A Snapshot of Current Languages Used in Industry

Jennifer Polack-Wahl, Stephen Davies and Karen Anewalt
University of Mary Washington, japwahl@gmail.com, stephen@umw.edu, anewalt@umw.edu

Abstract— **We provide results from a nationwide survey of advertisements for jobs in the technology sector. This snapshot of 521 job postings provides an interesting glimpse into the state of the computing industry in the U.S., quantifying the programming languages most frequently requested by employers today. This study reveals industry preference for Java and C++ skills. C++ is requested most frequently in the South Atlantic Region and was also favored on the west coast. Additionally, SQL skills were more requested for positions related to testing, JSP skills were more requested in architect positions, and both C and C++ were more requested in analyst positions. While academic practices should not be based solely on industry practice, industry demand for languages serves as one useful data point when institutions make pedagogical choices.**

Keywords-computer science education; languages; curriculum; industry

I. Introduction

In 2010, the first U.S. national survey of programming languages and practices used in introductory Computer Science courses in colleges and universities was conducted [1]. The survey showed that Java was the most frequently taught language in both CS1 and CS2 courses. One reason often cited by educators for languages choice in a particular course is industry demand [3,4] but few resources exist to quantify the industry demand. Our study provides a current data point about language popularity and also provides more specifics about the correlation of language popularity to other variables such as geographic region and job classification.

Trends in language use nationally as well as regionally are examined. Additionally we report on trends seen for various job categories. While academic practices should not be based solely on industry practice, industry use of languages does provide an important consideration for institutions to consider when making pedagogical choices. Comparing the data based on geographic regions and job categories can potentially allow faculty to consider the data in a more meaningful way for their institution and the students that they serve.

II. Background

Several past studies have reported on language prevalence in industry. In 2002, a national survey of languages used by industry in Australia was conducted. Educators sampled employment advertisements in the IT section of The Australian newspaper and recorded the languages mentioned for programmer positions. It was found that the average advertisement required 1.84 languages, that 48% of jobs required more than one language, and the most popular languages were C++ and Java (30% each)[2]. TIOBE Software conducts ongoing surveys of programming language popularity by tracking the number of hits when querying popular web sites such as Google, Blogger, Wikipedia, YouTube, Yahoo!, Bing, and Baidu. The results as of August 2011 show Java as the most widely used language (19%), followed by C (17%) and C++ (8%)[5]. As reported by TIOBE, these three languages have maintained the same popularity for the last 12 months.

These previous works provide interesting data about language choice, but fail to correlate the languages required to common programming jobs and also do not report on regional differences in language prevalence within the U.S. This study provides a current data point about language popularity and also provides more specifics about the correlation of language popularity to other variables such as geographic region and job title.

III. Procedure and Methods

We gathered computer science related job advertisements from online newspapers for major U.S. cities from the various geographic regions of the country, specifically: Silicon Valley, California; Seattle, Washington; District of Columbia; Denver, Colorado; New York City, New York and Detroit, Michigan. We also included other major newspapers and geographically dependent publications to get a balance of geographic regions.

The data was gathered every fourth day from January 29, 2011 until August 8, 2011. We gathered fifteen different job advertisements each day until April 2011, half using the keyword “programmer” and the other half using the keywords “computer science”. Additional data was collected during the summer months. Jobs that had no reference to requested language or experience in language were discarded, otherwise, every third entry was manually parsed into the following; date, keyword used, language, job title, location and URL. If the job was already entered with the alternative keyword (programmer or computer science), the job was discarded and the next job in the list was entered.

After collecting and parsing the data set we categorized the jobs by the regions and divisions listed in each job description. We used the US Census Regions and Divisions as shown in Table 1 [6].

TABLE I. U.S. CENSUS DEFINED GEOGRAPHIC REGIONS

US Census Regions			
Region 1: Northeast	Region 2:Midwest	Region 3:South	Region 4:West
104 jobs	100 jobs	162 jobs	155 jobs
Division 1: New England (44 jobs)	Division 3: East North Central (57 jobs)	Division 5: South Atlantic (83 jobs)	Division 8: Mountain (57 jobs)
Division 2: Mid-Atlantic (60 jobs)	Division 4: West North Central (43 jobs)	Division 6: East South Central (36 jobs)	Division 9: Pacific (98 jobs)
		Division 7: West South Central (43 jobs)	

IV. RESULTS

We collected data on a total of 521 job postings over an eight-month period. These postings appeared in a variety of sources, from geographically dependent publications (e.g., Maine Today) to national job boards (e.g.,dice.com.) The number of ads represented from each source is depicted in Table 2.

TABLE II. SOURCES FOR JOB ADS

Source	Number of Postings Used
Washington Post	67
Dice.com	58
Nytimes.monster.com	56
Denver Post	55
SJ Mercury News	54
Detroit Free Press	47
Seattle Times	45
Minnesota Star	24
Boston.com	23
Job hunt	23
Tennessean	17
News OK	16
Careerbuilder.com	14
Maine Today	12
Lexington Herald-Leader	10

For simplicity, we use the most conservative "universal" margin of error in computing the confidence interval for all languages, using 0.5 as the sample proportion $\hat{p}$. We assume an infinite population, and use 521 for the sample size, which gives us a (conservative) margin of error of

$$1.98\, x \sqrt{\frac{\hat{p}\, x\, (1-\hat{p})}{n}} \pm 4.29\%. \tag{1}$$

A. *Overall Language Prevalence*

Figure 1 depicts the overall prevalence of the most popular languages and technologies, in terms of the total percentage of all sample job ads containing that language. Java was mentioned most often, but this is (just) within the margin of error, and so we consider Java, C++, SQL, and C# in a virtual tie. C, presumably used mostly in operating systems and embedded applications, still has a very strong showing despite its age. Note that many of the Web-only languages and technologies (PHP, ASP, Ajax, etc.) are much less frequently requested overall than the mainstream application development languages, although the number of different choices here makes the overall Web contribution still quite significant. Python, despite its popularity in academic circles, is still far from mainstream in industry.

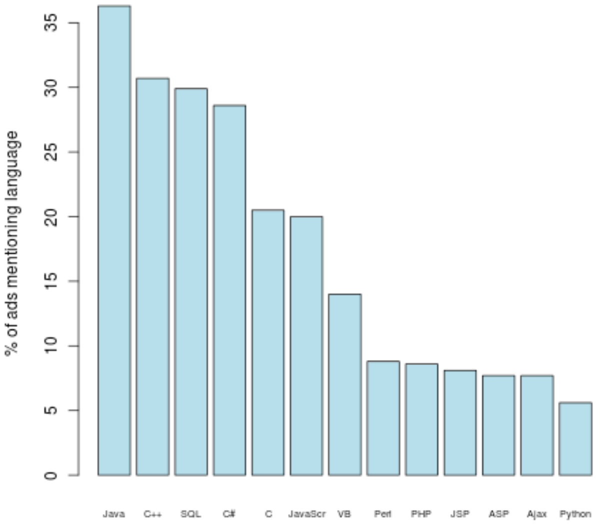

Figure 1. Percentage of all sample job ads that mentioned certain languages. (Only languages that appeared in at least 20 of 521 job ads are shown.) Note that most job ads mention more than one language, so the total is much greater than 100%.

B. *Results by Job Title*

We attempted to gain insight into how language use varies among workers with particular job functions. A typical software development effort involves not only programmers but also architects, analysts, testing and QA, and support staff. How do the languages advertised correlate with these sets of responsibilities?

It is difficult to get a precise measure because postings are highly variable in the language they use to describe job positions. One company advertises for a "software developer" and another for a "programmer." Are these essentially the same? Does "software engineer" imply programming, or is it a broader term encompassing other aspects of the development process? "Programmer/analyst" is a common catchphrase; so are "programmer" and "analyst" still different descriptions?

Although the data is thus a bit dirty, we still believe there is value in ferreting out some of these subtleties. We defined five broad categories of job positions -- architect, analyst, programmer, tester, and miscellaneous -- and used the following decision procedure to classify each ad:

1. If the word "test" or "tester" appeared in the job title, it was classified as a **tester** position (even if other indicative words appeared; see below.)
2. If "development," "developer," or "programmer” were in the title, it was classified as **programmer**. Note that this is true even for the numerous "programmer/analyst" descriptors, since we reasoned that such positions would involve writing substantial project code.
3. If "analyst," "engineer," or "researcher" were in the title, it was classified as **analyst** (except as indicated above.)
4. If "architect" or "designer" were in the title, it was classified as **architect**.
5. Finally, if none of the above rules applied, the ad was classified as **miscellaneous**.

The totals for each job classification are presented in Table 3.

TABLE III. NUMBER OF ADS CLASSIFIED IN VARIOUS BROAD JOB CATEGORIES.

Job Classification	Number	Percentage
Architect	41	7.9%
Analyst	120	23.0%
Programmer	288	55.3%
Tester	46	8.8%
Miscellaneous	26	5.0%

To detect correlations between job classifications and language prevalence, we used a χ^2 test for each popular language with α set conservatively to .01 (instead of .05) to compensate for the total number of tests being so large. Even with α so low, there were still a large number of languages whose null hypothesis was rejected. To cross-check and further refine the results, we examined the job classification's standardized (adjusted) Pearson residuals for each language whose χ^2 test was positive. Only if at least one residual had an absolute value greater than 2.5 (meaning that its observed value was more than 2.5 standard deviations away from its expected value) do we include it in the list of significant results as shown in Table 4.

TABLE IV. SIGNIFICANT CORRELATIONS BETWEEN LANGAUGE PREVALENCE AND JOB CLASSIFICATION. RESULTS ARE SHOWN IN THE TABLE ONLY FOR LANGUAGE/CLASSIFICATION PAIRS WHOSE χ^2 TEST HAD $P < .01$ AND A STANDARDARIZED (ADJUSTED) PEARSON RESIDUAL WHOSE ABSOLUTE VALUE WAS GREATER THAN 2.5.

Language	More prevalent among	Less prevalent among	Overall Percentage
C	Analysts (29.2%) Misc (42.3%)	Programmers (16.3%)	20.5%

Language	More prevalent among	Less prevalent among	Overall Percentage
C++	Analyst (47.5%)	Testers (10.9%)	30.7%
VB	Programmers (19.8%)		14.0%
JSP	Architects (31.7%)		8.1%
SQL	Testers (45.7%)	Analysts (19.2%)	30.0%
PL/SQL	Misc (11.5%)		2.1%
Ruby	Misc (15.4%)	Programmers (0.7%)	2.5%

Several tentative observations can be surmised from these results. SQL is prominently demanded for tester positions, perhaps indicating that managing a suite of test cases, and implementing scenarios to test an application's edge cases, frequently involve database skills in today's software processes. JSP appears far more often in ads for architects/designers than anything else, which may be a signal that the majority of "programming" work in a typical J2EE Web application is actually done in Java libraries rather than servlets themselves, which are the realm of architects and designers. Ruby and PL/SQL, two languages that appear very rarely in the overall data set, are actually fairly frequent among miscellaneous ads. This may indicate that these languages are rarely used in product implementation, but can be important tools employed by "utility players" on the support staff.

Interestingly, the term "programmer" appears to be used more often to describe (perhaps) developers of high-level end user applications, as evidenced by the prevalence of Visual Basic requested for those job titles. "Analyst," on the other hand, correlates more often with C and C++ development, and less with database interfaces, perhaps indicating lower-level infrastructure.

C. Results by Geographic Region

Our sample job postings obviously contained information about the physical location of the job opportunities, so it is interesting to explore whether there are overall geographic trends. Following the above reasoning, we again used a χ^2 test with α=0.01 for each language, and identified standardized Pearson residuals greater than 2.5 for specific region or division.

The results reveal some dramatic differences between different parts of the country. For one, the South Atlantic region (stretching from Delaware down to Florida) is a hotbed of C/C++ development, far more than most other areas of the country. This area features the Research Triangle Park in North Carolina, the Florida High Tech Corridor, and Virginia's strong Department of Defense presence. It is possible that sites like these have influenced projects to adopt proven, reliable, natively compiled, and fast implementation languages. C++ is also very strong on the west coast, which may be surprising to those who associate that region with Web applications and rapid prototyping. By contrast, the C language -- while still retaining an important niche throughout most of the country -- has nearly disappeared from the northeast region, especially New England where it is largely absent.

The most atypical region overall is the midwest, and particularly the East North Central division of Illinois, Indiana, Michigan, Ohio, and Wisconsin. This area has been hit as hard as any other by the recent economic downturn, and hence its economy is distinctive in many ways. Perhaps this has had some influence in driving high-tech project decisions away from the national norm. In any event, this region is currently strongly focused on Microsoft environments like VB and ASP, and only rarely searches for C++ developers. The prevalence of Visual Basic is also very noticeable in the neighboring East South Central region of Alabama, Kentucky, Tennessee, and Mississippi.

Note that Java, the programming language most heavily sought after overall, did not appear in any of the correlations in this section or the previous one. It is strong and well-represented in all sectors, and can almost be considered the lingua franca of the high-tech world.

TABLE V. SIGNIFICANT CORRELATIONS BETWEEN LANGUAGE PREVALANCE AND GEOGRAPHIC REGION. RESULTS ARE SHOWN IN THE TALBE ONLY FOR LANGUAGE/REGION OR LANGUAGE/DIVISION PAIRS WHOSE χ^2 TEST HAD P<.01 AND A STANDARDIZED (ADJUSTED) PEARSON RESIDUAL WHOSE ABSOLUTE VALUE WAS GREATER THAN 2.5.

Language	More prevalent among	Less prevalent among	Overall Percent age
C	South Atlantic (31.3%)	NE (7.7%), especially New England (4.5%)	20.5%
C++	West (41.9%) Especially Pacific (46.9%) also South Atlantic (45.8%)	Midwest (16.0%) especially East North Central (12.3%)	30.7%

Language	More prevalent among	Less prevalent among	Overall Percent age
VB	East North Central (31.6%) and East South Central (38.9%)		14.0%
ASP	East North Central (17.5%)		7.7%

D. *Co-occurrence of languages*

Finally, it is of interest to consider which languages often appear together in the same job ad. To do this, we use the Phi coefficient φ to measure statistical significance of co-occurrence. $\varphi \times \sqrt{N}$ gives a distribution that is approximately standard normal, in which positive values indicate positive correlations. It is a more accurate measure of correlation than a simple count of co-occurrences would be since it adjusts for the overall frequency of appearance of each language by itself. Following our previous benchmark, we consider values whose $\varphi \times \sqrt{N} \geq 2.5$ (that is, more than 2.5 standard deviations from its expected value) to be significant. The results are shown in Table 6.

As can be seen, C and C++ are by far the most tightly correlated languages, appearing together in 87 ads, which works out to a phi statistic over 12 standard deviations higher than the expected mean. Many of these co-occurring languages are not surprising (for instance, JavaScript is a necessary component of Ajax, and Java of JSP, while VB and C# are two of the most common CLI languages used in ASP.)

One bit of insight can be gleaned by observing that Perl frequently occurs with a number of different languages, probably indicating that it serves a utility role in many development teams, complementing a main implementation language. It is also perhaps surprising that C++ and Java co-occur so frequently, since they are often seen as implementation alternatives. On the other hand, the pairing may simply represent an industry preference for object-oriented language experience, rather than a specific language. The same could be said for C++ and Python. This could be an indicator of multi-language projects that include both a low-level, performance-driven component in addition to a user interface that benefits from a virtual machine's features. Clearly, there are numerous job opportunities that require developers to have experience with both.

TABLE VI. SIGNIFICANT CO-OCCURRENCES OF LANGUAGES. THE “FREQUENCE” COLUMN GIVES THE TOTAL NUMBER OF TIMES THE TWO LANGUAGES WERE MENTIONED IN THE SAME JOB AD (OUT OF 521 ADS), AND THE “$\varphi \times \sqrt{N} \geq 2.5$” COLUMN GIVES AN APPROXIMATE STANDARD NORMALLY DISTRIBUTED STATISTIC INDICATING THE STRENGTH OF THE CO-OCCURRENCE. LANGUAGE PAIRS WHOSE PHI STATISTIC IS GREATER THAN 2.5 ARE SHOWN.

Language 1	Language 2	Frequency	$\varphi \times \sqrt{N}$
C	C++	87	12.78
JavaScript	Ajax	29	8.67
Python	Perl	15	8.45
Java	JSP	36	6.85
VB	ASP	20	6.85
VB	C#	44	6.39
PHP	JavaScript	25	6.16
ASP	C#	27	5.71
SQL	PHP	28	5.02
PHP	Ajax	11	4.34
Java	Perl	28	3.65
C++	Java	74	3.20
C++	Python	16	2.97
C++	Perl	23	2.97
C	Perl	16	2.51

V. CONCLUSION

As many institutions report that popularity of a language in industry is one item taken into consideration when a language is selected for classroom instruction, it is interesting and useful to compare the industry results to those of the previously conducted national survey of languages used in introductory programming courses [1]. Table 7 compares the top 4 languages mentioned in job ads to the same languages' reported use in CS1 ad CS2. In both studies, languages could be selected multiple times, *i.e.* an institution may have indicated that some sections of CS1 used Java and some sections used C++ and similarly a job ad may have mentioned multiple languages. As a result the percentages do not sum to 100% in either study.

TABLE VII. COMPARISION OF LANGUAGE PREVALENCE IN INDUSTRY AND EDUCATION.

Language	Industry job ads mentioning	CS1	CS2
Java	36.3%	48.2%	55.8%
C++	30.7%	28.8%	36.1%

Language	Industry job ads mentioning	CS1	CS2
SQL	29.9%	N/A	N/A
C#	28.6%	1.9%	1.6%

As can be seen, C# is often requested in industry job ads but seldom taught in CS1 and CS2 courses. Java and C++ are frequently requested by industry and frequently taught in CS1 and CS2. SQL is rarely if ever taught in introductory programming courses. The previous study referenced did not collect data about languages used in upper level or elective courses, so it unclear whether students typically receive experience in SQL as part of their undergraduate education.
Given the relative frequency in which job ads mention multiple languages, it is interesting to observe that in the previous study approximately 50% of institutions reported teaching the same language in CS1 and CS2. Additionally we observe that 63% of CS1 and CS2 courses used only object-oriented languages, while in industry, languages supporting a range of paradigms are used. Again, it is possible that institutions are teaching these alternate paradigms in upper level courses, which were not included in the previous survey [1].

As a final note, we remark that it may be beneficial to the computer science education community to develop a practice of collecting data about language use in industry and education, as this can provide a practical consideration as we develop curriculum for the future.

ACKNOWLEDGMENT

We are grateful to Dr. Debra Hydorn for invaluable advice on statistical analysis and methods.

REFERENCES

[1] Davies, S., Polack-Wahl, J., Anewalt, K., A Snapshot of Current Practices in Teaching the Introductory Programming Sequence. *SigCSE, '11 Proceedings of the 42nd ACM technical symposium on Computer Science Education* (2011), 625-630.

[2] de Raadt, M., Watson, R., and Toleman, M. 2002. Language Tug-Of-War: Industry Demand and Academic Choice. *ACE'03 Proceedings of the fifth Australasian Conference on Computer Science Education.* 20 (2003), 795-825.

[3] Dingle, A. and Zander, C. 2001. Assessing the Ripple Effect of CS1 Language Choice. *Journal of Computing Sciences in Colleges*. 16, 2 (May 2001), 85-94.

[4] Hind, M. 2008. Addressing the Disconnect Between the Good and the Popular. In *ACM SigPLAN Notices.* 43, 11 (Nov 2008). ACM, 74-76.

[5] TIOBE Software. Programming Community Index. 2011. Retrieved August 26, 2011 from http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html.

[6] U.S. Census. Geographic Regions. 2011. Retrieved August 26, 2011 from www.**census**.gov/geo/www/**us**_regdiv.pdf.