

# Analysis of Technical Skills in Job Advertisements Targeted at Software Developers

Sami SURAKKA

*Helsinki University of Technology*  
*P.O. Box 5400, FIN-02015 HUT, Finland*  
*e-mail: sami.surakka@hut.fi*

Received: January 2005

**Abstract.** The present paper has two parts. The first part is a trend analysis from the period 1990–2004 and the second part is a cross-sectional analysis from the year 2004. In both parts, the purpose was to find the most common technical skills sought in American job advertisements for software developer positions. Unlike in previous researches, also distributed technology skills were analyzed thoroughly because as a consequence of World Wide Web technology, these skills are required now more often than ten years ago. According to the trend analysis, the mean of the number of required technical skills increased from 3.6 to 7.7, and the technical requirements have changed as more versatile. The proportion of distributed technology skills increased very strongly: from 0% in 1990 to 65.1% in 2004. According to the cross-sectional analysis, the top five skills sought in 2004 were Windows, Java, C++, SQL, and Unix. In addition, implications to computer science education are considered.

**Key words:** content analysis of job advertisements, need assessment, software developer, technical skills.

## 1. Introduction

In the present research, the goal has been to identify technical skills that are important for software developers' work. The research originates from the need to better understand what topics and skills should be included in the Master's level education of software systems specialization in a technical university. Job advertisements were analyzed to find out the most common technical skills sought in various software developer positions. The present research is a part of my ongoing doctoral thesis work where different research methods have been used to evaluate software systems education (e.g., a Delphi study targeted at experienced software developers and questionnaires targeted at graduating Master's students).

For any reader wishing to get an overview of research into need assessments in the field of information technology, two good starting points are the papers by Nakayama and Sutcliffe (2000; 2001). Most previous relevant researches have been carried out by educators and researchers who work for information systems (IS) or information technology (IT) degree programs, not for computer science (CS) or software engineering (SE) programs. 42 publications that are related to the present research were found. 35 (83.3%)

of them were from the area of IS/IT. The results have been typically published in publications like the MIS Quarterly, Journal of Computer Information Systems, and the proceedings of ACM's Special Interest Group for Computer Personnel Research (SIGCPR, currently merged with SIGMIS). These 42 publications were classified according to the research methods: 22 (52.4%) were content analyses of job advertisements as the present research, 9 (21.4%) were surveys, 4 (9.5%) were literature surveys or research in progress reports, 1 (2.4%) was an interview, and 6 (14.3%) used more than one method.

The following characteristics are original or different to the present research when compared with previous job advertisement analyses: (a) The present research is limited to software developer positions and technical skills are analyzed in a more detailed manner than previously. For example, distributed technology skills were analyzed thoroughly because as a consequence of World Wide Web technology, these skills are required now more often than ten years ago. (b) Advertisements for more scientific or engineering-oriented positions have sometimes been excluded from the previous samples for an obvious reason: these positions are not very suitable for IS/IT graduates. I work at a university of technology and, therefore, included also more scientific or engineering-oriented positions in the sample.

The results of this study might be useful for the training departments of companies, training institutes, and curriculum designers in universities – in particular for those educators who are responsible for software systems specializations. Students might use the results when they are selecting elective computer science courses, especially in industry-oriented master's programs. A software developer working in industry might want to compare his or her skills to the results of the present paper. The results might even reveal something about what technologies were used in new projects if it is assumed that new employees are often hired for such projects. From this viewpoint, the competition between Microsoft's and Sun's technologies for distributed systems (Section 3.3.2) is particularly interesting.

The paper is divided into two parts. First, the trend analysis from the period 1990–2004 is presented in Section 2. Second, the cross-sectional analysis from the year 2004 is presented in Section 3. Both parts include their own subsections of related work, research methods, results, and evaluation. Finally in Section 4, some conclusions are drawn and the implications to education are considered.

The present paper is an extended version of the conference paper (Surakka, 2004). During the conference, trend analysis was suggested as a possible topic of future research. As a consequence of this suggestion, I conducted a trend analysis at the end of the year 2004. Thus, the first part (Section 2) is new when compared with the conference paper. The second part (Section 3) is based on the conference paper.

## **2. Part 1. Trend Analysis**

The two main questions of the present trend analysis were: (1) Has the number of required technical skills increased during the past 15 years? Todd, McKeen, and Gallupe (1995)

reported that the number of technical phrases in job advertisements for programmer positions increased from the mean of 2.2 in 1970 to 4.2 in 1990. Has this increase continued after the year 1990? (2) In particular, how has the number of required distributed technology skills increased? World Wide Web technology was released in 1993. After this, the number of web sites has increased rapidly. As a consequence, skills related to distributed systems are now required more often than ten years ago.

First, related work is presented in Section 2.1. The research method is described in Section 2.2 and Section 2.3 presents the results. In Section 2.4, this trend analysis is evaluated. For brevity, the present paper presents only some general results. More detailed results about individual technical skills such as the most common programming languages can be found in (Surakka, 2005).

### 2.1. *Related Work*

From the 17 job advertisement analyses mentioned in Section 1, seven are longitudinal researches as this part of the present research (Athey and Plotnicki, 1998; Gallivan *et al.*, 2002; Litecky *et al.*, 1996; Maier *et al.*, 1998; Prabhakar *et al.*, 1995; Todd *et al.*, 1995; Trower 1995). However, Litecky *et al.* (1996) and Prabhakar *et al.* (1995) contain results only from two or three subsequent years from the period 1993–1995. In other longitudinal researches the periods were 6–21 years. The periods used by Litecky *et al.* (1996) and Prabhakar *et al.* (1995) are so short that I have classified their researches as cross-sectional researches. The five other longitudinal researches are the most relevant to the present research. Later in Sections 2.2 and 2.3, the comparisons of methodological choices and results are targeted to these.

All these five previous longitudinal researches were targeted at IS/IT field. Next, these researches are shortly presented:

- Athey and Plotnicki (1998) analyzed individual technical skills such as Cobol, Oracle, and Windows, and categories such as mainframes and minicomputers. They reported results from ten different cities as well. The samples were from the years 1989, 1992, 1993, and 1996.
- Gallivan *et al.* (2002) counted, for example, proportions of different job titles and averages of skills per advertisement. The samples were from the years 1988, 1995, and 2001. The coding scheme included also categories for individual technical skills such as Cobol and C, but their paper did not include these results.
- Maier *et al.* (1998) analyzed mostly individual technical skills such as C, Cobol, and Unix. The samples were from the years 1978/79, 1983/84, 1988/89, and 1993/1994.
- Todd *et al.* (1995) analyzed technical, business, and systems skills for programmers, systems analysts, and managers. The samples were collected from the years 1970, 1975, 1980, 1985, and 1990. From these five researches, Todd *et al.* is the most relevant to the present research because it had also detailed results about programmers.
- Trower (1995) analyzed individual technical skills and combined them as categories such as “3GL” (e.g., Cobol, Fortran, or Pascal). The samples come from

the period of 1990–1995 at one-year intervals. His research is particularly interesting from the viewpoint of the present paper because he reported also results about distributed technology skills.

Beside the previous scientific publications, there is one non-scientific report series that is worth mentioning. A British company Salary Services Ltd. has conducted job advertisement analyses for the past 15 years. The data were collected from several British newspapers and web recruiting services. In particular, this report series is interesting because it is problematic to get past data from web recruiting services (see Section 2.2.1). A recent report (Salary Services, 2004a) is a combined longitudinal analysis and cross-sectional analysis as the present research. The data were processed automatically, and the sample was much greater than in the previously mentioned scientific longitudinal researches. From the viewpoint of trend analysis, the interesting part is the ranking list of 150 individual technical skills from the years 1999–2003 with a one-year interval (*ibid.*, pp. 224–229). Although this report is not a scientific publication, in my opinion, it is convincing. Unfortunately, the report is protected by copyright that makes comparisons of results more difficult (see Section 2.3.2).

## 2.2. Method

Content analysis is a method that is widely used in communications research. Some good properties of content analysis are: (a) it is a non-disturbing method because data occur regardless of whether the research is carried out or not, and (b) it is often possible to get a representative sample. In the present research, a quantitative content analysis of job advertisements was carried out; that is, the frequencies of different phrases such as C++, Java, WebLogic, and “operating systems” were simply calculated.

### 2.2.1. Choice of Data Source

In the previous longitudinal researches, newspapers or professional magazines have been used as data sources; that is, web recruiting services have not been used. The number of newspapers and magazines has varied from two to ten. The justification for several data sources has been to reduce the effect of possible regional differences. Use of newspapers has been more common than the use of professional magazines. Gallivan *et al.* (2002) is the only previous research that has used a professional magazine (Computerworld) as a data source. The commercial analysis (Salary Services 2004a, p. 286) used three professional magazines, six newspapers, and six web recruiting services.

In the present research, web recruiting services were not selected because they have not operated long enough for the purpose of the present research and data from the past few years are not publicly available. This was a very problematic situation because according to Salary Services (2004a, p. 2), the web services have dominated the IT job advertising market at least since the year 2000 and therefore, it was not certain if newspapers and professional magazines were still representative data sources. However, in my opinion (a), it was better to conduct a trend analysis using a magazine or a newspaper than not to conduct a trend analysis at all because data from web services were not available, and (b) the selected data source was the most suitable that was available.

Only one data source was used to keep the amount of work as reasonable. Newspapers such as New York Times were not selected because they were too regional. From various magazines targeted to IT professionals, Computerworld was chosen. Other possible professional magazines would be, for example, Communications of the ACM, IEEE Computer, and IEEE Software. The circulations of these four magazines are approximately as follows (Ulrich's periodical directory, 2004): Computerworld 250,000, IEEE Computer 97,000, Communications of the ACM 85,000, and IEEE Software 23,000. Computerworld is published weekly and the other three magazines six or 12 times per year. The main reasons to choose Computerworld were that it is a national magazine and might be more compelling to advertisers because it has the biggest circulation and is published weekly.

### 2.2.2. *Sampling*

In the previous longitudinal analyses, the periods have varied from six to 20 years and the intervals from one to seven years. Athey and Plotnicki (1998) and Gallivan *et al.* (2001) used unequal intervals whereas others used equal intervals. In the present study, the year 1990 was chosen as the starting year because WWW technology was released in 1993 and I wanted to get some results also before that year. Every second year was chosen as the interval to get more detailed results from possible trends. Every fifth year was not used because I assumed that changes in distributed technologies, in particular, might be so fast and large during the period that it would be beneficial to use a shorter interval.

Gallivan *et al.* (2001) and Maier *et al.* (1998) collected data from four numbers per year. Todd *et al.* (1995) collected data from each month of the year to avoid the possibility of seasonal or cyclical effects on data. However, in these three papers, it was not reported if there were any seasonal or cyclical effects. Athey and Plotnicki (1998), and Trower (1995) collected data from one number per year. In the present research, one number per year was chosen as sampling strategy. This decision was based on an assumption that there are no significant seasonal or cyclical differences. Normally, the sampling was the number 36. However, in 2002 and 2004 the sampling was three numbers (36–38) to get large enough subsamples from every year. The limit was set to at least 100 positions per year.

### 2.2.3. *Coding*

The selected 12 numbers had altogether 1004 job advertisements that were read and coded manually. In the previous longitudinal researches, apparently one advertisement was used as the unit of analysis. It is possible that one advertisement contains several job titles and one job title contains several positions. In (Salary Services, 2004a, p. 285) and in the present research, one position was used as the unit of analysis.

Gallivan *et al.* (2002) wrote: "In *Computerworld*, we initially treated ads for the U.S. East coast region and non-East regions separately. Subsequent analysis revealed that there was no statistical difference between the distribution of ads by job category or skill, and we subsequently combined the data for these geographic regions". In the present research, the advertisements were not separated by regions.

An advertisement was included if it contained at least one suitable position such as “Programmer”, “Programmer Analyst”, “Software developer”, or “Software engineer”. For example, the job titles “C++ developer” and “Web developer” were classified as software developer positions and included. A systems analyst position was included if it included also programming tasks. If this was unclear, a systems analyst position was excluded. A systems programmer position was included if it was rather a developer position than a systems administrator position. A systems programmer position was excluded if this was unclear. The following positions, for example, were excluded: “Business analyst”, “Consultant”, “Database administrator”, “Project manager”, “Quality assurance engineer”, and “Systems administrator”. A position was excluded if no position title was given or it was too general, for example “IT professional”.

In addition, also programmer etc. positions were excluded if the required degree was any doctoral degree or the field of study was not suitable for a computer science graduate, for example, Masters in Electrical Engineering was required. In other words, the position should be suitable for a Bachelor’s or Master’s degree graduate from a computer science program. Contractor positions were included, and part-time positions excluded.

Each suitable job advertisement described one or more open positions. If the number of positions was not given, it was coded as one position. If the number of positions was not given but text indicated plural (e.g., “Programmers”), it was coded as two positions. The exact number of positions was used if it was given (e.g., “5 programmers”). Altogether, these job advertisements contained 1291 suitable positions, which is the sample of the present research (i.e.,  $N = 1291$ ). The number of suitable positions for each year varied from 112 to 265. These are the subsamples of the present research (i.e.,  $n = 112...265$ ). The numbers of included advertisements, excluded advertisements, and excluded positions were not counted.

From advertisements of these suitable positions, it was searched for technical skills such as Cobol, Java, SQL, and Windows. These phrases were typically names or abbreviations of different programming languages, operating systems, database vendors, and protocols. Only required skills were included, and desired skills were excluded. A skill was coded as required if it was not mentioned if it was required or only desired.

#### 2.2.4. *Statistical Methods*

For each individual skill, a proportion was calculated for each year. For example, the sample of the year 1990 had 189 positions and in 77 of these positions, Cobol was required. Thus, the proportion of Cobol was 40.7%. In addition to these proportions for individual skills, several other figures were calculated. These coding principles are explained later before the corresponding results.

The Student t test and the Smith-Satterthwaite procedure (e.g., Milton and Arnold, 2003, pp. 347–349) were used to test if the difference between two means was statistically significant. The Z test for proportions (e.g., *ibid.*, p. 324) was used to test if the difference between two proportions was statistically significant. For significance, the following limits were used: not significant  $p \geq 0.05$ , almost significant  $p < 0.05$ , significant  $p < 0.01$ , and very significant  $p < 0.001$ .

In the previous longitudinal analyses, the use of statistical tests to analyze the results has been rare or non-existing, or at least it is not reported. Gallivan *et al.* (2001) have apparently used statistical methods but from other analyses I found nothing that would indicate the use of statistical tests.

### 2.3. Results

In this section, results about the number of required technical skills and five main skill categories are presented.

#### 2.3.1. Number of Required Technical Skills

The number of required individual technical skills was calculated for each position. For example, if Cobol and DB2 were required, the number of skills would be two. The minimum number was used if alternatives were given. For example, for the text “C++ or Java” the number of skills would be one. The mean value of these numbers was calculated for each year. Fig. 1 presents these means for three categories: “All”, “Programmers”, and “Others”. The software developer, software engineer, and systems analyst positions were combined as the category “Others” because some of the subsamples were too small to be presented alone. It can be noticed that the means have increased during the period. For example, the mean of the category “All” increased from 3.57 in 1990 to 7.66 in 2004. Based on the Student t test and the Smith-Satterthwaite procedure, the differences between the means of the years 1990 and 2004 are statistically very significant ( $p < 0.001$ ) for the categories “All” and “Programmers”, and significant ( $p < 0.01$ ) for the sample “Others”.

As mentioned at the beginning of Section 2, Todd *et al.* (1995) reported that the number of technical phrases in job advertisements for programmer positions increased from the mean of 2.2 in 1970 to 4.2 in 1990. They did not report if this difference was statistically significant, and it is not possible to test this because standard deviations were not reported, and the standard deviations or the original data were no longer available.

Maier *et al.* (1998, p. 38) and Gallivan *et al.* (2002, p. 7) have reported similar results. However, their samples included all kinds of MIS or IT positions, not just programmers. According to Maier *et al.*, the mean of skills per advertisement was 2.63 for the time period 1978/1979 and 3.50 for 1993/1994. Gallivan *et al.* (2002, p. 7) reported that the mean of technical skills increased from 3.0 skills per advertisement in 1988 to 4.2 in 2001. Maier *et al.* and Gallivan *et al.* neither reported if these differences were statistically significant nor standard deviations.

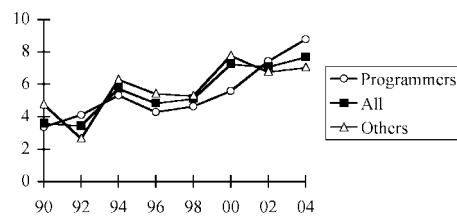


Fig. 1. Means of number of required technical skills in years 1990–2004.

### 2.3.2. Results for Five Main Skill Categories

During this part of analysis, the following criteria were used: (a) at least one common programming language skill (C, C++, Cobol, Java, or Visual Basic), (b) at least one operating systems skill (e.g., AS/400 or Windows NT), (c) at least one database skill (e.g., DB2, Oracle, or SQL), (d) at least one networking or computer networks skill (e.g., LAN or TCP/IP), and (e) at least one distributed technology skill (e.g., client/server or ASP). These five categories are called “Programming language”, “Operating systems”, “Database”, “Networking”, and “Distributed technology”. For each category proportion was calculated. For example, in the subsample of 1990 were 189 positions and in 92 of these at least one common programming language was mentioned. Thus, the proportion is 48.7%. Similar results for each year and for all five groups are presented in Fig. 2.

It can be noticed that the proportions for every category have increased. However, the increase of the category “Networking” has been small and no trend can be noticed. Based on the Z test for proportions, the differences between the proportions of the years 1990 and 2004 are statistically not significant ( $p \geq 0.05$ ) for the category “Networking” and very significant ( $p < 0.001$ ) for other categories.

Todd *et al.* (1995, p. 9) used the skill categories “Database” and “Operating systems” that are close enough to the corresponding categories in the present research. They did not report proportions for these categories, but they reported the numbers of phrases and the sample sizes. Based on these results, the proportions of the categories “Database” and “Operating systems” have apparently increased during the period 1970–1990. However, it is not possible to calculate proportions in the same way as in the present research. Athey and Plotnicki (1998, p. 76) reported that over 70% of all job opportunities required some knowledge of relational database technology. However, they did not report from which year this proportion was; their period was 1989–1996. They reported increasing trends to individual database skills. Also Trower (1995, p. 599) reported increasing trend to the category “Relational DB”.

From previous researches, Athey and Plotnicki (1998), and Trower (1995) are the most suitable for comparing the category “Distributed technology”. Trower (p. 599) reported that the number of ads in the category “Client/Server” increased from approximately from 40 in 1990 to 220 in 1995. It is not possible to count the proportions because

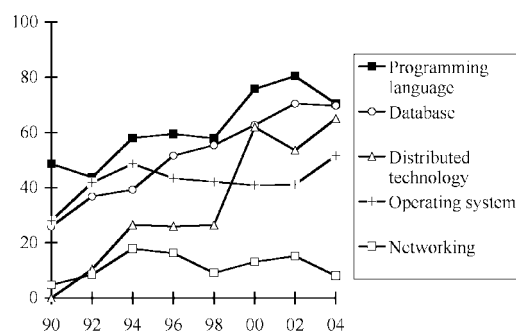


Fig. 2. Proportions (%) of at least one skill in five skill categories in years 1990–2004.



the sizes of the subsamples were not reported. However, Trower reported an index of all job advertisements as well. The numbers of the category “Client/Server” increased a lot faster than the index did. Anyhow, a more serious problem for comparison is that his coding principles for the client/server skills are problematic. He wrote (p. 598): “Finally, the leading client/server skills mentioned in the 1995 want ads are Windows (88 ads) and OS/2 (37 ads) for the GUI front-ends to client/server applications”. In my opinion, Windows and OS/2 should be classified as operating systems skills, not as a distributed technology skills.<sup>1</sup>

Athey and Plotnicki (1998) reported that the proportion of “Client Server” was 6.9% in 1993 and 7.4% in 1996. The proportion of 1993 matches well enough with the results of the present research but their result for the year 1996 is a lot smaller than the result of the present research. They did not report proportions of “Client Server” for the subsamples of the years 1989 and 1992.

Trower (1995, p. 599) reported that the number of advertisements of the category “Network” increased during the period 1990–95 faster than the index did. This matches with the results of the present research. Other previous researches had no suitable categories or results for comparison. However, Athey and Plotnicki (1998, p. 84) wrote: “Certainly networking is becoming more widespread. However, the number of advertised job opportunities in this area are surprisingly small”.

I found no previous results that would be suitable for comparing the category “Programming language” because previously, for example, the categories “2GL” and “3GL” have been used.

The report by Salary Services (2004a, pp. 224–229) presents the numbers of positions for each individual skill but no proportions. It was not possible to count the proportions because the subsample sizes were not reported for the years 1999–2002, but I counted the ranks for 150 individual skills for the years 1999–2003 from the results that were presented in the report. In addition, I used the ranks for the ten most common skills from the third quarter of 2004 that were presented at the web page (Salary Services 2004b). However, the ranks are not presented here but only commented because the permission from Salary Services Ltd. is required before any part of the survey may be reproduced. I compared the ranks against the results of the present research. These ranks confirm that the importance of distributed technology skills has increased during the past five years.

#### *2.4. Evaluation of Trend Analysis*

The use of web recruiting services has increased strongly during the past five years and as a consequence, the proportion of newspaper advertisements has decreased. For example, I estimated that the web recruiting service Dice<sup>2</sup> would publish approximately 0.3–0.8 million job advertisements and Computerworld approximately 3,000 job advertisements

---

<sup>1</sup>To be fair to Trower, it is understandable if he had problems to classify distributed technologies. Even nine years later, I had most problems with this category. For example, I reanalyzed data about distributed technology skills four or five times because I made progressive changes to the classification principles.

<sup>2</sup>Dice is the biggest web recruiting service in the USA (see Section 3 and <http://www.dice.com>).

in 2004. In other words, currently the web recruiting services appear to dominate the IT job advertisement market in the USA. Thus, it is possible that Computerworld is not a representative source for the period 2000–2004. However, in my opinion, Computerworld should be a representative source for the period 1990–1998.

There are two other problems with the representativeness of the sample: (a) Only one number per year was chosen as the sampling strategy. (b) It is possible that some well-known companies such as HP, IBM, Microsoft, and Sun Microsystems do not advertise in Computerworld at all or only a little because interested job seekers search job advertisements directly from the web site of the company. As will be reported later in Section 3.4, it was found that Microsoft was announcing in Dice only a very small proportion of open positions when compared with the number of positions that were announced on Microsoft's own web site. Anyhow, a much bigger problem for the present research is the increased use of web recruiting services in general.

### 3. Part 2. Cross-Sectional Analysis

The four main questions of this cross-sectional analysis were: (1) What skills were needed most in positions for programmers, software developers, and software engineers? In particular, distributed technology skills were analyzed thoroughly because as a consequence of World Wide Web technology, these skills might be required now more often than ten years ago. (2) What were differences between these three job titles? (3) What were the differences between entry-level and senior-level positions? (4) How well do entry-level job requirements match the requirements of a typical undergraduate program in computer science?

In the USA, web recruiting services are now dominant in the information technology job advertising market. The biggest service is Dice (<http://www.dice.com>) that was used for the present analysis because a large number of advertisements was necessary for some parts of the present analysis (e.g., finding a large enough sample of entry-level positions).

#### 3.1. Related Work

Next, only those five job advertisement analyses that have been published after the year 1999 are presented. Only the more recent publications are used for comparisons because older results are relevant to trend analysis but less relevant to the present cross-sectional analysis. The main topic of Adelman (2000) was certification. In addition, he has written a WWW page (Adelman, no date) that contains results about technical skills such as C++. Litecky and Arnett's (2001) paper contains results from both newspaper advertisements and web recruiting service. The used service was Monster.com and thus different than in the present research. Their paper includes results about individual technical skills such as Cobol, Unix, and Oracle. Apparently, the samples of Adelman, and Litecky and Arnett included IT positions of all kind, not just software developer positions. Gallivan *et al.* (2002) is the longitudinal research that was mentioned already earlier in the first part of

the present paper. However, their paper is too general to be relevant to the present analysis. Their coding scheme included also categories for individual technical skills such as Cobol and C, but their paper did not include these results.

Beside the three previous scientific publications and publications in professional magazines, there are two non-scientific reports and one conference presentation that are worth mentioning: (a) The quarterly reports from a British research company Salary Services Ltd. were mentioned already earlier in Section 2.1. (b) The content analysis of job advertisements in the ITAA report (Information Technology Association of America 2002, pp. 45–53) is based on Dice’s data, and thus, the data source is the same as in the cross-sectional analysis of the present research. (c) The slides from Prabhakar *et al.* (2004, p. 11) presentation contain results of individual technical skills such as Unix. The data source is apparently Monster.com,  $N = 4070$ , and the sampling period September 2004. These results are particularly relevant because they are recent and include some distributed skills as well.

From other type of research, the most relevant is the survey by Lethbridge (2000). He asked respondents their opinions on 75 topics: How much they had learned about it in their formal education, how much they knew about it at the time of answering, and how important the topic has been for their career?

### 3.2. *Method*

In February and July 2004, I searched at Dice advertisements that had the job titles Programmer, Software developer, or Software engineer. The searches produced 9680 advertisements. From these 9680 advertisements it was searched for technical skills, using phrases such as Java, SQL, TCP/IP, and Windows. These phrases were typically names or abbreviations of different programming languages, operating systems, database vendors, and protocols. Some more general phrases like “embedded”, “object-oriented”, or “relational” were used as well but this was not common. Note that during this part of analysis the advertisements were not read but only Dice’s search function was used. However, a part of the results was calculated with five smaller samples ( $N = 41\dots334$ ). The advertisements of these smaller samples were read and coded manually.

The following statistical equations, tests, and limits were used. The two-sided confidence intervals for proportions were calculated using equation from (Milton and Arnold, 2003, p. 315) and  $\alpha = 0.01$ . The Z test for proportions (e.g., *ibid.*, p. 324) was used to test if the difference between two proportions was statistically significant. For statistical significance, the following limits were used: not significant  $p \geq 0.05$ , almost significant  $p < 0.05$ , significant  $p < 0.01$ , and very significant  $p < 0.001$ .

### 3.3. *Results*

The results are divided into three subsections. First, the most common platform, programming language, and database skills are presented in the subsection “Updating earlier results”, where the results of the present research are also compared with prior findings.

Second, in the subsection “Results characteristic of the present research” results from topics that are more characteristics of the approach used in the present research are presented. Based on the literature survey, the topics of the second subsection have been researched only a little or not at all previously. Third, in the subsection “Verification of results” some results from other data sources are presented.

### 3.3.1. Updating Earlier Results

In Table 1, the top five platforms, programming languages, and databases are presented. For example, the proportion of Java was 35% as Java was mentioned in 3359 advertisements and the number of advertisements was 9680. In each column, the sum of the proportions can be greater than 100 because one position could be classified in more categories than one. Below the table are presented the confidence intervals for the worst cases inside each category.

Only the coding principles of the column “Platforms” are explained because they are less obvious than the coding principles of the two other columns. The following categories were used: Macintosh, Mainframe/midrange, Unix, Windows, and Cross-platform. For example, Windows refers to those positions where some Windows operating system or Windows based software such as Visual Basic or SQL Server was mentioned. Products that are available for both Windows and Macintosh (e.g., Word and Excel) were classified as Windows if Macintosh was not explicitly mentioned. The category “Cross-platform” refers to positions where only cross-platform products such as Oracle were mentioned.

The results of the present analysis were compared against the results in (Adelman, no date; Information Technology Association of America, 2002; Litecky and Arnett, 2001; Prabhakar *et al.*, 2004), and (Salary Services, 2004a; 2004b). For brevity and due the copyright issues, the detailed results from these sources are not presented but only the most important changes and general observations are mentioned. Not very much has changed during the past five years. For programming languages, the biggest change has been the strong increase of Java. For databases, the biggest change is the increased need of Microsoft SQL Server. The order of platforms is the same as five years ago when the category “Cross-platform” is excluded from the comparison. The proportions of the

Table 1  
Top five platforms, programming languages and databases

Rank	Platform	Programming language	Database
1	Windows 42%	Java 35%	Oracle 22%
2	Unix 29%	C++ 34%	SQL Server 11%
3	Mainframe/midrange 23%	C 26%	DB2 7%
4	Cross-platform 17%	Visual Basic 10%	Sybase 5%
5	Macintosh 0%	C# 9%	Access 4%

Confidence interval is  $\pm 5\%$  ( $N = 224$ ) for platforms, and  $\pm 1\%$  ( $N = 9680$ ) for programming languages and databases (except  $\pm 5\%$  and  $N = 224$  for Access).

present analysis are typically greater than the results presented by Prabhakar *et al.* (2004, p. 11) but the orders of skills match quite well. This is reasonable because apparently their sample was for all IT positions.

The results of Salary Services (2004a) from the last quarter of the year 2003 were detailed enough that I was able to count the proportions of individual skills for similar subsample as in the present research. In addition, I used results from the CD-ROM that was provided with the report. The report and CD-ROM listed results for approximately 50 job titles. I counted the proportions using the following job titles: Analyst Programmer, Graduate developer/analyst programmer, Programmer, Senior programmer, Senior software engineer, Senior systems developer, Software engineer, and Systems developer. I compared these proportions against the results of the present analysis. The results of programming language and database skills matched quite well. In platform skills, the proportions of Windows and Unix matched well enough but the proportion of mainframe/midrange was considerably smaller in Salary Services (2004a) than in the present research. As a consequence, the proportion of DB2 was considerably smaller than in the present research as well.

### 3.3.2. *Results Characteristic of the Present Research*

Results about distributed technologies, differences between job titles, differences between entry-level and senior-level positions, and comparing requirements in job advertisements against the degree requirements are presented in the following subsections.

#### *Distributed technologies*

For distributed technologies, three categories were used: Microsoft, Sun, and Other. A position was classified in a certain category if at least one skill of the category was mentioned. It is possible that one position was classified to several categories. The skills of each category are presented in the following lists:

- Microsoft: .NET, Active X, ASP, DCOM, IIS, and MTS;
- Sun: EJB, J2EE, JSP, RMI, and Servlets;
- other: technologies that do not belong in the previous two categories (e.g., CORBA, Tuxedo, Tibco, WebLogic, WebSphere, client-server, or applications server).

In 40% of the positions at least one distributed technology skill was required or desired. The proportions of categories were Sun 20%, Microsoft 17%, and other 8%. These results were calculated with the smaller sample ( $N = 224$ ), manual coding was used, and the confidence interval is  $\pm 5\%$ . The difference between Sun and Microsoft is statistically not significant and therefore, Sun's and Microsoft's technologies appear to have held an equally strong position. The most common individual distributed technology skills and their proportions are presented in Table 2.

In addition, 98 job advertisements that were published in Computerworld magazine in the year 1992 were analyzed. The purpose was to analyze if the need for distributed technology skills has increased from the year 1992 to 2004. The sample of Computerworld magazine was coded manually. The year 1992 was chosen because WWW technology was released in 1993. One hundred and twenty-one software developer positions were offered in these 98 job advertisements. From these 121 positions, eight were excluded

Table 2  
Most common distributed technology skills

Skill	Abbreviation	Company	Proportion (%)
.NET	–	Microsoft	19
Active Server Pages	ASP	Microsoft	18
Java 2 Enterprise Edition	J2EE	Sun	13
Java Server Pages	JSP	Sun	8
WebLogic	–	BEA	5
WebSphere	–	IBM	5
Enterprise Java Beans	EJB	Sun	4
Java Servlets	–	Sun	4
Internet Information Server	IIS	Microsoft	3
Common Object Request Broker Architecture	CORBA	–	2
Distributed Component Object Model	DCOM	Microsoft	2
Microsoft Transaction Server	MTS	Microsoft	1

Confidence interval  $\pm 1\%$ ,  $N = 9680$ .

because skills were not mentioned. Only in 4% of the positions ( $N = 113$ ) at least one distributed skill was required or desirable. This difference between Computerworld 1992 (4%) and Dice 2004 (40%) is statistically very significant ( $p < 0.001$ ).

Adelman's (no date) web page does not have results on distributed technologies. Litecky and Arnett (2001) reported that the proportion of WWW was 6%, Internet 9%, and "Network General" 9% in 1999. Prabhakar *et al.* (2004, p. 11) reported the following proportions: Web Programming 25.3%, .Net development 12.4%, and Client/Server or "C/S" 4.4%. These results match quite well with the results of the present analysis because their results were for all IT positions. However, based on these results it is not possible to conclude if Microsoft's technologies are required more often than Sun's. For example, it is possible that the category "Web Programming" of Prabhakar *et al.* refers to both Microsoft's and Sun's technologies. In addition, I counted from (Salary Services, 2004a; 2004b) the proportions for distributed technology skills in a similar manner as explained in Section 3.3.1. According to these proportions, Microsoft's distributed technologies are required more often than Sun's. However, this comparison was problematic because Salary Services Ltd. apparently did not use the category "J2EE" that was the most often mentioned Sun's distributed technology skill in the present research.

#### *Differences between job titles*

During this part of research, the purpose was to analyze if low-level programming skills would be more common in software engineering positions. According to Salary Services (2004a, pp. 294 and 296), (a) software engineers have in depth knowledge of real time or embedded software and associated hardware, and (b) software engineers are employed mainly by the electronics, computer, aviation, and defense industries. I assume that the job titles "Programmer", "Software developer", and "Software engineer"

Table 3  
Some differences in required or desired skills between job titles

Skill	Programmer (%)	Software developer (%)	Software engineer (%)
Assembler	1 (a)	4	14 (a)
C	16 (b,c)	30 (b,d)	40 (c,d)
C++	22 (e,f)	48 (e)	50 (f)
“embedded”	1 (g)	5	15 (g)

Confidence intervals are  $\pm 1...4\%$ ,  $n = 924...5418$ .

Letter pairs indicate that the difference is statistically significant ( $p < 0.01$ ).

are often used as synonyms. However, it is reasonable to expect that requirements for software engineers on average would emphasize low-level programming skills more.

Table 3 shows some results for the subsamples of programmers ( $n = 5418$ ), software developers ( $n = 924$ ), and software engineers ( $n = 3338$ ). Only those results that best show the differences about the low-level programming skills are presented. It can be noticed that assembler, C, C++, and the phrase “embedded” were more common for software engineering positions. Some differences between the job titles are statistically significant ( $p < 0.01$ ). These pairs are marked with small letters. For example, two letters “a” in the row “Assembler” mean that the difference between the proportions of programmers and software engineers is statistically significant. Thus, there is some evidence that low-level programming skills are more common in software engineer positions.

(Adelman, no date; Litecky and Arnett, 2001), and (Prabhakar *et al.*, 2004) are not suitable for comparison because they do not present results for different job titles. From the various job titles of Salary Services (2004a), I selected the titles Programmer, Software engineer, and Systems developer, and counted proportions for these subsamples. The report did not include the job title Software developer and the closest alternative was Systems developer. These proportions matched well with the results of the present analysis because according to the results of Salary Services Ltd., the low-level programming skills are more common in software engineer positions. However, according to the results of Salary Services Ltd., there were no big differences between programmers and systems developers.

#### *Entry-level versus senior-level positions*

Two groups were compared against each other: (1) Entry-level positions that had no word “senior” in the job title and the number of required working years was 0–1 ( $N = 41$ ). Often, these positions had the word “junior” as part of the job title. This sample was collected from Dice mainly in March 2004, using phrases like “junior” and “jr”. (2) Senior-level positions that had the word “advanced”, “lead”, “principal”, or “senior” in the job title or at least five years work experience was required ( $N = 73$ ). These two samples were coded manually. During this analysis, only the required skills were compared. In addition, individual skills such as Java and SQL were not compared because the sample sizes were so small. Only the following criteria were used: (a) at least one

common programming language (C, C++, Cobol, Java, or Visual Basic), (b) at least one common database skill (Access, DB2, “database”, Oracle, SQL, SQL Server, or Sybase), and (c) at least one distributed technology skill.

The average number of required skills was somewhat greater for the senior-level group. The average was 3.7 for the entry-level group and 5.2 for the senior-level group. The proportions of the different skill categories are presented in Table 4. According to the Z test for proportions, the difference in distributed technology skills was statistically very significant ( $p < 0.001$ ).

In addition, the average numbers of software development life cycle phases when the number of required experience varied were calculated. The phases presented in the IEEE standard (Institute of Electrical and Electronics Engineers, 1990, p. 68) were used for analysis. There were no big differences. Even the least experienced developers were typically required to take part in more phases than just implementation. Tutoring younger developers and leading small groups of developers were mentioned often for senior-level positions but obviously not for entry-level positions. The proportions of these duties were not calculated because they are non-technical skills.

(Adelman, no date; Litecky and Arnett, 2001), and (Prabhakar *et al.*, 2004) are not suitable for comparison because they do not present results for entry-level positions. Salary Services (2004a) contains the job title Graduate developer/analyst programmer that (p. 293) “is used to classify all programmers and developers who essentially have less than six months commercial experience on programming”. In addition, the report contains the job titles Senior programmer, Senior software engineer, and Senior systems developer. It was not possible to compare the groups using categories such as “at least one common database skill” because the report did not include suitable results. However, from the results presented in the report, I counted proportions for individual technical skills and compared these two groups against each other. On average, the proportions of senior-level positions were a little greater than for entry-level positions but the differences were not very big. In particular, the differences of individual distributed technology skills were so small that statistical tests were not even used to analyze the differences. Therefore, these results did not confirm the result of the present analysis that distributed technology skills were required in senior-level positions more often.

#### *Undergraduate programs versus required skills*

Next, let us consider how well current curricula in the USA correspond to the job market. McCauley and Manaris (2002) reported that in ABET/CAC accredited under-

Table 4

Proportions (%) of different skills categories for entry-level ( $N = 41$ ) and senior-level positions ( $N = 73$ )

Group	Programming language (%)	Database (%)	Distributed technology (%)
Entry-level	68	38	27*
Senior-level	73	48	59*

Star (\*) indicates that the difference is statistically very significant ( $p < 0.001$ ).



graduate programs the three most common programming languages that were taught first during the academic year 2001–2002 were Java (49%), C++ (40%), and C (11%). In this respect, the match between curricula and the job market is good because these three languages are exactly the same as the three most common programming languages in the job advertisements. This comparison does not imply that all degree programs should use these three languages. There can be other reasons than the popularity of language in industry to choose the programming language used in education – especially the first one. For example, some institutes might use Scheme as the first language because its syntax is simple.

In addition, McCauley and Manaris reported how often various upper-level courses were required. Related skills are presented in Table 5. The table combines results from two surveys and from the present research. The column “Proportion” is based on McCauley and Manaris’ survey and refers to the number of times a course was required in accredited programs.

The column “Importance” is based on the survey (Lethbridge, 2000) that was targeted to software developers. The means indicate how important the respondents thought that related skills are. The scale was 0–5, and a greater mean indicates greater importance. The data come from the Excel file that can be found in (Lethbridge, no date). If the survey had more than one related item per course title, the means were combined for the present paper.

In the column “Required in advertisements” is presented my estimation how often the related skills were mentioned in Dice’s advertisements. For this analysis, the exact proportions of phrases were calculated but they are not presented because the table would

Table 5

Most common upper-level courses, their proportions in accredited programs, importance of related skills in Lethbridge’s survey and estimation how often related skills were required in job advertisements. See the body text of paper for the explanation of the column “Required in advertisements”

<b>Course name</b>	<b>Proportion (%)*</b>	<b>Importance (mean)#</b>	<b>Required in advertisements (estimation)</b>
Operating Systems	96	3.3	Sometimes
Programming Languages	87	2.7	Hardly ever
Software Engineering	76	3.0	Sometimes
Architecture	69	2.7	Sometimes
Analysis of Algorithms	67	2.6	Sometimes
Theory of Computation	49	2.2	Hardly ever
Database Management Systems	31	3.3	Often
Networks	18	3.1	Sometimes
Compiler Construction	16	2.3	Sometimes
Artificial Intelligence	9	1.3	Hardly ever
Human-computer Interaction	4	3.3	Sometimes

\*Source (McCauley and Manaris, 2002). #Source (Lethbridge, no date).

become too complex. For example, for course Database Management Systems phrases “SQL”, “database”, “relational”, and “query” were searched for. The respective proportions were 7–32%. Similarly, related phrases from job advertisements for other courses were searched, too. The text “Hardly ever” refers to proportions 0–1%, “Sometimes” to 2–19%, and “Often” to at least 20%.

This part of research was the most problematic. Here, maybe the main finding is not the results presented in the column “Required in advertisements (estimation)” *per se* but realizing that this analysis has severe limitations because job advertisements do not contain enough suitable information. Simply put, one can find out from job advertisements that some skills are probably important but solving out if some particular skill or subject is *not important* can be much more difficult. Analysis works quite well for language and product names such as Java and WebSphere. For search phrases of other kind – that are typically more general terms, selection of phrases can have a dramatic impact into results. For example, if I had assumed that the course Programming Languages deepens the understanding of programming principles and is thus, relevant for any advertisement that mentions programming or a programming language, my estimation would be “Often” instead of “Hardly ever”. However, I have not omitted Table 5 because the results show that topics for at least eight out of 11 courses are required sometimes or often.

### 3.3.3. Verification of Results

A possible methodological problem with the present research is that Dice’s data were only from the first quarter of 2004. To verify the results also advertisements that were published in the Computerworld magazine in 2003 were analyzed using manual coding. The sample size was 334 that referred to the number of software developer positions. There were some differences between Dice’s and the Computerworld’s results, but the results of the present analysis would be similar if the data source were changed from Dice to Computerworld. Some differences were statistically significant but the order of skills was so similar that the differences had only a little practical relevance. For example, the biggest absolute difference was that the proportion of Java was 35% in Dice and 54% in the Computerworld. This difference is statistically very significant ( $p < 0.001$ ) but it has a little practical relevance because both results show that the need for Java was high.

In addition, the results of the present research were compared against the results of the recent British analysis (Salary Services, 2004a; 2004b). As already mentioned in Section 3.3.1, the most important difference was that, in the UK, the mainframe/midrange platform was mentioned more often than in the USA. As a consequence, the mainframe/midrange related skills Cobol and DB2 were more common in the USA as well.

### 3.4. Evaluation of the Cross-Sectional Analysis

Maybe the biggest problem with the used sample is a possibility that some well-known companies do not advertise in Dice at all or only a little because interested job seekers search job advertisements directly from the web site of the company. I tried to solve if this is the case for the following companies: HP, IBM, Microsoft, and Sun Microsystems.

From these four companies, it appears that Microsoft is announcing in Dice only a very small proportion of open positions but other three companies do announce in Dice. In October 14, 2004, Microsoft offered approximately 930 software development and testing positions in its own web site but only 20 positions in Dice. However, the total number of software developer positions in Dice was approximately 19,000. Compared with these 19,000 positions, the proportion of Microsoft's positions is only approximately 5%. Thus, correcting the sample by including some Microsoft's advertisements from the web page of the company would have only a small or moderate effect on results. Obviously, correcting the sample would probably increase proportions of skills that are related to Microsoft's products.

A typical coding problem for job advertisement analyses is so called fishing expedition advertisements where almost every common skill is mentioned. In some previous researches, for example (Litecky and Arnett, 2001, p. 7), these advertisements have been excluded. In the present research, fishing expedition advertisements were included because it was not possible to exclude them when Dice's automatic search function was used. This problem will probably result in a kind of constant background humming for the most common skills; that is, the proportions are a little greater than if fishing expedition advertisements would have been excluded.

Obviously, also the research method has its limitations. In some cases, job advertisements do not contain enough suitable information. Simply put, one can find out from job advertisements that some skills are probably important but solving out if some particular skill or subject is not important can be much more difficult. Analysis appears to work quite well for language and product names such as Java and WebSphere. For search phrases of other kind – that are typically more general terms such as “theory of computation” or “data structures and algorithms”, situation is much more difficult. Other research methods can be used to gather information about the importance of these subjects and skills. For example, Lethbridge (2000, p. 46) found in his survey that the respondents considered “Data Structures” and “Software Architecture” as being important. Third example might be concurrent programming because, for example, Java can be used for both object-oriented and concurrent programming but it is usually not possible to find out from the job advertisements if concurrent programming is necessary. According to my unpublished Delphi study, a small group of experienced software developers ( $N = 11$ ) considered concurrent programming as being important: the mean was 3.1 on the scale 1 Not at all important . . . 4 Very important.

#### **4. Conclusions and Implications for CS Degree Programs**

Implications of the previous results are considered only for university education, not for training institutions or for an individual software developer working in industry. Further, implications are limited only to the typical requirements of accredited computer science programs in the USA because there are no statistics from other countries similar to survey by McCauley and Manaris (2002). The paper by Parnas (1999) concerned the differences

between CS and SE programs. He wrote: “In the SE program, the priority will be usefulness and applicability; for the CS program it is important to give priority to intellectual interest, to future developments in the field, and to teaching the scientific methods that are used in studying computers and software development”. Obviously, the results of the present research are more relevant to SE programs. However, I write about CS programs and software systems specializations because the number of SE programs is so small. As Parnas put it, “Computer Science departments have tried to fill the gap by including so-called ‘Systems’ or ‘Applied Computer Science’ courses in their offerings”.

Based on the results of the present research, the results by Gallivan *et al.* (2002), Maier *et al.* (1998), and Todd *et al.* (1995), the conclusion is that the technical requirements for software development positions have changed during the past 35 years so that the number of required individual skills has increased on average. This change is so large and found in more than one research that problems with sampling or other simple explanations are not plausible.

Duties of software developers have changed as technically more versatile in that respect that typically it is no longer enough to have skills only in 1–2 programming languages. In particular, the results in Section 2.3.2 indicate that it has become more and more common that software developers are expected to have database and distributed technology skills as well. For universities and other institutes, one possible strategy or tactic to respond to the increased and more complicated technical demands of industry is specialization to the certain skill groups or sectors of the job market.

A typical or reasonable list of skills for a graduate from an undergraduate program in computer science might be like the following when individual technical skills similar to the present research would be used: C, Java, SQL, UML, and Unix; that is, five skills altogether. For example, it is not reasonable to expect that graduates from an undergraduate program would have good skills in more than two programming languages. For a master degree, a similar list could be, for example, the following: C, Java, JDBC, RMI, Scheme, SQL, UML, and Unix; that is, eight skills. 5–8 skills are still at the same level as the number of required skills in the job advertisements on average in the year 2004. However, if the number of required skills continues to increase, degree programs might have severe difficulties in following this change. After all, degree requirements typically include many other topics than technical skills as well (e.g., mathematics, general skills such as project management, communication and language skills).

Based on the results of the present research, the course Database Management Systems should be compulsory more often. This being said, it would be fair to suggest which one of the other upper-level courses could be changed from compulsory to elective in order to make room for a Database Management Systems course. However, I do not make a such suggestion. As explained in Section 3.3.2, this type of analysis is too problematic to find out subjects or skills that are not important.

The results of this study imply that the need for distributed technology skills has increased during the last ten years. However, the results also imply that distributed technologies are not common enough in entry-level positions to make a distributed systems course compulsory in undergraduate programs. These skills are common in intermediate-level and senior-level positions and therefore, the distributed systems are a very suitable

topic for a compulsory course in a software systems specialization in an industry-oriented master's program.

The need for continuous education should be high because technology has changed continuously. One way to promote continuous education would be to offer part-time master's programs to complement traditional degree programs. In Europe, an interesting question is whether the on-going harmonization of degrees called as the Bologna process (e.g., The Bologna Process. . . , 2005) will increase the number of part-time master's programs. Anyhow, software systems specialization is not exceptional in this respect because the need for continuous education is apparently high for most if not all areas of computer science.

### Acknowledgements

I would like to thank professor L. Malmi from the Helsinki University of Technology and professor emeritus V. Meisalo from the University of Helsinki for commenting on the manuscripts of the present paper.

### References

- Adelman, C. (2000). A parallel universe. *Change*, **32**(3), 20–29.
- Adelman, C. *A Parallel Universe, Expanded*. Retrieved on February 3, 2004, from the American Association for Higher Education (AAHE) web site:  
<http://www.aahe.org/change/paralleluniverse.htm>
- Athey, S., and J. Plotnicki (1998). The evaluation of job opportunities for IT professionals. *Journal of Computer Information Systems*, **38**(3), 71–88.
- Gallivan, M., D.P. Truex III, and L. Kvasny (2002). An analysis of the changing demand patterns for information technology professionals. In *Proceedings of the 2002 ACM SIGCPR Conference on Computer Personnel Research*, May 14–16, 2002, Kristiansand, Norway, pp. 1–13.
- Information Technology Association of America (2002). *Bouncing Back: Jobs, Skills and the Continuing Demand for IT Workers*.
- Institute of Electrical and Electronics Engineers (1990). *IEEE Standard Glossary of Software Engineering Terminology*. IEEE Std 610.12–1990.
- Lethbridge, T. (2000). What knowledge is important to a software professional? *Computer*, **33**(5), 44–50.
- Lethbridge, T. (no date). *1998 Education Relevance Survey Results*. Retrieved on June 20, 2004, from University of Ottawa web site:  
<http://www.site.uottawa.ca/~tcl/edrel/EdrelData1998.xls>
- Litecky, C., and K. Arnett (2001). An update on measurement of IT job skills for managers and professionals. In *Proceedings of the Seventh Americas Conference on Information Systems*. Boston, MA, August 2001, pp. 1922–1922.
- Litecky, C., B. Prabhakar and K. Arnett (1996). MIS job market: Shaken but not stirred. *Journal of Systems Management*, **47**(4), 51–54.
- Maier, J.L., W.J. Clark and Jr. W.S. Remington (1998). A longitudinal study of the management information systems (MIS) job market. *Journal of Computer Information Systems*, **39**(1) (Fall), 37–42.
- McCauley, R., and B. Manaris (2002). *Comprehensive Report on the 2001 Survey of Departments Offering CAC – Accredited Degree Programs*. Retrieved on February 11, 2004, from College of Charleston web site:  
<http://stono.cs.cofc.edu/~mccauley/survey/report2001/CompRep2001.pdf>
- Milton, J., and J. Arnold (2003). *Introduction to Probability and Statistics*. 4th Edition. McGrawHill, New York.
- Nakayama, M., and N. Sutcliffe (2000). Introduction to research on IT skill issues. In *Proceedings of Americas Conference on Information Systems 2000*, August 10–13, Long Beach, CA, pp. 1930–1934.
- Nakayama, M., and N. Sutcliffe (2001). IT skills portfolio research in SIGCPR proceedings: Analysis, synthesis and proposals. In *Proceedings of the 2001 ACM SIGCPR Conference on Computer Personnel Research*. San Diego, CA, pp. 100–113.

- Parnas, D. (1999). Software engineering programs are not computer science programs. *IEEE Software*, **16**(6), 19–30.
- Prabhakar, B., K. Arnett, and C. Litecky (2004). *Turnaround in the MIS job market?* Slides from the presentation in the meeting of Decision Science Institute. The meeting was organized apparently November 20–23, 2004, Boston, MA. The slides were obtained by e-mail from Litecky.
- Prabhakar, B., C. Litecky and K. Arnett (1995). Boom times ahead! *Journal of Systems Management*, **46**(1), 24–28.
- Salary Services Ltd. (2004a). *ComputerWeekly. Survey of Appointments Data and Trends*. Quarterly survey. January 2004.
- Salary Services Ltd. (2004b). *TOP IT Skills–All*. Retrieved December 27, 2004, from the Salary Services Ltd. web site:  
<http://www.salaryservices.co.uk>
- Surakka, S. (2004). Analysis of job advertisements: what technical skills do software developers need? In A. Korhonen and L. Malmi (Eds.), *Kolin Kolistelut–Koli Calling 2004. Proceedings of the Fourth Finnish/Baltic Sea Conference on Computer Science Education*, October 1–3, 2004, Koli, Finland, pp. 47–56.
- Surakka, S. (2005). *Trend Analysis of Job Advertisements: What Technical Skills Do Software Developers Need?* Helsinki University of Technology, Department of Computer Science and Engineering, Laboratory of Information Processing Science. TKK-TKO-B156.  
<http://www.cs.hut.fi/Publications/Reports/B156.pdf>
- The Bologna Process – Towards the European Higher Education Area* (2005). Retrieved March 22, 2005, from the Bologna Process web site:  
<http://www.bologna-bergen2005.no/EN/BASIC/Pros-descr.HTM>
- Todd, P.A., J.D. McKeen and R.B. Gallupe (1995). The evolution of IS job skills: a content analysis of IS job advertisements from 1970 to 1990. *MIS Quarterly*, **19**(1), 1–27.
- Trower, J.K. (1995). The impact of job skill requirements on I.S. curricula. In *Proceedings of the First Americas Conference on Information Systems*. August 25–27, 1995, Pittsburg, PA, pp. 597–599.
- Ulrich's Periodical Directory* (2004). Retrieved November 24, 2004, from the web site:  
<http://www.ulrichsweb.com/ulrichsweb>

**S. Surakka** is a doctoral student at the Helsinki University of Technology where he received master's degree in 1993. The topic of his doctoral thesis is the evaluation of software systems education.

## Programuotojų ieškančiuose skelbimuose reikalaujamų techninių igūdžių analizė

Sami SURAKKA

Straipsnį sudaro dvi dalys. Pirmojoje atliekamas 1990–2004 metais vyravusių tendencijų aptarimas, antrojoje – 2004 metų padėties rinkoje skersinis pjūvis. Abiejose dalyse stengiamasi išryškinti JAV rinkoje programuotojams dažniausiai taikomų techninių igūdžių reikalavimus. Skirtingai nuo ankstesnių tyrimų, didžiausios paklausos susilaukiantys techniniai igūdžiai analizuojami atsižvelgiant į internetinės technologijos galimybes: šiandien specialistai, turintys šių igūdžių, kur kas labiau paklausūs nei prieš dešimt metų. Turint omeny esamų tendencijų analizę, reikalingų techninių igūdžių skaičiaus vidurkis nuo 3,6 pakilo iki 7,7. Tokiu būdu programuotojams keliami reikalavimai tapo kur kas įvairiapusiškesni. Technologinių igūdžių pareikalavimo proporcijos smarkiai pakilo: jei 1990 jos sudarė 0%, tai 2004 m. – jau 65,1%. 2004 m. darbo skelbimų analizė parodė, jog šiuo metu labiausiai reikalaujamų techninių igūdžių penketukas išsirikiuoja šitaip: *Windows*, *Java*, *C++*, *SQL*, *Unix*. Be to, informatiko išsilavinimas dažnai suprantamas kaip privaloma sąlyga.