

프로그래밍 언어들 간의 인지적인 거리측정에 관한 연구

(A Study of the Measurement of the Perceived
Distances among Programming Languages)

이충권*, 유상진*, 진상민**

(Choong Kwon Lee, Sangjin Yoo, and Sangmin Jin)

요약 컴퓨터관련 교육기관들은 시대의 변천에 따라 다양한 프로그래밍 언어들을 교육시켜 왔으나 많은 학생들은 컴퓨터 활용의 측면에서는 능숙하지만 프로그래밍은 배우기 어려운 분야로 인식되고 있다. 프로그래밍에 대한 거부감은 컴퓨터관련 전공에 대한 기피현상으로 이어졌고, 전 세계적으로 프로그래머 인력의 부족 문제가 대두되고 있는 실정이다. 본 연구는 프로그래밍 언어들에 대해서 프로그래머들이 지각하고 있는 언어들 간의 유사성을 조사함으로써 각 언어의 상대적인 위치를 파악하고, 좌표평면 상에 나타내어 각 차원이 의미하는 바를 추정하고 규명하였다.

핵심주제어 : 프로그래밍 언어, 다차원척도법, 인지적인 거리

Abstract Although the educational institutions of computers have evolved to train students in pace with the changes of computing, many students still believe that computer programming is hard to be mastered. The reluctance of students toward programming languages have been tied to the decrease of computer-related majors and now many countries are in short of computer programmers. By investigating the similarities among eleven popular programming languages, this research is intended to measure the relative distances and understand the dimensions that scattered the languages.

Key Words : Programming Languages, Multi-Dimensional Scaling, Perceived Distances

1. 서론

오늘날 우리가 사용하는 컴퓨터는 1945년 처음 개발되었으며, 많은 사람들의 노력으로 엄청난 발전을 이룩하여 20세기 최고의 발명품으로 자리매김하였다. 컴퓨터의 활용분야는 과학기술의 개발로부터 출발하여 기업 활동으로 확산되었고, 업무자동화, 부가가치

창출 등 기업경영 환경에 있어 많은 변화를 초래하였다. 이러한 컴퓨터의 놀라운 발전에 있어서 프로그래밍 언어의 존재는 그 중심에 서있었다. 프로그래밍 언어는 매우 빠른 속도로 발전하여 약 2500개가 등장하였고, 지금도 새로운 언어의 개발이 시도되고 있다. 그러나, 소프트웨어의 개발에 있어서 필수적인 역할을 하고 있음에도 불구하고, 컴퓨터 프로그래밍은 여전히 배우기 어려운 분야로 인식되고 있다. 컴퓨터 초보자들은 물론이고, 경험이 많은 프로그래머들조차도 새

* 계명대학교 경영정보학과

** 대구일보 전산실

로운 언어와 기술을 배우고 적용하는데 있어서 많은 부담을 느끼고, 타직종으로의 이직을 고려하는 매우 중요한 이유들 중의 하나로 알려져 있다(권남훈의 11인, 2001). 특히, 정보시스템의 개발에 있어서 어떤 프로그래밍 언어를 사용할 것인가를 결정하는 것은 매우 중요한 의사결정이다. 하지만, 보유하고 있는 인력이 익숙하지 않은 프로그래밍 언어로 개발을 시도할 때에는 별도의 교육과 훈련이 필요하고 훨씬 더 많은 시간을 소모하게 된다.

이처럼 많은 프로그래밍 언어들이 존재하고 실제로 널리 쓰이고 있지만, 어떤 언어를 먼저 배우는 것이 유리한지에 대해서는 연구가 거의 없다. 또한, 새로운 언어가 등장했을 때, 기존의 프로그래머들이 새롭게 배우고 적용하는데 얼마나 걸릴 지에 대한 예측도 어렵다. 이것은 프로그래밍 언어에 대한 기존의 연구가 기술적인 면에 치우쳐서 성능의 개선에 초점을 맞추었기 때문이다. 본 연구는 프로그래밍 언어들이 서로 어떤 관련을 맺고 있는지를 조사하고 분석함으로써 언어들을 인지적인 측면에서 바라보고자 한다. 다양한 프로그래밍 언어들이 존재하기 때문에, 각 언어들 간의 인지적인 거리를 밝히는 것은 매우 중요하다. 예를 들어, C 언어를 잘 다루는 프로그래머가 Java 언어를 습득하려면 얼마나 많은 노력과 시간을 들여야 할지를 미리 예측하는 것이 중요하기 때문이다.

본 연구의 목적은 프로그래밍 언어들에 대해서 프로그래머들이 지각하고 있는 언어들 간의 유사성을 조사함으로써 각 언어의 상대적인 위치를 규명하는데 그 목적이 있다. 즉, 프로그래머들이 지각하고 있는 프로그래밍 언어들 간의 심리적 위치를 규명하고 이를 비교분석하고자 한다. 본 연구는 다음과 같은 순서로 진행된다. 첫째, 프로그래밍 언어들 간의 인지적 거리를 조사함으로써 언어들 간의 상대적인 위치를 표현하는 지도를 작성한다. 둘째, 첫 번째 연구 과제를 통해 파악한 각 언어들의 상대적 위치를 나타내는 좌표평면 상에서 각 차원이 의미하는 바를 추정하고 규명한다. 끝으로, 인구통계학적 변수에 따른 유사성 지각도의 위치관계의 차이를 비교 분석한다.

2. 이론적 배경

2.1 프로그래머에 관한 연구

기존의 연구들은 프로그래머의 역할이 역사적으로 변해왔다는 것을 밝혔다. Todd et al. (1995)은 1970년부터 1990년까지 신문에 게재된 구인광고를 수집하고 분석하여 프로그래밍과 같은 테크니컬한 스킬이나 지식 뿐만 아니라 사용자 요구사항에 대한 이해나 팀워크 같은 테크니컬하지 않은 기술까지도 프로그래머들에게 요구되고 있다고 하였다. 비슷한 결과를 제시한 다른 연구들도 있었다. 예를 들어, 의사소통능력(Chen, et al., 2003), 논리적 사고능력(Davis, 2003), 그리고 비즈니스 지식(Noll and Wilkins, 2002; Cheney and Lyons, 1980) 등이 프로그래머들에게도 중요하다고 밝혀졌다. 이러한 연구 결과는 IT분야 전반에 걸쳐서 나타났고, IT인력에게 비즈니스에 대한 이해를 요구하게 되었다(Markus and Benjamin, 1996; Nelson and Coopridge, 1996; Davis, 2003). 그러나, IT인력의 기술적 능력은 여전히 업무수행에 있어서 가장 기본적이고, 조직내에서 자기의 효능감을 느끼게 만드는 중요한 요소인 것으로 판단된다(유상진과 이유진, 2010).

프로그래머의 역할이 다양해졌을 지라도, 주 업무는 시스템을 실제로 개발하고 구현하는 일이기 때문에, 필연적으로 C++나 Java와 같은 프로그래밍 언어를 사용할 수밖에 없다. 따라서 많은 시간을 프로그래밍 언어를 사용하여 애플리케이션을 개발하는데 쓰고 있고, 새로운 프로그래밍 언어에 대한 훈련과 교육이 지속적으로 필요하다. Bailey와 Mitchell (2006)은 미국 노동부 통계국의 자료를 바탕으로 프로그래머에 대한 수요가 조금씩 느린 속도로 높아지고 있기는 하지만, 잘 훈련된 고급프로그래머들에 대한 수요는 여전히 매우 높은 상황이라고 밝히고 있다. 우리나라에서도 IT인력에게 있어서 프로그래밍언어에 관한 지식과 스킬에 대한 요구는 매우 높은 것으로 확인되고 있다(고석하, 2010; 이찬 외 5인, 2009).

2.2 프로그래밍 언어에 관한 연구

프로그래밍 언어는 현재까지 1세대로부터 5세대에 이르고 있고, 약 2500개의 종류가 있는 것으로 알려져 있다. 1세대(1GL: First Generation programming languages) 언어는 컴퓨터가 처음 개발되었을 때부터 지금까지 사용되는 기계어로 반도체에 내장되어 사용되어 진다. 2세대(2GL) 언어는 어셈블리어로 기계어보다 조금 더 쉽게 하드웨어를 제어하기 위하여 개발된 언어이며 하드웨어 드라이버와 같이 고속 알고리즘 구현에 사용된다. 3세대(3GL) 언어는 본격적인 프로그래밍 언어이며 C, 포트란, 파스칼, 베이직, 코볼, 등 절차 지향 언어로써 함수 개념을 사용하여 코드 재사용이 가능하다.

4세대(4GL) 언어는 객체지향 언어로써 GUI(Graphic User Interface) 환경에서 데이터베이스 관련 프로그램을 쉽게 개발 가능한 현업에서 많이 사용되는 언어이다. 대표적으로 비주얼 베이직(마이크로소프트), 델파이(블렌드), 파워빌더(사이베이스)가 있다. Ginzberg와 Baroudi (1986)는 4세대 언어가 프로그래머와 사용자 간의 의사소통을 더욱 편리하게 만들어주었다고 하였고, Gallant (1984)은 4세대 언어가 Cobol과 같은 절차 지향적인 언어들에 익숙한 프로그래머들에게 불리할 것이라고 한 예측은 거의 맞아 들어갔다.

5세대(5GL) 언어는 네트워크 언어이다. 통신의 발달과 보급으로 인해 네트워크를 통하여 서로 다른 운영체제가 연결되고, 운영체제와 상관없이 프로그램이 실행 가능한 5세대 언어가 출현하였다. 대표적인 언어로 썬 마이크로시스템즈에서 개발한 Java와 마이크로소프트에서 개발된 C#이 있다. 김인재(2001)의 연구는 기존의 절차지향적인 언어에 익숙한 프로그래머들이 객체지향언어를 습득하는데 있어서 기존의 경험이 중요한 변수이고, 개인의 혁신성은 크게 영향을 미치지 않는 것으로 나타났다.

새로운 프로그래밍언어를 습득하려면 많은 시간과 비용이 들어간다고 알려지고 있다(Boyle, et al., 2002; Gould and Rimmer, 2000). 특히, Java와 같은 언어는 기존의 전통적인 절차지향적인 언어인 코볼이나 파스칼과 같은 언어에 익숙한 프로그래머들조차도 배우기가 어렵다고 한다(Benander, et al., 2003). Campione, et al. (2000)은 Java가 C++보다 생산성이 높기 때문에 Java 프로그래머들이 C++프로그래머들 보다 더 높은

연봉을 받아야 한다고 주장하였고, Dattero, et al., (2003)은 Java와 C++의 경제적인 가치를 비교하여 연구하였는데, Java의 가치가 높기는 하지만 두 언어를 모두 아는 사람이 가장 높은 연봉을 받는다고 하였다.

2.3 프로그래밍 언어들 간의 상호관계

이처럼 정보시스템을 개발하는데 있어서 필수적인 요소인 프로그래밍언어 자체에 대한 기존의 연구는 언어들 간의 관계를 단순히 기술적인 측면에서 역사적인 세대별로 분류하거나 절차지향과 객체지향이라는 이분법적인 기준으로만 분류하는 것이 일반적이었다. 이러한 분류는 실제 개발을 위하여 새로운 언어를 배워야만 하는 프로그래머들에게는 큰 의미를 주지 못한다. 예를 들어, 3세대 언어인 C는 절차지향적이고 5세대 언어인 C++는 객체지향이지만, 서로 비슷한 면을 많이 가지고 있다. 또한, 최근에 등장한 웹프로그래밍 언어들의 경우에는 서로 경계가 모호한 것이 현실이라서 Javascript는 절차지향적인 특성과 객체지향적인 특성을 동시에 가지고 있다. 따라서, 프로그래머의 입장에서는 자신이 보유하고 있는 언어가 새롭게 배우려는 언어와 어느 정도 가까운지를 미리 알 수 없기 때문에, 습득에 투입되는 시간과 노력을 미리 예측할 수 없는 단순한 분류인 것이다.

프로그래밍 언어들 간의 상호관계를 판단함에 있어서 그 유사성이나 상이성을 객관적인 인식에 근거하여 각 언어의 위치를 파악하는 것이 필요하다고 여겨진다. 이러한 유사성과 상이성을 판단할 수 있는 방법으로 제시된 것이 포지셔닝이다. 일반적으로 포지셔닝이란 상품, 서비스, 기업, 등 특정 대상의 이미지를 사람들의 인지 속에 인식시키는 것으로 정의되고 있다. 마케팅 측면에서 포지셔닝은 기업이 자신의 차별적인 이미지를 상품이나 서비스 혹은 특정한 기업활동 등을 이용하여 고객들의 머릿속에 각인시키는 것이다(이훈영, 2004). 포지셔닝맵(positioning map)이란 특정 제품의 경우 소비자가 생각하고 있는 상대적인 위치를 경쟁제품들과의 비교를 통해서 2차원 또는 3차원의 도면으로 작성한 것이다. 마케팅관리자는 포지셔닝맵을 작성하여 경쟁제품들 간의 관계를 분석함으로써 다양한 전략을 수립할 수 있다. 이처럼 프로그래밍 언

어들에 대하여 프로그래머의 지각에 근거하여 인지적 포지셔닝맵을 도출할 수 있다면 새로운 언어를 배우거나 가르칠 때 훨씬 더 좋은 전략을 세울 수 있을 것이라 판단된다. 포지셔닝맵의 작성은 다차원척도법(Multi-Dimensional Scaling)을 이용한 작성방법이 가장 일반적으로 사용된다.

다차원척도법은 브랜드나 기업과 같은 특정 대상에 대한 응답자들의 평가에 내재되어 있는 주요 차원들을 규명하기 위한 분석기법이며, 응답자가 느끼고 있는 다양한 측면의 지각도나 선호도(또는 다변량 자료)를 측정하여 측정된 수치를 토대로 대상간의 복잡한 관계를 그래프 등의 공간에서 시각화하여 보여주는 기법이다. 다차원척도법이란 들쭉 찢어진 개체간의 유사성(또는 상이성) 자료를 토대로 m 차원 공간상에 이들 n 개의 개체를 상대적으로 좌표화하여 형상화(Configuration)하는 기법이다(이재창과 박정섭, 1986). 다차원척도법을 사용하는 경우, 연구대상이 서로 얼마나 다른 특성을 가지고 있는지 나타내는 거리(distance) 또는 비유사성(dissimilarity)이나 연구 대상이 얼마나 비슷한 특성을 가지고 있는지를 나타내는 유사성(similarity)에 대한 측정이 이루어지며, 이러한 측정자료를 대상으로 최적척도법(optimal scaling)이 적용되어 거리나 유사성을 그래프상에 잘 표현할 수 있도록 좌표상의 점으로 변환시켜주는 과정을 거치게 된다(남은희, 2003). 박성준과 김성훈(2004)은 차량선호도에 기초하여 자동차 네비게이션시스템에 대한 소비자들의 감성요소를 분석하는데 다차원 분석법을 활용하였다. 정보시스템 분야의 연구에 있어서도 최근에 고석하(2010)는 다차원척도법을 이용하여 정보기술이나 지식에 대하여 각 직업군이 어떻게 다른 인식을 가지고 있는지를 보여주었다.

본 연구에서는 프로그래머들이 인지하고 있는 프로그래밍 언어들 간의 유사성을 근거로 다차원척도법을 활용하여 포지셔닝맵을 도출하여 분석함으로써 실무에 언어들 간의 상호관계를 이해하고자 한다. 이러한 이해를 바탕으로 새로운 언어를 습득하려 하는 프로그래머들에게 어느 정도의 시간과 노력을 투입해야 하는지에 대한 기초자료를 제공할 수 있을 것으로 기대된다.

3. 연구방법

3.1 자료의 수집

본 연구의 설문조사는 우리나라에 근무하는 시스템 소프트웨어 개발자, 응용 소프트웨어 개발자, 웹프로그래머, 등 관련 실무자를 대상으로 하여 11개의 언어들(Java, C, PHP, C++, Visual Basic, C#, Python, Perl, JavaScript, Ruby, Delphi) 상호 간의 유사성에 응답하는 방식으로 온라인 설문지를 개발하였다. 설문은 웹 브라우저를 통하여 응답하는 방식으로 조사를 실시하였으며, 웹설문에 대한 응답은 MySQL 4.1.22 버전을 이용하여 저장되었다. TIOBE는 Google, Yahoo, 그리고 MSN과 같은 포털사이트에서의 검색순위를 기초로 하여 매월 프로그래밍 언어들의 인기도를 조사하여 순위를 발표하고 있는데, Turing Completeness라는 기준에 근거하여 HTML, XML, 그리고 SQL과 같은 데이터 표현에 가까운 언어들은 조사대상에서 제외하고 있다.

자료수집 방법으로 프로그래머의 방문이 많은 포털사이트(데브피아, PHP School)에 설문지 등록과 소프트웨어 관련 업체 실무자들을 대상으로 설문조사 이메일을 통해 설문조사를 실시하였으며, 총 1,167회의 설문지 열람, 221부의 응답을 회수하여 프로그래밍의 경험이 1년 이하인 경우와 응답이 불성실하거나 일관성이 없다고 판단되는 83부를 제외한 138부를 유효 표본으로 적용하였다.

3.2 조사설계

본 연구의 대상은 통계청이 제시한 전산관련 직업 분류표에 의거하여 프로그래밍과 관련된 업무종사자들을 대상으로 <표 1>과 같이 TIOBE Software(www.tiobe.com)에서 발표된 상위 11개의 프로그래밍 언어들 간의 관계를 도출하고자 설문조사를 실시하였다.

<표 1> 인기 순위 11개 프로그래밍언어

순위		순위 변동	프로그래밍 언어	2011년 9월	2010년 9월과의 차이
2011년 9월	2010년 9월				
1	1	=	Java	18.8%	+0.85%
2	2	=	C	18.0%	+0.86%
3	3	=	C++	8.9%	-0.96%
4	6	↓	C#	6.8%	+1.80%
5	4	↑	PHP	6.6%	-1.77%
6	8	↓	Objective-C	6.2%	+2.79%
7	5	↑	Visual Basic	4.4%	-1.38%
8	7	↑	Python	4.0%	-0.58%
9	9	=	Perl	2.5%	+0.03%
10	11	↓	JavaScript	1.5%	-0.20%
11	10	↑	Ruby	1.4%	-0.47%

4. 연구 분석 및 결과 해석

4.1 자료의 분석

본 연구에서 조사된 자료의 분석을 위하여 SPSS Statistics 17.0 통계 패키지 프로그램을 사용하였다. 응답자들의 인구통계학적 특성을 알아보기 위하여 조사된 자료를 바탕으로 빈도분석을 실시하였다. 본 연구의 설문에 응답한 응답자의 인구통계학적 특성은 <표 2>와 같다.

본 연구에서 조사된 11개의 프로그래밍언어에 대한 숙련도 조사결과는 <표 3>과 같다. 일곱 개(Java, C, C++, C#, PHP, Visual Basic, JavaScript)의 언어에 대해서는 <표 3>과 같이 응답자의 50% 이상이 초급 이상의 숙련정도를 나타내었고, 나머지 4개(Objective-C, Python, Perl, Ruby)에 대해서는 사용한 경험이 없다고 응답한 비율이 75%이상을 차지하였고, 이러한 응답은 다른 언어들과의 상대적인 거리를 측정하는데 있어서 의미 있는 결과를 도출하기 어려우므로 분석 대상에서 제외하였다. 따라서, 본 연구는 50% 이상의

응답자들이 초급이상의 숙련도를 갖고 있다고 응답한 일곱 개의 언어들에 대해서만 다차원척도법을 활용하여 상호 간의 거리 측정을 실시하였다.

<표 2> 응답자의 인구통계학적 특성

	구분	빈도	비율 (%)
성별	남자	125	90.6
	여자	13	9.4
	소계	138	100.0
나이	20 ~ 25세	12	8.7
	26 ~ 30세	56	40.6
	31 ~ 35세	39	28.2
	36 ~ 40세	23	16.7
	41 ~ 45세	7	5.1
	46 ~ 48세	1	0.7
	소계	138	100.0
업무 경력	1 ~ 3년	47	34.1
	4 ~ 6년	40	29.0
	7 ~ 10년	29	21.0
	11년 이상	22	15.9
	소계	138	100.0
교육 수준	고졸	4	2.9
	전문대졸	31	22.5
	대졸	90	65.2
	대학원졸	13	9.4
	소계	138	100.0
직업	응용소프트웨어개발자	35	25.4
	컴퓨터소프트웨어 개발관리자	26	18.8
	웹프로그래머	17	12.3
	게임프로그래머	15	10.9
	시스템소프트웨어 프로그래머	11	8.0
	기타	34	24.6
	소계	138	100.0

<표 3> 응답자의 프로그래밍에 대한 숙련도

프로그래밍 언어		사용 경험 없음	초급	중급	고급	합계
Java	응답자 수	31	60	33	14	138
	비율(%)	22.5	43.5	23.9	10.1	100
C	응답자 수	10	40	58	30	138
	비율(%)	7.3	29.0	42.0	21.7	100
C++	응답자 수	15	45	51	27	138
	비율(%)	10.9	32.6	37.0	19.5	100
C#	응답자 수	50	66	19	3	138
	비율(%)	36.2	47.8	13.8	2.2	100
PHP	응답자 수	69	44	15	10	138
	비율(%)	50.0	31.9	10.9	7.2	100
Objective-C	응답자 수	107	24	6	1	138
	비율(%)	77.5	17.5	4.3	0.7	100
Visual Basic	응답자 수	28	71	34	5	138
	비율(%)	20.3	51.5	24.6	3.6	100
Python	응답자 수	114	21	1	2	138
	비율(%)	82.6	15.3	0.7	1.4	100
Perl	응답자 수	105	31	2	0	138
	비율(%)	76.1	22.5	1.4	0.0	100
Java Script	응답자 수	36	55	30	17	138
	비율(%)	26.1	39.9	21.7	12.3	100
Ruby	응답자 수	122	15	1	0	138
	비율(%)	88.4	10.9	0.7	0.0	100

4.2 연구결과 해석

본 연구는 조사된 자료의 분석을 위하여 SPSS Statistics 17.0 통계 패키지 프로그램을 사용하였다. 분석 방법으로 7개 언어들 간의 유사성을 측정하기 위하여 ALSCAL 분석방법을 사용하였다. 응답내용을 바탕으로 데이터를 구성하는 행렬의 각 셀(cell)들은 데이터의 행과 열을 구성하는 요소들 각 쌍 간의 유사성 정도를 나타내는 데이터 자체가 거리행렬이므로 거리는 정방대칭행의 데이터로부터 거리를 측정하고자 하였다. 다차원척도법 분석에서 제시된 언어들 간의 지각된 유사성 정도를 간격척도로 측정하였으므로 측정수준을 '구간척도'로 설정하였고, 척도화 모형에서 입력 데이터 행렬 간에 가중치를 부여하지 않는 '유클리디언 거리'를 선택하고, 숫자들을 거리 행렬 내에서 비교하는 '행렬' 조건으로 2차원 분석을 실시하였다.

계산을 반복함에 따라 스트레스 값이 작아지면서 개선되는데, 결과의 적합도를 증가시키기 위해 반복에 따른 스트레스 개선 값이 0.0001 보다 작을 때 반복과정을 중지하도록 설정하였고, 많은 반복을 통해 결과

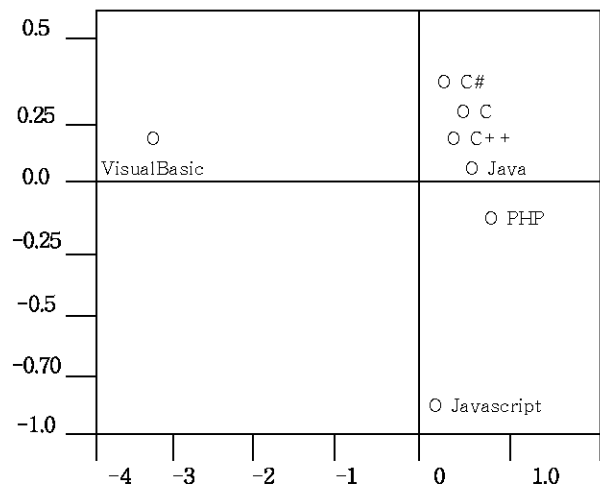
의 적합도를 증가시키기 위해서 최소 S-스트레스 수렴기준을 0.001로 설정하였다.

스트레스 값은 다차원척도법 모형에 의해서 설명되지 않는 분산의 불일치 정도로서 대상들간의 실제거리와 추정된 거리사이의 오차를 의미한다. 다차원척도법의 계산은 스트레스를 최소화하기 위한 최적화(최소 제곱 단조회기 ; least squares monotone regression - 가우스, 뉴턴 등) 알고리즘을 사용하여 얻어진다. 스트레스는 0과 1 사이의 값을 가지며 추정거리와 실제거리가 완전히 일치하면 0이 된다. 그러므로 스트레스 값이 작을수록 추정거리의 적합도는 높으며, Kruskal 이 적합도에 대하여 제시한 기준은 다음과 같다.

다차원척도법에 의한 분석을 실시한 결과, 스트레스 값은 0.03410 으로 모형의 적합도가 매우 좋은 것으로 나왔고, 회귀분석에서의 R Square와 유사한 개념으로서 다차원척도법으로 변환된 자료가 전체 분산을 설명하는 정도를 나타내는 결정계수(RSQ) 또한 0.99851 을 나타내고 있어 모형의 설명력이 매우 우수한 것으로 나타났다.

<표 4> Kruskal의 스트레스 값

스트레스 값	적합도 평가
0.2 이상	매우 나쁘다
0.2	나쁘다
0.1	보통이다
0.05	좋다
0.025	매우 좋다
0	완벽하다

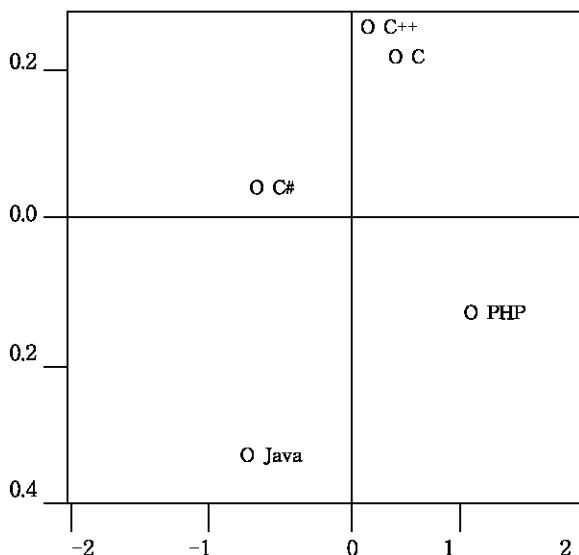


<그림 1> 전체 유클리디언 모형

그림 1은 2차원 상에 7개의 프로그래밍 언어들 간의 거리를 나타내고 있다. 설문에 응답한 프로그래머들은 Visual Basic은 다른 6개의 언어들로부터 거리가 매우 멀다고 인식하고 있는 것으로 나타났다. 이것은 Visual Basic이 문법과 실행방법에 있어서 다른 6개의 언어들과는 확연하게 다른 구조를 가지고 있기 때문인 것으로 판단된다. 그리고, Javascript도 다른 언어들과는 거리가 먼 것으로 나타났는데, 이것은 Javascript는 웹상에서 직접 수행이 가능하여 동적인 웹페이지를 작성하는데 주로 사용되고, 다른 언어에 비해 웹브라우저에서의 편리성과 유용성이 강한 언어라는데 있어서 차이가 있기 때문인 것으로 여겨진다.

그러나, 웹에서 활용되는 언어라는 점에서 비슷한 목적을 가지고 있는 PHP의 경우에는 Java에 더 가까운 언어로 인식되고 있다. 이것은 Javascript가 클라이언트에 설치된 웹브라우저에서 실행되는 반면에, PHP는 서버에서 실행되기 때문인 것으로 여겨진다. 즉, PHP는 서버에서 실행되고 웹브라우저에서는 그 결과만을 보여주는 반면에, Javascript는 그 스크립트가 들어있는 웹 문서를 읽는 브라우저에서 실행기 때문에, 브라우저에서 Javascript 기능이 없거나 꺼두었다면 자바스크립트는 실행되지 않는다.

서로 비슷해 보이는 5개의 언어들(C, C#, C++, PHP, Java)의 상대적인 위치를 좀더 정확하게 이해하



<그림 2> 5개 언어들에 대한 유클리디안 모형

기 위하여 이들 다섯 개 언어들에 대해서만 다차원척도법에 의한 분석을 실시하였다. 그 결과, 스트레스 값은 0.0218로서 모형의 적합도가 역시 좋았고, 결정계수(RSQ) 또한 0.99843으로 설명력이 매우 우수한 것으로 나타났다. 그리고, 그림 2는 5개 언어들에 대한 상대적인 거리를 보여주고 있다.

그림 2에서는 Visual Basic이나 Javascript와 비교하여 이들 5개의 언어들끼리 서로 매우 비슷해 보였지만, 그림 1에서 서로 가깝게 위치해서 비슷해 보이는 언어들끼리 그림 2에서는 상대적인 거리가 멀게 인지되어 서로 다르게 인식되고 있다는 것을 알 수 있다. 즉, 설문에 응한 프로그래머들이 C와 C++, C#, 그리고 Java 간에도 차이가 있다는 것으로 인식하고 있다는 것이다. 흥미로운 것은 C++와 Java가 모두 객체지향이고, C는 절차지향적인 언어임에도 불구하고, C와 C++ 간의 거리 보다 Java와 C++ 간의 거리가 더 크다고 인식하는 것으로 나타났다는 것이다. 이것은 C++가 C언어의 철학과 개념을 많이 계승했기 때문으로 여겨진다. C#은 C++와 Java의 중간에 위치하는 것으로 나타났는데, 이것은 상속(Inheritance), 인터페이스, 그리고 메모리관리에 있어서 Java의 특성을 대부분 수용하면서도 포인터와 같은 C, C++의 개념도 부분적으로 가지고 있기 때문인 것으로 보인다.

프로그래밍 언어의 지속적인 발전으로 언어의 성격과 문법구조 등이 조금씩 변하고 있지만 응답자들을 분석해본 결과 변수, 함수의 사용과 같이 언어들마다 특성보다 근본적인 문법 구조나 사용목적에 따라 프로그래밍 언어의 유사정도를 인지하는 것으로 분석되었다.

C언어와 PHP는 프로그래밍 용도와 문법의 차이가 있지만 PHP가 개발될 당시 C언어의 영향을 받아 기본 문법스타일은 유지되고, 보다 발전되어 성격이 다른 스크립트언어로 개발되었기 때문에 C와 PHP의 거리가 Javascript 보다 상대적으로 가깝게 나타났고, C#과 Java 역시 동작 플랫폼과 같이 약간의 차이가 존재하지만 문법적인 특성이 아주 유사하기 때문에 C++보다 가깝게 느끼는 것으로 여겨진다.

5. 결 론

본 연구에서 조사한 7개(Java, C, PHP, C++, Visual Basic, C#, JavaScript) 프로그래밍 언어들 간에 프로그래머가 지각하고 있는 인지적 거리를 측정하였다. 그 결과, 언어의 문법적 유사성과 객체지향, 명령형 등 언어의 성격에 따라 각각 유사하거나 다르게 나타났다.

일반적으로 1명의 프로그래머가 여러 가지의 언어를 이용하지 않아 11개의 언어를 대상으로 조사를 했을 때 만족하지 못하는 결과값이 도출되었지만, 대부분의 응답자들의 응답 성실도가 높은 6개(C, C++, C#, Java, PHP, Visual Basic)의 프로그래밍 언어를 대상으로 분석을 한 결과 각 언어들 간의 문법적 특성, 구조적 특성에 따라 각각 유사하게 인지하는 것으로 분석되었다. 즉, 2차원의 좌표상 도출된 개체들간에 거리가 가까울수록 문법적, 구조적 특성이 유사하기 때문에 학습하기 용이하고, 좌표상 개체간의 거리가 멀어질수록 문법적 특성과 구조적 특성이 상이하어 학습에 있어 많은 시간과 노력이 요구된다.

본 연구에 있어서 한계점은 프로그래밍 언어와 관련하여 다차원척도법 연구가 많이 부족하여 선행연구에 어려움이 있었으며, 표본 확보에 있어 설문조사의 인구통계학적 분석에서도 알 수 있듯이 응답자의 대부분이 초급, 중급 프로그래머에 편재되었다는 것이다. 이 외에도 본 연구 대상과 측정 방법 등에서 다음과 같은 제한점을 지니고 있다. 본 연구는 대부분의 설문 응답을 개발자가 자주 이용하는 특정 목적의 사이트에 설문지를 등록하여 사이트의 이용자들에게 제한되어 설문조사가 실시되었으므로 연구대상의 대표성과 관련된 문제가 제기될 수 있다.

향후 지속적인 연구를 통하여 표본수를 늘리고, 경험이 풍부한 고급프로그래머들을 표본으로 조사한다면, 더욱 의미 있는 결과를 도출할 수 있을 것으로 기대된다. 특히, 숙련도가 높은 응답자가 부족해서 다차원분석을 실시할 수 없었던 나머지 4개의 언어들(Python, Perl, Ruby, Delphi)에 대해서도 경험이 풍부한 프로그래머를 설문응답에 참여시킨다면, 많이 활용되는 프로그래밍 언어들에 대한 전반적인 이해를 하는데 도움을 줄 수 있을 것으로 기대된다. 또한, 유사

프로그래밍언어의 학습에 있어서의 효율성 측정을 통해 학습 용이한 교육과정에 관한 연구와 프로그래밍 언어 인지에 대한 국가별, 경력별 분석을 통하여 포지셔닝의 차이점에 대한 연구가 필요할 것이다.

참 고 문 헌

- [1] 고석하, "IS 실무자의 지식 및 기술 필요: IS 실무자, 채용 담당자, 교수, 학생 간의 인식 차에 대한 비교 연구", *Information Systems Review*, 제12권, 2호, pp.205-223, 2005.
- [2] 권남훈, 이인찬, 강순희, 김성현, 전병유, 금재호, 김종일, 최강식, 이상오, 오정숙, 김기현, 정윤형, "정보통신 인력의 특성, 수급실태 및 전망", 정보통신정책연구원, 2001.
- [3] 김인재, "객체지향 컴퓨팅 채택에 미치는 영향요인: 프로그래밍 경험과 개인혁신성의 경우", *정보시스템연구*, 제10권, 1호, pp.243-255, 2001.
- [4] 남은희, "컨벤션 개척지와 컨벤션 시설 속성의 MDS 분석", *동국대학교 석사학위논문*, 2003.
- [5] 박성준, 김성훈, "선호별 자동차 네비게이션시스템의 감성평가", *한국산업정보학회논문지*, 제9권, 3호, pp.71-79, 2004.
- [6] 유상진, 이유진, "IT인력의 경력변경의도 영향요인 분석: KSA와 자아효능감을 중심으로", *한국산업정보학회논문지*, 제14권, 5호, pp.197-209, 2010.
- [7] 이재창, 박정섭, "다차원척도측척기법", *응용통계*, 제1권 1호, pp.61-80, 1986.
- [8] 이찬, 정진철, 박인섭, 임규건, 신재호, 김태연, "IT인력에 대한 산업체의 교육 실태 및 요구 분석", *농업교육과 인적자원개발*, 제41권, 1호, pp.161-189, 2009.
- [9] 이훈영(2004). e-마케팅 플러스. 무역경영사.
- [10] J. Bailey and R. B. Mitchell, "Industry perceptions of the competencies needed by computer programmers: technical, Business, and soft skills", *Journal of Computer Information Systems*, vol. 47, no. 2, pp.28-33, 2006/2007.
- [11] A. C. Benander, B. A., Benander, and M. Lin,

- "Perceptions of Java -- experienced programmers' perspective." *Journal of Computer Information Systems*, vol. 43, no. 4, pp.1-7, 2003.
- [12] R. D. Boyle, J. Carter, and M. Clark, "What makes them succeed? Entry, progression, and graduation in computer sciences". *Journal of Further and Higher Education*, vol. 26, pp.3-18, 2002.
- [13] M. Campione, K. Walrath, and A. Huml *The Java tutorial* (3rd edition). Boston, MA: Addison-Wesley, 2000.
- [14] L. Chen, M. N. Frolick, and A. Muthitacharoen, "Investigating the use of role play training to improve the communication skills of IS professionals", *Journal of Computer Information Systems*, vol. 43, no. 3, pp.67 - 74, 2003.
- [15] P. H. Cheney and N. R. Lyons, "Information systems skill requirements: a survey", *MIS Quarterly*, vol. 4, no. 1, pp.35 - 43, 1980.
- [16] R. Dattero, J. Quan, and S. Galup, "Estimating the value of Java and C++ skills", *Communications of Association for Information Systems*, vol. 11, pp.310-321, 2003.
- [17] D. C. Davis, "Job titles, tasks, and experiences of information systems and technologies graduates from a midwestern university", *Journal of Information Systems Education*, vol. 14, no. 1, pp.59-68. 2003.
- [18] J. Gallant, "Fourth-generation language use must be planned, report says", *Computer World*, I&I8, pp.64ff, Apr. 30. 1984.
- [19] M. H. Ginzberg, and J. J. Baroudi. "Impact of the Technological Environment on Programmer/Analyst Job Outcomes". *Communications of the ACM*, vol. 29, no. 6. pp.546 - 555, 1986.
- [20] A. Goold, and R. Rimmer, "Factors affecting performance in first-year computing", *SIGCSE Bulletin*, vol. 32, pp.39-43, 2000.
- [21] M. L. Markus and R. Benjamin, "Change agency - the next IS frontier", *MIS Quarterly*, vol. 20, no. 4, pp.385 - 407, 1996.
- [22] K. M. Nelson and J. G. Coopridge, "The contribution of shared knowledge to IS group performance", *MIS Quarterly*, vol. 20, no. 4, pp.409 - 429. 1996.
- [23] C. L. Noll and M. Wilkins, "Critical skills of IS professionals: a model for curriculum development", *Journal of Information Technology Education*, vol. 1, no. 3, pp.143 - 154, 2002.
- [24] TIOBE Software. *TIOBE programming community index for January 2007*. Retrieved September 31, 2007, from the TIOBE Software website: http://www.tiobe.com/index.htm?tiobe_index. September 2011.
- [25] P. A. Todd, J. D. McKeen, and R. B. Gallupe, "The evolution of IS job skills: a content analysis of IS job advertisements from 1970 to 1990", *MIS Quarterly*, vol. 19, no. 1, pp.1 - 24, 1995.



이 충 권 (Choong Kwon Lee)

- 미국 University of Nebraska-Lincoln에서 MIS전공으로 Ph. D. 취득
- 현재 계명대학교 경영정보학과에서 부교수로 재직

• 주요관심분야: IT Job Skills



유 상 진 (Sangjin Yoo)

- 미국 University of Nebraska-Lincoln에서 MIS전공으로 Ph. D. 취득
- 현재 계명대학교 경영정보학과에서 교수로 재직

• 주요관심분야: IS/IT 전략



진 상 민 (Sangmin Jin)

- 계명대학교 MIS전공으로 석사 취득
- 현재 대구일보 전산실 근무
- 주요관심분야: 정보시스템 설계 및 구현

논문 접수 일: 2012년 01월 02일

1차수 정완료일: 2012년 02월 01일

게재 확정 일: 2012년 02월 06일