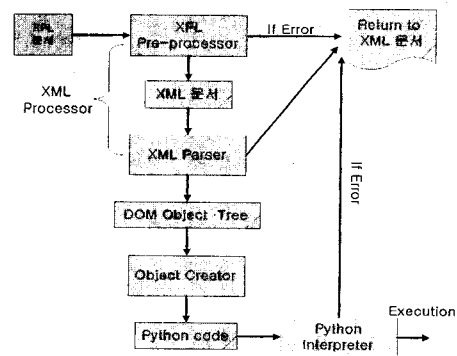


그림 4 Atilla 작업 흐름도

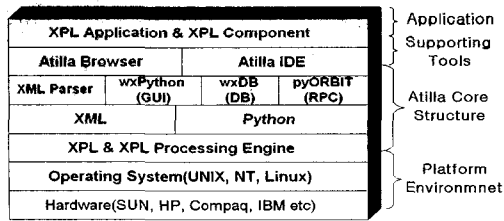


그림 5 Atilla 코어 프레임워크

형 스크립트 언어로 생산성이 높기 때문이다. 이 기종 환경에서 자유로운 사용이 가능하다. 프로토타이핑에 적합한 언어이다. 인터넷에서 사용되는 스크립트 언어보다 훨씬 광범위한 모듈을 보유하고 있다.

Atilla 브라우저 개발을 위하여 XPL 문서를 브라우저하는 기능을 가지며 HTML문서도 브라우저할 수 있다.

Atilla IDE 개발을 위해서 화면 디자이너, tool Pallet 개발, 소스코드 편집기, Property/Event 편집기 개발로 나뉘어 진행되었다.

이러한 많은 내용을 개발함에도 불구하고 파이썬을 이용하여 개발함으로써 총 20 Man Month로 매우 짧은 개발비용을 가지고 개발되었다. 이는 자유소프트웨어 운동의 가지는 효과를 단적으로 보여주는 예이다.

3.3 Kew

Kew는 폐쇄와 수집의 사용을 확장하는 것을 통하여 대단히 풍부한 힘을 얻는 간단하고 세련된 객체지향형 언어이다. 또, 많은 강력한 특성들을 가지고 있으며 작고 간단한 API를 지향하고 있다. Smalltalk, Scheme, Miranda, Java, Perl 등의 특성에 영향을 받았다. Kew는 객체가 가지는 어떤 형태에 관한 혼동되지 않는 강력한 동적 형태의 시스템을 가진다. Kew는 자동화된 빠른 garbage collection을 제공하고, 메모리 관리 자체를 하지 않아도 되며 메모리와 고장의 누출이 없다[20].

자유소프트웨어운동을 주도하는 다른 객체프로그램 언어와 마찬가지로 간단한 구문 높은 생산성, 강력한 내장 collection과 제어 구조들, 좋은 보안의 특성을 가지며, 독특한 특성은 다음과 같다.

- (1) 하나의 인터프리터, JIT 가상 머신과 자연코드
- (2) 모듈러
- (3) 예외처리

(4) 내장성과 확장성(embeddable and extensible)
 유통성 있는 인터프리터/컴파일 기능은 컴파일러를 위해 기다리지 않고, 빠르게 개발을 할 수 있다는 의미이다. 따라서 최대속도로 컴파일 된다.

Kew의 모듈러에서는 Kew 소스코드는 번역 단계 없이 스크립트에서 직접 실행한다. 또, 소스는 안전하게 분산된 기계-독립 바이트 코드로 번역되고, JIT(Just-In-Time) 가상 머신에서 실행된다. C 번역기를 통한 native code로 소스코드나 바이트코드가 번역되어 DLL/공유라이브러리를 만들어 낸다.

Kew는 객체와 블록 모두를 가지며, 형식 모두가 폐쇄이다. 폐쇄는 간단하고 논리적인 코드를 만들어 내는 절차에 매우 대단한 노력을 하는데 사용될 수 있다. 상속이 없고 어휘의 영역과 객체 구성이 매우 간단하여 상속의 복잡성과 비유지보수성이 필요 없는 것을 이용하여 공유를 구현한다.

Kew는 클래스 연장(non-local jumps)이 내장되어 자바 형식의 jumping 예외의 구현을 할 수 있을 뿐만 아니라, 하나의 영역 내에 있는 깊은 중첩 루프로 jump 할 수 있다.

Kew는 매우 작은 코드와 필연적으로 매우 작은 최소 footprint를 가진다. 모든 것이 이것위에 놓여져서 적은 메모리 응용을 위한 Kew 전형을 만들어 낸다. Kew의 잘 정의된 C 인터페이스와 사전 수행 처리모델 역시 스크립트 언어로써 애플리케이션들에서 내장할 수 있다.

3.4 elastic

elastic은 마르코 펜타레오니에 의해서 설계되고 개발되었다. C와 같은 구문을 가지고 있는 이식성 있는 고수준의 객체 지향 인터프리터 언어이다. 위의 다른 언어들과 같이 공개 소프트웨어이며, C와 유사한 구문으로 친숙하고 가벼운 바이트 코드 번역을 한다. 그밖에 주요 특성들은 다음과 같다[21].

- (1) garbage collection
- (2) 메타 프로그래밍
- (3) functional programming 지원
- (4) 계층적 네임 공간
- (5) 많은 기본 type
- (6) C의 확장과 내장

자동적이고 빠른 garbage collection 기능을 가지며 smalltalk의 영향으로 메타 프로그래밍을 지원하는 객체지향형 특성을 가진다. 또 어휘 영역의 폐쇄

와 같은 개요와 functional 프로그래밍의 특성을 포함하고 있다. 또, 동적 배열, 디셔너리, 심볼 등 풍부한 기본 type들을 가지며 C를 확장할 수 있는 기능을 포함한다.

elastiC는 C, Smalltalk, Scheme, 파이썬에 강력한 영향을 받았고, 이 언어들의 최상의 특성들을 융합하려고 시도되었으며 계속해서 이 특성들을 유지하고 있다. elastiC의 배포는 elastiC emacs 모드와 elastiC JED 모드를 위한 파일을 함께 포함하고 있다.

4. 결론

파이썬, 루비 등의 3세대 객체 지향형 스크립트 언어들은 소리 없이 놀라운 정도의 속도로 개발되어 많은 가능성을 제시하고 있다. 이러한 언어를 이용하거나 강력한 영향을 받은 언어들이 자유소프트웨어 운동을 주도하고 있다.

이러한 언어들의 공통적인 특성은 자료의 형식이 없으며, 개발 속도가 빠르며, 단점을 보완하기 위해서 다른 언어와의 접합이 쉽다. 또 풍부한 관련도구들이 개발되어 있으며, 인터넷 연결을 쉽게 하기 위한 기능들이 탑재되어 있다. garbage collection과 메모리 관리기능을 포함하는 경향을 가진다.

이러한 특성과 더불어 차세대 스크립트 언어들은 MOO, JIT 가상머신(Just-IN-Time Virtual Machine), 메타 프로그래밍, functional 프로그래밍의 기능을 추가하는 추세이다.

더욱이 이들을 이용한 상업적으로 성공한 많은 도구들이 자유소프트웨어 운동이 지향했던 소프트웨어를 통한 자유의 표현이라는 목표가 허상이 아닌 것을 증명하는 계기가 되고 있다.

파이썬을 이용한 Atilla 프로젝트의 경우, 국내에서 주도하는 최초의 프로그램 언어계열의 자유소프트웨어 운동이라는 면에서 특히 주목받고 있다. 이러한 경향은 3세대 객체지향형 스크립트 언어를 이용한 새로운 형태의 언어 창조하는 면에서 자유소프트웨어 운동의 방향을 제시하는 시금석이라고 할 수 있겠다.

그러나 국제화에는 성공하기 위해서는 멀티 바이트 문자열 처리 이상의 능력을 가져야 한다. 현재는 루비가 멀티 바이트 문자열뿐 아니라 UTF-8도 다룰 수 있으므로 지금까지는 약간 앞서 있다.

앞으로는 이러한 새로운 언어들이 상호 영향을 주

면서, 여러 가지 특성들이 일정한 흐름을 보일 것이며, 그 성공여부가 자유소프트웨어 운동의 많은 영향을 줄 것이다.

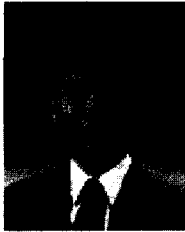
참고문헌

- [1] "Free Software," 인터넷 URL <http://www.free-soft.org>
- [2] "GCC Home Page," 인터넷 URL, <http://www.gnu.org/>
- [3] "GCC Home Page-GNU Project," 인터넷 URL, <http://gcc.gnu.org/>
- [4] "한국파이썬 사용자모임," 인터넷 URL, <http://users.python.or.kr:9080/PyKUG>
- [5] "파이썬(Python) 정보광장," 인터넷 URL, <http://www.python.or.kr:8080/>
- [6] "Python Language Website," 인터넷 URL, <http://www.python.org>
- [7] "Learning Perl," 인터넷 URL, <http://www.myperl.pe.kr/>
- [8] "Ruby," 인터넷 URL, <http://www.ruby-lang.org/>
- [9] "Atilla," 인터넷 URL, <http://www.atilla.co.kr/>
- [10] 박응용, 점프투 파이썬, pp.18~23, 정보게이트, 서울, 2001.
- [11] 정일권, "파이썬 프로그래밍", 리눅스 매거진, 제18호, pp.198~207, 2001. 3.
- [12] "Medusa : A High-Performance Internet Server Architecture," 인터넷 URL, <http://www.nightmare.com/medusa>
- [13] "Grail Home Page," 인터넷 URL, <http://grail.sourceforge.net/>
- [14] "MailMan, the GNU Mailing List Manager," 인터넷 URL, <http://www.list.org/>
- [15] Tomoya Akamatsu, "Ruby," 리눅스 매거진, 제4호, pp.171~178, 2000. 1.
- [16] Tomoya Akamatsu, "Ruby," 리눅스 매거진, 제5호, pp.165~172, 2000. 2.
- [17] Tomoya Akamatsu, "Ruby," 리눅스 매거진, 제6호, pp.175~180, 2000. 3.
- [18] Tomoya Akamatsu, "Ruby," 리눅스 매거진, 제18호, pp.229~236, 2001. 3.
- [19] "The Pandora Engine," 인터넷 URL, <http://>

www.rocklyte.com/pandora

- [20] "Kew Programming Languages," 인터넷 URL, <http://dunkworks.com/projects/kew/>
- [21] "elasitC World," 인터넷 URL, <http://www.elasticworld.org>

이재용



1985년 인하대학교 전자계산학과(이학사)
1990년 인하대학교 전자계산학과(이학석사)
2000년 인하대학교 전자계산학과(공학박사)
2000~현재 한서대학교 컴퓨터통신공학과 조교수
E-mail : jylee@hanseo.ac.kr

홍종준



1991년 인하대학교 전자계산학과(공학사)
1993년 인하대학교 전자계산학과(공학석사)
2002년 인하대학교 전자계산학과(공학박사)
1999년~현재 청강문화산업대학 컴퓨터소프트웨어과 조교수
E-mail : jjhong@mail.chungkang.ac.kr

● HCI 2003 학술대회 ●

- 일 자 : 2003년 2월 10~13일
- 장 소 : 보광피닉스
- 주 최 : 인간과컴퓨터상호작용연구회
- 문 의 처 : 대회사무국 이선경
Tel. 02-552-8321(교환 : 319)