Informatics in Education, 2008, Vol. 7, No. 2, 237–258

A Methodological Review of the Program Evaluations in K-12 Computer Science Education *

Justus J. RANDOLPH
Department of Computer Science and Statistics, University of Joensuu
P.O. Box 111, FIN-80101 Joensuu, Finland
e-mail: justusrandolph@gmail.com

Received: June 2006

Abstract. Because of the potential for methodological reviews to improve practice, this article presents the results of a methodological review, and meta-analysis, of kindergarten through 12th grade computer science education evaluation reports published before March 2005. A search of major academic databases, the Internet, and a query to computer science education researchers resulted in 29 evaluation reports that met stringent criteria for inclusion. Those reports were coded in terms of their demographic characteristics, program characteristics, evaluation characteristics, and evaluation findings.

It was found that most of the programs offered direct computer science instruction to North American high school students. Stakeholder attitudes, program enrollment, academic achievement in core courses, and achievement in computer science courses were the most frequently measured outcomes. Questionnaires, existing sources of data, standardized tests, and teacher- or researcher-made tests were the most frequently used types of measures. Based on eight programs that offered direct computer science instruction, the average increase on tests of computer science achievement over the course of the program was 1.10 standard deviations, or the statistical equivalent of 73 out of 100 program participants having shown improvement. Some of the main challenges for the evaluation of computer science education programs are the absence of standardized, reliable, and valid measures of K-12 computer science education and coming to understand the causal links between program activities, gender, and program outcomes.

Keywords: methodological review, meta-analysis, computer science education, evaluation, methods.

1. Introduction

There are both economic and social needs for high-quality kindergarten through 12th grade (K-12) computer science education. The U.S. Department of Labor, Bureau of Labor Statistics, projects that the "employment of computer specialists is expected to grow much faster than average for all occupations as organizations continue to adopt and integrate increasingly sophisticated technologies" (2004). K-12 computer science education helps prepare individuals to attain advanced computing degrees, which, in turn,

*This research was supported in part by the Fulbright Center for Finnish-American Academic Exchange and by a special projects grant from the Association for Computing Machinery's Special Interest Group on Computer Science Education.

help those individuals meet the rapidly changing technological needs of business and industry. Even for those who do not intend to go into computing as a profession, some degree of computing skill and knowledge will be necessary to meaningfully participate in the technologically oriented societies of the future (Breslin, 1990; The National Research Council Committee on Information Technology Literacy, 1999).

The SIGCSE Working Group on Evaluation (Almstrum *et al.*, 1996) pointed out that there are many groups who stand to gain from the practice of evaluation[1]: those "constituencies that stand to benefit from what we [computer science educators] learn [from evaluation] include ourselves, our community of colleagues, and society as a whole. The ultimate beneficiaries of our learning, however, are our students" (p. 202). Some of the reasons that Almstrum *et al.* give for conducting evaluations of computer science education programs[2] are presented below:

- to satisfy our curiosity about what works and what doesn't;
- to discover issues of importance to ourselves and our students;
- inform our course development process;
- to compare alternatives;
- to help to identify important factors in a complex phenomenon;
- to gain the ability to make informed decisions;
- confirm or refute conventional wisdom;
- justify actions with cost/benefit analysis;
- validate research proposed to outside sources (p. 202).

Conducting a review of program evaluations is as necessary to evaluation as conducting a high quality literature review is to research. A review of previous evaluations helps evaluators get acquainted with the contexts and issues in their program's field, it familiarizes them with the research designs and measures being used by their peers, it helps identify key variables, and it can indicate what the expected results of a particular type of program should be. A review can also be an indicator of the state of the research and, thereby, motivate evaluators to keep doing what they do well and rectify what they do not do so well. Moreover, systematic reviews of program evaluations also benefit policy makers directly by synthesizing information that is needed for informed decision making (Carvalho and White, 2004; Cooper and Hedges, 1994; Joint Committee on Standards for Educational Evaluation, 1994; Weiss, 1998).

There have been several methodological reviews of computer science education research (see, e.g., Randolph, 2007a; Randolph *et al.*, 2005a; and Valentine, 2004) and reviews of resources for evaluating programs in computer science education (Randolph and Hartikainen, 2004). However, there have been no previous systematic reviews of the *program evaluations* in K-12 computer science education.

[1]Throughout this article, because of the unresolved debate regarding what should be considered *evaluation* and what should be considered *research*, I consider an investigative activity to be evaluation, rather than research, if the investigators state that they are doing evaluation, rather than research. In general, I define (program) evaluation as an activity whose primary goal is to answer questions that are important to program stakeholders; whereas, I define research as an activity whose primary goal is to answer questions that are important to the scientific community. See (Randolph, 2007b).

[2]I mean *program* in the sense of a project, not in the sense of software.

Given the benefits of systematic reviews of program evaluations, and a lack of such reviews in the field of computer science education, I conducted a systematic, methodological review of the evaluations of K-12 computer science education programs. The research questions answered by this review are listed below:

1. What are the methodological characteristics of computer science education program evaluations?
2. What are the demographic characteristics of computer science education evaluation reports?
3. What are the characteristics of computer science education programs that are being evaluated?
4. What is the average effect of a particular type of program on computer science achievement?

The answers to Questions 1, 2, and 3 will help evaluators of computer science education programs acquaint themselves with the methods that have been used in the past, with the trends and contexts of the field, and with the characteristics of the programs that they may be asked to evaluate. The answer to Question 4 will potentially allow evaluators to compare the effects of the programs that they evaluate to the effects of other, similar programs. For example, the answer to Question 4 will allow evaluators to make statements like "the effects of this program are greater than the effects of similar programs", instead of simply stating "this program has an effect greater than zero". Finally, this review, because it draws on evaluations from both computing science and program evaluation traditions, will help bridge the gap between those fields.

In the next section, I discuss the coding procedure, coding variables, literature search, criteria for inclusion, and methods of data analysis that were used. In the results section, I report the methodological, demographic, and program characteristics of all of the evaluations included in the review and report the pooled effect size, in terms of computer science achievement, for eight evaluations in which an experimental or quasi-experimental method was used. I also report the results of a subgroup analysis of types of programs because six of eight effect sizes came from evaluations of the same program. In the discussion section, I report potential biases in the literature, discuss the results for each study question, and point out study limitations. In the final section, I summarize the results, spell out their implications for practitioners and evaluators of computer science education programs, and discuss some of the main challenges for the field.

Methods

In this section, I report on the search strategies used to find relevant evaluation reports, the criteria used for including an evaluation report in this analysis, the variables that were coded, and the procedures for establishing interrater reliability. The variables that were coded can be grouped into four categories: *demographic characteristics*, *program characteristics*, *evaluation characteristics*, and *findings*.

Search Strategy

Several search strategies were used to find evaluation reports for this review. First, the academic databases – *Academic Search Premier, TOC Premier, PRE-CINAHL, Computer Source, ERIC, Library literature and Information Science, Newspaper Source, Psychology and Behavioral Science Collection, PSCYINFO, Social Science Abstracts, Communication and Mass Media Complete*, and *Vocational and Career Collection* – were searched, via *EBSCHO HOST*, in July of 2004 using the keywords *computer science education* and *program evaluation*. The unit of data collection was the evaluation report. In March of 2005, electronic searches of the ACM Digital Library and of the Internet, via the Google search engine, were conducted using six combinations of the phrases "*computer science education*", "*K-12*", "*evaluation*", and "*program evaluation*". The abstracts, descriptions, or links of the first 200 entries of the Internet and ACM Digital Library searches were examined for each combination until it could be determined that the entry would not plausibly lead to an evaluation report that would meet the criteria for inclusion.

From the references section of the articles that were found from the electronic searches, a branching, hand-search was used to identify other reports that would meet the criteria for inclusion until a point of saturation had been reached. After a preliminary list of evaluation reports was gathered from the electronic and hand searches, an e-mail message was sent to the 2,795 subscribers of the EVALTALK listserv and to the 1,112 members of the ACM SIGCSE-Members listserv on March 15, 2005; the message asked subscribers to send information about evaluation reports that met the criteria for inclusion but were not on the preliminary list.

Criteria for Inclusion

The following criteria were used to determine which evaluation reports, (i.e., reports in which the authors specified that they conducted 'evaluation') would be included in the review:

1. The evaluation report concentrated on a particular computer science education program and not on a particular computer science education strategy or application.
2. The report was written in English.
3. The direct beneficiaries of the program were K-12 students.
4. The programs delivered the types of computer science education content mentioned in the ACM Model Curriculum for K-12 Computer Science (Tucker *et al.*, 2002) to K-12 students.
5. Evaluation reports that concentrated only on the evaluation of computer infrastructure for computer science education programs were not included.
6. Evaluations of computer science education teacher training programs were only included if they examined students' resulting computer science achievement.
7. Studies were included in the meta-analyses proper if there was enough information reported to calculate Cohen's d and if they met the previous six criteria.

Variables Coded and Procedures for the Interrater Reliability Check

After all of the categories for each of the variables were determined and the evaluations were coded by the primary rater, a secondary rater coded the key study characteristic variables – *type of inquiry, type of experimental design* and *study quality* – on four randomly selected evaluation reports. Kappa (i.e., Brennan and Prediger's (1981)) κ_m) and percent of overall agreement were used as the interrater agreement statistics.

The variables of the coding sheet were grouped into four categories: *demographic characteristics, program characteristics, evaluation characteristics*, and *findings*. Categories for each variable were created using an emergent coding procedure, except for *curriculum area, type of inquiry*, and *evaluation approach*, where a priori coding categories was used. The categories that resulted via the emergent coding procedure are presented in Tables 2 through 6.

Demographic Study Characteristics. Demographic variables included *evaluation author, country of origin*, and *source of publication*. It also included *year of publication.*

Program characteristics. Several characteristics of the programs were coded, such as *type of program activities, target population, type of school* (i.e., public or private), *grade level,* and *type of delivery* (i.e., onsite or distance). Additionally, the type of instruction that each program delivered was classified according to the various areas of the Association for Computing Machinery's (ACM) K-12 computer science education curriculum (Tucker *et al.*, 2003).

Evaluation methodology characteristics. Several evaluation methodology characteristics were coded for each evaluation report: *the outcomes that the evaluation examined; the type of inquiry used; the type of instrument used; whether the instrument was quantitative, qualitative, or mixed; the moderating variables investigated*; and *type of evaluation approach.*

The categories used for type of inquiry, which are adapted from (Randolph *et al.*, 2005a), were *survey research, qualitative research, causal- comparative research, experimental/quasi-experimental research, correlational research*, or *classification research*. The definitions for each category are explained briefly below. Survey research describes the characteristics of a population without comparing groups or making causal conclusions. Qualitative studies explain a phenomenon through what Mohr (1999) calls physical causal reasoning, through what Scriven (1976) calls the modus operandi approach, or through what Shadish, Cook, and Campbell (2002) call causal explanation. Causal comparative studies compare two or more groups on an inherent variable. In experimental/quasi experimental investigations, the evaluator compares a factual to a counterfactual condition to make causal conclusions (Shadish *et al.*, 2002). Correlational investigations examine how levels of one variable covary with levels of another variable. If studies were classified as *experimental/quasi-experimental*, the experimental design was classified into one of the following categories: *Pretest-postest with control, pretest-postest without control, posttest with control, one-group posttest-only*, and *longitudinal*. See (Randolph *et al.*, 2005a) for a more-detailed description of these categories.

Stufflebeam's (2001) framework of evaluation approaches was originally used to categorize the sample of evaluations into four categories: *questions or methods oriented,*

decision/improvement oriented, pseudo-evaluation, or *social/agenda oriented*. However, this variable was abandoned because acceptable levels of interrater reliability could not be established.

Ratings of study quality for experimental/quasi-experimental designs were based on study design and the degree of controls for the threats to internal validity (see Shadish *et al.*, 2002). Study quality was rated as *high* if the evaluator used a pretest-posttest with control group design or a multiphase, repeated measures design and there was no evidence of threats to internal validity. If there was evidence of threats to internal validity, studies using those designs were rated as *medium*. A study was rated as *high* if a pretest-posttest without control group design or a posttest-only with control group design was used and there was no evidence of threats to validity. Otherwise, studies using those designs were rated as *medium*. Studies that used the one-group posttest-only design were rated as *low* unless there was very strong evidence that they controlled for threats to internal validity, in which case studies that used that design were rated as *medium*.

Findings. For experimental/quasi-experimental evaluations that quantitatively examined the effects of a program on computer science ability and reported means and standard deviations, or F or T statistics, Cohen's d was the effect size metric used.

Data Analysis

To answer study questions about demographic, program, and evaluation characteristics; frequencies were calculated using the evaluation case, which were sometimes single reports and sometimes series of evaluations of the same program, as the unit of analysis. For the study question about the average effect of computer science programs on student achievement, the unit of analysis was the evaluation report. Cohen's *d*, with Hedge's (g^U) bias correction (Rosenthal, 1994), was used as the common metric for outcomes of quantitative measures of computer science achievement (i.e., teacher- or research-made tests). The bias-corrected effect sizes were calculated using *Effect Size Calculator* (n.d.) software. A variance and within-study sample size / study quality weighting approach as described in (Shadish and Haddock, 1994), was used to weight studies. Lipsey and Wilson's (2001) *Metaf* SPSS macro was used to calculate statistics for main effects and for interactions between type of program (i.e., either the Nature-Computer Camp program or programs other than Nature-Computer Camp) and outcomes of computer science achievement. A random-effects model was used for these analyses if the fixed-effect homogeneity of variance was rejected, as indicated by a fixed-effect p value of Q_{total} less than 0.05 (Hedges, 1994; Raudenbush, 1994).

Results

Search Results

The EBSCO host search yielded 85 entries; the electronic searches yielded 1,123 entries. Although those entries led to many evaluation reports of computer science education

Table 1

Description of program evaluations included in this review

Evaluation	*Description*
Lew, 1971	A questionnaire-based evaluation of a computer science program for secondary-school-age youth who are underprivileged
Durward, 1973	Methods-based program evaluation of computer-science and computer-assisted instruction in Vancouver secondary schools; partly quasi-experimental.
Haughey *et al.*, 1980	An evaluation of CS/data processing in Manitoba schools. It examines a cohort of graduates in terms of university studies and careers.
Worthington City School District, 1983	Informal evaluation of a pilot project that used Logo in the K-3 classrooms.
Still, 1985	An evaluation of CS abilities using a pretest-posttest with control group design and investigating attitudes and achievement of a computer literacy program for grades K-8.
Carabetta, 1987	An informal, cost-benefit evaluation of a high school programming contest.
Akenegbu, 1992; DC, 1983, 1985a, 1985b, 1986; Negero, 1994	Multiple evaluations investigating academic achievement, behavior and socialization of 6th Grade participants in Nature-Computer Camp from 1983 to 1994.
Berney and Alvarez, 1990a, 1990b	An evaluation of a program that provided instruction in computer skills to limited-English-proficient Spanish-speaking students in a New York high school.
Atwater, 1991	An evaluation of Computers Unlimited Magnet Elementary schools that examined program implementation, stakeholder attitudes, academic achievement, and participation of minorities.
Kirkpatrick *et al.*, 1991	An evaluation, using an experimental design, of 21 science, math, and computer enrichment programs.
Atwater, 1992	An evaluation, using standardized tests with experimental designs, of Computers Unlimited Magnet High Schools.
Piña, 1992	An informal evaluation of a computer literacy program.
Seever, 1992	An evaluation using a standardized test and experimental designs of Computers Unlimited Magnet Middle Schools.
Fitzgerald and Hines, 1996	An informal evaluation of a computer science fair for 6th–12th grade students.
Walker and Rodgers, 1996	An evaluation of a program to decrease the pipelining of female students of computer science.
Golan and Means, 1998a, 1998b; Penuel *et al.*, 2000, 2001; Means *et al.*, 2001	A series of reports that used a variety of designs and measures to evaluate the Silicon Valley Challenge 2000 program from 1998 to 2001.
Crombie *et al.*, 2002	Evaluation study in all-female, high school classrooms.
Torvinen, 2004	An evaluation of a distance, computer science education program targeted for high school students in eastern Finland.
Randolph *et al.*, 2005	An evaluation of a computer science education program, which concentrates on robotics teaching, in eastern Finland.

programs, only 29 evaluation reports met the criteria for inclusion. The evaluation reports that were included in this review are preceded by an asterisk in the references section. In instances when there were multiple, periodic evaluation reports that used the same evaluation methodology, those reports were collapsed and considered to be one evaluative case. After collapsing reports, there were 19 evaluative cases. See Table 1 for a brief description of the 19 evaluative cases included in this study.

Akenegbu (1992), District of Columbia – Division of Quality Assurance [Hereafter *DC*] (1983, 1985a, 1985b, 1986), and Negero (1994); Berney and Alvarez (1990a, 1990b); and Golan and Means (1998a, 1998b), Means, Penuel, Korbak, and Kantor (2001), Penuel, Golan, Means, and Korbak (2000), and Penuel, Korbak, Yarnall, and Pacpaco (2001) were collapsed into single evaluative cases. (That is, these 13 cases were collapsed into 3 cases.) Of the 29 evaluations reports that met the first six criteria, only 8 reports (Akenegbu, 1992; DC, 1983, 1985a, 1985b, 1986; Negero, 1994; Durward, 1973; and Still, 1985) met the seventh criterion and were included in the meta-analysis of computer science achievement.

Demographic Characteristics

Table 2 presents the demographic characteristics for the 19 evaluative cases. It shows that most of the evaluation reports came from North America, most were found from the ERIC database, and that there had been an increasing number of computer science evaluations being reported every decade since the 1970's.

Program Characteristics

Table 3 presents the target participants, their grade levels, the curriculum area that was targeted, and the activities that were conducted in the 19 evaluative cases. In general, the

Table 2
Demographic characteristics: region, source, and decade of publication

Report characteristics	Frequency (%)
Region of origin (out of 19 cases)	
North America	17 (89.5)
Europe	2 (10.5)
Source (out of 19 cases)	
ERIC	13 (68.4)
Journal	3 (15.8)
Unpublished	3 (15.8)
Decade of publication (out of 19 cases)	
1970s	2 (10.5)
1980s	5 (26.3)
1990s	8 (42.1)
2000s	4 (21.1)

Table 3

Program characteristics: grade level, target population, curriculum area, activities

Program characteristic	*Frequency (%)*
Grade level of target participants (out of 19 cases)	
K-3	1 (5.3)
4–6	2 (10.5)
7–9	2 (10.5)
10–12	9 (47.4)
Mixed	5 (26.3)
Target population (out of 19 cases)	
General students	15 (78.9)
Students with limited proficiency in language of instruction	1 (5.3)
Students with disabilities	1 (5.3)
Female students	2 (10.5)
ACM Curriculum area (out of 17 cases)	
k-2	2 (11.8)
3–5	6 (35.3)
6–8	8 (47.1)
9–10	10 (58.8)
10–11	5 (29.4)
11–12	1 (5.9)
Program activities (out of 64 program activities in 19 cases)	
Student instruction	55 (86.0)
Teacher instruction	4 (6.3)
Computer science fair/contests	2 (3.2)
Mentoring	1 (1.5)
Speakers at school	1 (1.5)
Computer science field trips	1 (1.5)

Note. More than one curriculum area was possible per case or program activity was possible per case. Two cases did not provide enough information to determine the curriculum area.

data in Table 3 indicate that general education, high school students were most often the target participants of the programs. The curriculum areas that were targeted correspond with the 6th–8th grade and 9th–10th grade levels of the Tucker *et al.* (2003) curriculum. The data also indicate that student instruction was the most frequent type of program activity. Unfortunately, the evaluation reports, in general, did not report in detail what approach to student instruction was taken.

Methodological Characteristics

Table 4 presents the findings that concern the program outcomes and the student-level factors that were examined. Stakeholder attitudes were the most frequently investigated outcome, followed by levels of enrollment, achievement in core subjects, and achieve-

Table 4
Methodological characteristics: outcomes and factors

Methodological characteristic	*Frequency* (%)
Outcome (out of 67 outcomes in 19 cases)	
Stakeholder attitudes	17 (25.4)
Enrollment	13 (19.4)
Achievement in core subjects	14 (20.9)
Computer science achievement	9 (13.4)
Teaching practices	5 (7.5)
Intentions for future CS jobs/courses	3 (4.5)
Program implementation	2 (3.0)
Costs and benefits	2 (3.0)
Socialization	1 (1.5)
Computer use	1 (1.5)
*Factors (from 19 cases)**	
Gender	3 (15.8)
Aptitude	3 (15.8)
Race/ethnic origin	5 (26.3)

* More than one factor was possible per case.

ment in computer science. Race/ethnic origin, aptitude, and gender were the student-level factors that were examined in the 19 evaluative cases.

Table 5 presents information about the measures used in the 19 evaluative cases in this sample. The most frequent type of measures were questionnaires, existing records (e.g., attendance logs), and standardized tests. Of the 67 measures that were used in these

Table 5
Methodological characteristics: measures

Methodological characteristic	*Frequency* (%)
Measures (out of 67 measures from 19 cases)	
Questionnaire	26 (38.8)
Existing records	15 (22.4)
Standardized test	10 (14.9)
Teacher- or researcher-made instrument	9 (13.4)
Direct observation	4 (6.0)
Grades	2 (3.0)
Focus groups	1 (1.5)
Type of measure (out of 67 measures from 19 cases)	
Quantitative	27 (40.3)
Qualitative	19 (28.4)
Mixed	21 (31.3)

evaluation cases, quantitative measures were used more frequently than qualitative or mixed-methods measures.

A crosstabuluation of the measures and outcomes, which is not presented here because of its large size and sparseness, showed that the measures were correctly matched with outcomes. For example, questionnaires or focus groups, which are considered to be appropriate means of collecting data about attitudes (Frechtling *et al.*, 2002), were used in 16 out of 17 cases in which stakeholder attitudes were examined. In the nine cases where computer science achievement was measured, the most frequently used measures were teacher- or researcher-made tests (5 out of 9), direct observation (2 out of 9), and standardized tests (1 out of 9), all of which are generally considered by the public to be appropriate measures of learning (Frechtling *et al.*, 2002). Only one evaluation used self-report questionnaires, which are generally considered to be unreliable measures of learning (Almstrum *et al.*, 2002, Silka, 1989), to measure computer science achievement.

Table 6 presents the frequencies of the various types of inquiry the evaluators used, the frequencies of experimental designs that were used, and information about study quality when experimental designs were used. Qualitative or experimental/quasi-experimental inquiries were the most common. Of the experimental designs, the pretest-posttest design with a control group was the most frequently used design, followed closely by the one-group posttest-only design.

Table 6

Methodological characteristics: type of inquiry, experimental design, and study quality

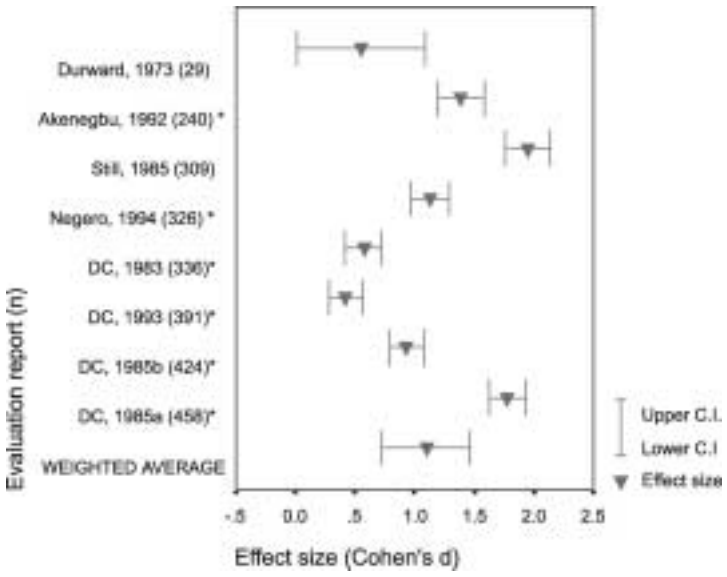
Methodological characteristic	*Frequency* (%)
Type of Inquiry (from 67 investigations)	
Survey research	23 (34.3)
Experimental/quasi-experimental	21 (31.3)
Qualitative	18 (26.9)
Causal-comparative	5 (7.5)
Correlational	0 (0.0)
Classification	0 (0.0)
Experimental design (out of 21 experimental designs)	
Pretest-posttest with control	7 (33.3)
Pretest-posttest without control	4 (19.0)
Posttest with control	3 (14.3)
One-group posttest-only	5 (23.8)
Longitudinal (nondependent measures)	2 (9.5)
Study quality (out of 21 experimental designs)	
High	8 (38.1)
Medium	8 (38.1)
Low	5 (23.8)

Note. Within the 19 evaluations, there were 67 different investigations reported, 21 of which were investigated experimentally.

Evaluation Findings

Fig. 1 shows the effect sizes, their 95% confidence intervals, and the n-sizes of the eight studies that quantitatively investigated the effects of a program on computer science achievement, used an experimental or quasi-experimental design, and gave enough information to calculate Cohen's d. At the bottom of Fig. 1, the weighted, average effect size and its confidence intervals is shown.

As indicated in Table 7, the weighted, average effect size (using a random-effects model) for the eight evaluations on computer science achievement (i.e., teacher- or researcher-made tests or quizzes) was 1.10 with 95% lower and upper confidence intervals of 0.72 and 1.47. Since Q_{total} for the pooled estimate, using a fixed-effects model, indicated heterogeneity of effect sizes across evaluations, a random-effects model was used. Homogeneity of effect sizes was found using a random-effects model, as indicated by a Q_{total} with a p value greater than 0.05 (see Table 7). Since six out of eight effect sizes came from evaluations of the same program (i.e., Nature-Computer Camp), I present the results of a subgroup analysis of Nature-Computer Camp program evaluations and evaluations of programs other than the Nature-Computer Camp. The data in Table 7 indicate that there was neither a statistically significant difference between the groups nor a large difference between the effect sizes of the two groups of evaluations.



Note. Effect sizes in the positive direction indicate an increase in computer science achievement. Evaluations reports followed by an asterisk are Nature-Computer Camp Evaluations. The number in parentheses is the N-size for each evaluation.

Fig. 1. Effect sizes for computer science achievement.

Table 7

Aggregated and disaggregated effect sizes for computer science achievement

Source	*Q*	*df*	*p*	*d*	*Lower CI*	*Upper CI*
Between groups	0.31	1	.58	–	–	–
Within groups	7.59	6	.27	–	–	–
Total	7.91	7	.34	1.10	.72	1.47
Nature-Computer Camp programs	4.43	5	.49	1.04	.60	1.47
Other programs	3.17	1	.08	1.29	.52	2.06

Note. A random effects model was used. The methods of moments random variance component was .29.

Sensitivity Analysis: Random versus Fixed Models

All of the sources of variance presented in Table 7 were statistically significant using a fixed-effects model; however, none of the sources of variance were statistically significant using a random-effects model. This discrepancy is not uncommon, however, because a random-effects model is generally more conservative than its corresponding fixed-effect model when there is a large amount of variance unaccounted for (Hedges, 1994). The results of homogeneity tests presented in Table 7 indicate that the random-effects model, however, had a better fit with these data than the fixed-effects model.

Coding Reliability

Kappa was 1.0 and percent of overall agreement was 100 for *type of inquiry* and *study design*. For *quality of study*, kappa was .62 and overall percent of agreement was 75.

Discussion

Potential Biases in the Reviewed Literature

Assuming that the universe of computer science education evaluations would be proportionally distributed across the globe and be published in a variety of sources, I am inclined to believe that this sample over-represents North American, general-education-centered evaluations (see Table 2). Although the literature search was fairly comprehensive and used international databases that were grounded both in education and computer science, I hypothesize that there are plenty of computer science education program evaluations being done; however, it is primarily North American evaluators who publish their evaluation reports in sources that are highly indexed by academic databases or Internet search engines.

Another possible bias is that six of the eight evaluation cases included in the meta-analysis evaluated the same program: Nature-Computer Camp. In order to investigate this possible source of bias, I conducted subgroup analyses, the results of which are presented

in Table 7. The results showed that there were no practically or statistically significant differences between the outcomes of Nature-Computer Camp evaluations and the outcomes of other computer science education program evaluations.

Program Characteristics

The majority of programs provided various kinds of student instruction targeted at K-12 students. Because of the well-documented pipelining of female students in computer science (Gürer and Camp, 2002), it is surprising that so few programs were geared towards females (see Table 3) and that so few evaluations examined gender interactions (see Table 4).

Methodological Characteristics

Surprisingly, computer science achievement was only the fourth most frequent outcome that was examined (see Table 4). Stakeholders attitudes, enrollment, and achievement in core subjects, which are known correlates of computer science achievement, were outcomes that were all examined more frequently than computer science achievement itself.

The frequency of types of measures that were used (see Table 5) align well with the frequency of outcomes that were examined (see Table 4). Stakeholder attitudes were measured through questionnaires, enrollment was measured through existing records, academic achievement on core subjects was measured through standardized tests, and computer science achievement was measured by teacher- or researcher-made tests.

The fact that the only measure of computer science achievement that reported validity or reliability estimates (Palormo, n.d.) is no longer available and that all other measures of computer science achievement were localized teacher- or researcher-made tests indicates a lack of validated, reliable, standardized measures in computer science education, or a lack of awareness about them. According to Haas and Hassell (1983) there was a need for reliable and validated measure of the effectiveness of computing education over 20 years ago; from the data in this review it appears that this is still the case today. Computer science evaluators might benefit from the work of Cooper, Cassel, Moskal, and Cunningham (2005), who give guidelines for creating outcomes-based measures for computer science education, or from (Fincher and Petre, 2004). Although there is a validated and reliable computer science subject test developed for the Graduate Record Examination (Educational Testing Service, 2004) it is neither available for administration by evaluators nor is it targeted for K-12 students.

The distribution of types of inquiry in this sample of evaluations is similar to the distributions of types of inquiry in educational technology research journals (see Randolph *et al.*, 2005b). There are almost equal frequencies of survey research, qualitative research, and experimental/quasi-experimental research. The most frequently used design (i.e., the pretest-posttest design with controls) in these evaluations is a strong design that controls for many threats to internal validity; the second most frequently used design (i.e., the one-group posttest-only design) is a weak design that is vulnerable to almost all threats

to internal validity (Shadish *et al.*, 2002). Overall, based on study design, most of the experimental or quasi-experimental investigations in the evaluations in this sample were deemed as having high or medium quality (see Table 6).

Evaluation Findings: Computer Science Achievement

Fig. 1 shows the effect sizes for each of the evaluations that used an experimental evaluation design and shows that the average standardized mean difference effect size was 1.10 on standardized or teacher- or researcher-made tests of computer science achievement. The confidence intervals for this estimate indicate that it is plausible that the effect size parameter might be as low as 0.72 or as high as 1.47. The programs' durations were one academic year or semester, except for the Nature-Computer Camp program, which consisted of five one-week sessions.

To aid in the interpretation of that effect size, in Table 8 I present a binomial effect size display (see Rosenthal *et al.*, 2005), which reframes an effect size of 1.10 as a two-by-two table showing how many students, on average, would be expected to improve and not improve as a result of participating in a computer science education program similar to the ones listed in Fig. 1. So, in this case, an effect size of 1.10 is statistically equivalent to about 73 out of 100 students showing improved achievement on teacher- or researcher-made computer science tests or quizzes after participating in a computer science education program. (If the computer science education programs had had no effect, by chance only 50 out 100 students would have been expected to have improved scores.)

It is no surprise that computer science instruction, in general, led to an increase in students' scores on computer science tests; however, this result – that students' scores increased by 1.10 standard deviations – might be useful to evaluators who want to compare the outcomes of the computer science program they are evaluating to the outcomes of the computer science programs reviewed here. For example, for a computer science education program to be as effective as the ones included here, about 73 out of 100 students would need to have improved scores on teacher- or researcher-made measures of computer science achievement.

Table 8

Binomial effect size display for computer science achievement ($d = 1.10$)

	Showed improvement in computer science	*Did not show improvement in computer science*	*Total*
Participated in a computer science education program	72	28	*100*
Did not participate in a computer science education program	28	72	*100*
Total	*100*	*100*	*200*

Study Limitations

Assuming that these data over-represent North-American, general-education-targeted programs in computer science education, the results are best generalized to those types of evaluations. Also, unfortunately, there were too few studies that could be included in the meta-analysis to determine what variations of the computer science education interventions were most effective, under which settings, with which types of participants, and under what research conditions. It was only possible to determine what effect the past computer science programs, in general, had on computer science achievement and whether there was a difference between the Nature-Computer Camp programs and other programs.

Conclusion

In summary, this review reported on an analysis of 29 evaluation reports of computer science education programs. Most of the programs that were evaluated offered direct computer science instruction to general education, high school students in North America. Most frequently, evaluators examined stakeholder attitudes, program enrollment, academic achievement in core courses, and achievement in computer science. The most frequently used measures were questionnaires, existing sources of data, standardized tests, and teacher- or researcher-made tests. The pooled effect size for eight programs that administered teacher- or researcher-made tests of computer science achievement was 1.10, which is statistically equivalent to 73 out of 100 students who participated in the program having made an improvement in computer science achievement.

The implications of this research for the practitioners in and designers of K-12 computer science education programs are that programs that concentrate on student instruction; from short, repeated, off-campus programs that combine computer science education with other subjects (such as Nature-Computer Camp) to programs that are a part of the regular school curriculum (such as the programs reported in (Durward, 1973), or (Still, 1985); are effective in increasing computer science achievement. It is not known whether other types of programs (e.g., programs that concentrate only on teacher-instruction) are effective. Also, what still remains to be seen is what aspects of those programs lead to increased achievement and which do not. Unfortunately, the reporting of the activities involved in those programs is insufficient for a practitioner or program planner to replicate those programs in detail and they are also insufficient for a researcher to investigate which aspects of the program lead to increases in achievement. Another finding of import to computer science education practitioners and program funders is that there have been surprisingly few programs designed to bridge the gender gap in computer science. Only 3 out of the 19 evaluations investigated here were intended to help bridge that gap.

The results presented here can also help identify some of the strengths and weakness of the current methods of computer science education evaluation. One strength is that the

methods of data collection align well with the outcomes being investigated. For example, test or direct observations, rather than self-reports of learning, tend to be used to measure computer science achievement. Another strength is that when experimental designs are used, high-quality designs tend to be used and the experiments are adequately controlled. Also, computer science education evaluators tend to use a wide variety of approaches to investigate their questions, from survey research, to experimental research, to qualitative research.

Concerning the weaknesses, first, although teacher- or researcher-made tests have much ecological validity (i.e., they are not outside of the scope of how students are used to being evaluated), those types of measures usually lack data about their reliability or validity. In all of the evaluations used here, only one used a standardized test that had validity or reliability information; however, that test (Palormo, n. d.) is no longer available. Therefore, there is a dire need for standardized, reliable, and valid measures of K-12 computer science achievement.

Second, because there is such a low degree of enrollment and such a high degree of attrition in postsecondary computer science education, it seems appropriate that student or teacher attitudes about a program was a frequently measured outcome – the rationale being that increased satisfaction with a program will increase enrollment and decrease attrition. However, it is surprising that computer science achievement, which is the obvious goal of most computer science education programs, is only the fourth most frequently measured outcome, behind stakeholder attitudes about the program, program enrollment, and achievement in core courses, in that order. This could also be related to the fact that there are not standardized, reliable, and valid measures of K-12 computer science education achievement.

Third, gender was only examined as a mediating or moderating variable in 3 out of 19 evaluations (see Table 4). This is a surprising finding given the egregious gender gap in the field of computer science (Gürer and Camp, 2002).

Fourth, and finally, the descriptions of program activities in evaluation reports tend to provide too little detail for other practitioners to replicate the program or for researchers to investigate the links between the different kinds of program activities and program outcomes. Understandably evaluation research is primarily meant to answer questions that are of interest to local stakeholders; however, simply reporting program activities in more detail would lead to increased utility of program results by others outside of the program (like practitioners in and evaluators of other similar programs).

Based on the implications mentioned above, two major challenges for computer science evaluation (and research) become clear. Those challenges are (a) to develop standardized, reliable, and valid measures of K-12 computer science achievement that are aligned with Tucker *et. al.*'s (2003) computing curriculum, (b) to investigate whether program activities have differential outcomes based on gender, and (c) to begin to attempt to causally link program activities with program outcomes. It is clear that computer science education works, what is significantly less clear is what aspects of computer science education work best, for whom, and why.

Acknowledgements. Thanks to Elina Hartikainen for her patient coding work.

References

*Akanegbu, B.N. (1992). *Nature-Computer Camp 1991*. Chapter 2 program evaluation report. District of Columbia Public Schools, Washington, D.C., Dept. of Research and Evaluation (ERIC Reproduction Service Document No. ED352345).

*Almstrum, V.L., N. Dale, A. Berglund, M. Granger, J.C. Little, D.M. Miller, M. Petre, P. Schragger and F. Springsteel (1996). Evaluation: turning technology to tool – report of the working group on evaluation. In *Proceedings of the 1st Conference on Integrating Technology into Computer Science Education*. ACM Press, New York, pp. 201–217.

Almstrum, V.L., D. Ginat, O. Hazzan and T. Morely (2002). Import and export to/fromcomputing science education. The case of mathematics education research. In *Proceedings of the 7th Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE '02*. ACM Press, New York, pp. 193–194.

*Atwater, J. (1991). *The Computers Unlimited Magnet Elementary Schools 1990–1991. Formative Evaluation*. Kansas City School District, MO (ERIC Document Reproduction Service No. ED348966).

*Atwater, J. (1992). *Achievement and Enrollment of the Central Computers Unlimited Magnet High School 1990–1991*. Kansas City School District, Mo (ERIC Document Reproduction Service No. ED348961).

*Berney, T.D., and R. Alvarez (1990a, April). *Bilingualism in the Computer Age 1988–89. OREA Evaluation Section Report.* New York City Board of Education, Brooklyn, NY. Office of Research, Evaluation, and Assessment (ERIC Document Reproduction Service No. ED320422).

*Berney, T.D., and R. Alvarez (1990b, August). *Bilingualism in the Computer Age 1989–90. OREA Evaluation Section Report.* New York City Board of Education, Brooklyn, NY. Office of Research, Evaluation, and Assessment (ERIC Document Reproduction Service No. ED337545).

Brennan, R.L., and D.J. Prediger (1981). Coefficient kappa: some uses, misuses, and alternatives. *Educational and Psychological Measurement*, **41**, 687–699.

Breslin, R.D. (1990, May). *Report of the National Science Foundation Workshop on the Dissemination and Transfer of Innovation in Science, Mathematics, and Engineering Education*. National Science Foundation, Washington, DC. Directorate for Education and Human Resources (ERIC Document Reproduction Service No. ED361213).

Bureau of Labor Statistics, U.S. Department of Labor. (2004, May 18). *Occupational Outlook Handbook, 2004–2005 edition: Computer Systems Analysts, Database Administrators, and Computer Scientists.* Retrieved September 2, 2004 from `http://www.bls.gov/oco/ocos042.htm`

*Carabetta, J.R. (1987). The planning and procedures associated with the Western New England College Winter Invitational High School Programming Contest. *SIGCSE Bulletin*, **25**(2), 29–35.

Carvalho, S., and H. White (2004). Theory-based evaluation: the case of social funds. *The American Journal of Evaluation*, **25**, 141–160.

Cooper, H., and L.V. Hedges (1994). Research synthesis as a scientific enterprise. In H. Cooper and L.V. Hedges (Eds.), *The Handbook of Research Synthesis*, Russell Sage Foundation, New York, pp. 3–14.

Cooper, S., L. Cassel, B. Moskal and S. Cunningham (2005). Outcomes-based computer science education. In *Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education*. ACM Press, New York, pp. 260–261.

*Crombie, G., T. Abarbanel and A. Trinner (2002). All-female classes in high school computer science: positive effects in three years of data. *Journal of Educational Computing Research*, **27**, 385–409.

*District of Columbia Public Schools – Division of Quality Assurance (1983, December). *Nature-Computer Camp. Final Evaluation Report 1982–1983*. E.C.I.A. Chapter 2 (ERIC Document Reproduction No. ED241262).

*District of Columbia Public Schools – Division of Quality Assurance (1985a, March). it Nature-Computer Camp. Final Evaluation Report, 1984–1985. E.C.I.A. Chapter 2 (ERIC Document Reproduction No. ED258822).

*District of Columbia Public Schools – Division of Quality Assurance (1985b, September). *Nature-Computer Camp. Final Evaluation Report 1984–1985*. E.C.I.A. Chapter 2 (ERIC Document Reproduction No. ED269219).

*The references preceded by an asterisk are evaluation reports that were included in this methodological review.

*District of Columbia Public Schools – Division of Quality Assurance (1986). *Final Evaluation Report for the Nature-Camp* (ERIC Document Reproduction Service No. ED279542).

*Durward, M.L. (1973). *The Evaluation of Computer-Based Instruction in Vancouver Secondary Schools*. Vancouver Board of Schools Trustees. Dept. of Planning and Evaluation. Report No. RR-73-12 (ERIC Document Reproduction Service No. ED088919).

Educational Testing Service (2004). *Graduate Records Examinations Computer Science Test Practice Book*. Retrieved March 28th from `http://ftp.ets.org/pub/gre/CompSci.pdf`.

Effect Size Calculator (Computer software and manual) (n.d.). Retrieved February 22, 2005 from `http://www.cemcentre.org/ebeuk/research/effectsize/Calculator.htm`.

Fincher, S., and M. Petre (2004). *Computer Science Education Research*. Taylor & Francis, London.

*Fitzgerald, S., and M.L. Hines (1996). The computer science fair: An alternative to the computer programming contest. In *Proceedings of the Twenty-Seventh SIGCSE Technical Symposium on Computer Science Education*. ACM Press, New York, pp. 368–372.

Frechtling, J., H. Frierson, S. Hood and G. Hughes (2002). *The User Friendly Handbook for Project Evaluation*. NSF02-057. NSF, Arlington, VA. Retrieved May 10, 2005, from `http://www.nsf.gov/pubs/2002/nsf02057/nsf02057.pdf`

*Golan, S., and B. Means (1998a, March). *Silicon Valley Challenge 2000: Year 2 Report*. SRI International. Retrieved March 21, 2005 from `http://pblmm.k12.ca.us/sri/ReportsPDFFiles/Year2.pdf`

*Golan, S., and B. Means (1998b, November). *Silicon Valley Challenge 2000: Year 3 Report*. Retrieved March 21, 2005 from `http://pblmm.k12.ca.us/sri/ReportsPDFFiles/Year3.pdf`

Gürer, D., and T. Camp (2002). An ACM-W literature review on women in computing. *ACM SIGCSE Bulletin*, **34**(2), 121–127.

Haas, M., and J. Hassell (1983). A proposal for a measure of program understanding. In *Proceedings of the Fourteenth Annual SIGCSE Technical Symposium of Computer Science Education*. ACM Press, New York, pp. 7–13.

*Haughey, D.G., H.F. McCort, C.N. Russell, L.K. Bremmer, L.E. Lee and Yakimishyn (1980, July). *Evaluation of the Manitoba High School Computer Network*. Manitoba Dept. of Education, Winnipeg. Planning and Research Branch (ERIC Document Reproduction Service No. ED224467).

Hedges, L.V. (1994). Random effects models. In H. Cooper and L.V. Hedges (Eds.), *The Handbook of Research Synthesis*. Russell Sage Foundation, New York, pp. 285–299.

Joint Committee on Standards for Educational Evaluation (1994). *The Program Evaluation Standards*, 2nd ed. Sage, Thousand Oaks, CA.

*Kirkpatrick, N.D., R.J. Tullis, K.S. Sanchez and J. Gonzalez (1991). *HISD Magnet Evaluation: Science, Math, and Computer Enrichment Programs, 1990–91*. Houston Independent School District, TX. Dept. of Research and Evaluation (ERIC Document Reproduction Service No. ED347068).

*Lew, A. (1971, April). *Project Soul*: *Computer Training Program for High School Students from Disadvantaged Areas*. *Part III, The Scientific Programming Course*. University of Southern California (ERIC Document Reproduction Service No. ED180765).

Lipsey, M.W., and D.B. Wilson (2001). *Practical Meta-Analysis*. Sage, Thousand Oaks.

*Means, B., W.R. Penuel, C. Korbak and H. Kantor (2001, January). *Silicon Valley Challenge 2000*: *Longitudinal Case Study Report*. SRI International. Retrieved March 21, 2005 from `http://pblmm.k12.ca.us/sri/ReportsPDFFiles/C2000casestudies.pdf`

Mohr, L.B. (1999). The qualitative method of impact analysis. *American Journal of Program Evaluation*, **20**(1), 69–84.

National Research Council Committee on Information Technology Literacy (1999, May). *Chapter 1*: *Why Know about Information Technology*: *Being Fluent with Information Technology*. National Academy Press, Washington, DC. Retrieved October 12, 2004, from `http://books.nap.edu/html/beingfluent/ch1.html`

*Negero, A. (1994). *Evaluation of the Nature-Computer Camp*: *Summer 1993*. Office of Educational Accountability, Assessment and Information, District of Columbia Public Schools, Washington D.C.

Palormo, J.M. (n.d.). *Computer Programmer Aptitude Battery – Manual*. Science Research Associates, Inc.

*Penuel, B., S. Golan, B. Means and C. Korbak (2000, January). *Silicon Valley Challenge 2000: Year 4 report*. SRI International. Retrieved March 21, 2005 from `http://pblmm.k12.ca.us/sri/ReportsPDFFiles/Year4.pdf`

*Penuel, B., C. Korbak, L. Yarnall and R. Pacpaco (2001, March). *Silicon Valley Challenge 2000: Year 5*

multimedia project report. SRI International. Retrieved March 21, 2005 from http://pblmm.k12.ca.us/sri/ReportsPDFFiles/MMPY5rpt.pdf

*Piña, A.A. (1992, November). Design, development and implementation of a middle school computer applications curriculum. Paper presented at the *Annual Conference of the Arizona Educational Research Association*, Phoenix, AZ, November 5–6, 1992 (ERIC Document Reproduction Service No. ED357731).

Randolph, J.J. (2007a). *Computer science education research at the crossroads: A methodological review of computer science education research: 2000–2005*. Unpublished doctoral dissertation. Utah State University. Retrieved April 11, 2007, from http://www.archive.org/details/randolph_dissertation

Randolph, J.J. (2007b). *Multidisciplinary Methods in Educational Technology Research and Development*. HAMK Press, Hämeenlinna, Finland.

Randolph, J.J., R. Bednarik and N. Myller (2005). A methodological review of the articles published in the proceedings of Koli Calling 2001–2004. In *Proceedings of the 5th Annual Finnish/Baltic Sea Conference on Computer Science Education*. Helsinki University of Technology Press, Finland, pp. 103–109.

Randolph, J.J., R. Bednarik, P. Silander, J. Gonzalez, N. Myller and E. Sutinen (2005). A critical analysis of the research methodologies reported in the full papers of the proceedings of ICALT 2004. In *Proceedings of the Fifth IEEE International Conference on Advanced Learning Technologies*. IEEE Press, Los Almitos, CA, pp. 10–14.

Randolph, J.J., and E. Hartikainen (2005). A review of resources for K-12 computer-science-education program evaluation. *Yhtenäistyvät vai erilaistuvat oppimisen ja koulutuksen polut: Kasvatustieteen päivien 2004 verkkojulkaisu* (in English, *Electronic Publication of the 2004 Finnish Educational Research Days Conference*). University of Joensuu Press, Finland, pp. 183–193.

*Randolph, J.J., M. Virnes, P.J. Eronen and I. Kuurama (n.d.). *Evaluation of Kids' Club 2004–2005: Attitudinal Change*. Evaluation report in progress. University of Joensuu, Finland.

Raudenbush, S.W. (1994). Random effects models. In H. Cooper and L.V. Hedges (Eds.), *The Handbook of Research Synthesis*. Russell Sage Foundation, New York, pp. 301–321.

Rosenthal, R. (1994). Parametric measures of effect size. In H. Cooper and L.V. Hedges (Eds.), *The Handbook of Research Synthesis*. Russell Sage Foundation, New York, pp. 231–260.

Rosenthal, R., R.L. Rosnow and D.B. Rubin (2000). *Contrasts and Effect Sizes in Behavioral Research: A Correlational Approach.* Cambridge University Press, Cambridge.

Scriven, M. (1976). Maximizing the power of causal investigation. In G.V. Glass (Ed.), *Evaluation Studies Review Annual*, vol. 1. Sage, Newbury Park, CA, pp. 101–118.

Shadish, W.R., T.D. Cook and D.T. Campbell (2002). *Experimental and Quasi-Experimental Design for Generalized Causal Inference*. Houghton Mifflin, New York.

Shadish, W.R., and C.K. Haddock (1994). Combining estimates of effect size. In H. Cooper and L.V. Hedges (Eds.), *The Handbook of Research Synthesis*. Russell Sage Foundation, New York, pp. 261–281.

Silka, L. (1989). *Intuitive Judgments of Chance*. Springer-Verlag, New York.

*Seever, M. (1992, April). *Achievement and Enrollment of the Central Computers Unlimited Magnet Middle School 1990–1991*. Kansas City School District, MO (ERIC Document Reproduction Service No. ED348962).

*Still, J.H. (1985, fall). Evaluation of a pilot program for computer literacy in grades K-8. *Florida Journal of Educational Research*, **27**(1), 43–59 (ERIC Document Reproduction Service No. ED323228).

Stufflebeam, D.L. (2001). Evaluation models. *New Directions for Evaluation*, **89**, 7–98.

*Torvinen, S. (2004, August). *Aspects of the Evaluation and Improvement Process in an Online Programming Course. Case: The ViSCoS Program*. Licentiate thesis. University of Joensuu, Finland.

Tucker, A., F. Deek, J. Jones, D. McCowan, C. Stephenson and A. Verno (2003, October 22). *A Model Curriculum for K-12 Computer Science: Final Report of the ACM Education Task Force Computer Science Curriculum Committee*. Retrieved September 7, 2004 from http://www.acm.org/k12/k12final1022.pdf

Valentine, D.W. (2004). CS educational research: A meta-analysis of SIGCSE technical proceedings. In *Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education*. ACM Press, New York, pp. 255–259.

*Walker, E., and S. Rodger (1996). PipeLINK: Connecting women and girls in the computer science pipeline. In *Call of the North, NECC'96. Proceedings of the Annual National Educational Computing Conference*. Minneapolis, MN (ERIC Document Reproduction Service No. ED398896).

Weiss, C.H. (1998). *Evaluation: Methods for Studying Programs and Policies*, 2nd ed. Prentice Hall, Upper Saddle River, NJ.

* Worthington City School District, OH (1983, May). *Progress and Planning Report: K-12 Use of Computers in the Instructional Setting* (ERIC Document Reproduction Service No. ED234983).

J.J. Randolph has a PhD in education research and program evaluation, an MEd in international education, and a certification in educational administration. Currently he is an assistant professor of elementary education at Finlandia University. In the past, Justus Randolph has worked as a program evaluator or researcher for organizations such as the Center for Policy and Program Evaluation, The Worldwide Institute for Research and Evaluation, the National Center for Hearing Assessment and Management, Utah State University, the University of Joensuu, HAMK University of Applied Sciences, and the University of Lapland. His research and evaluation experiences have concerned programs that involve newborn hearing assessment, school improvement, higher education evaluation, technology-enriched playgrounds, educational technology research methods, and computing education. He has developed and taught courses in quantitative and qualitative research methods, evaluation, and scholarly writing. He is the author of the book *Multidisciplinary Methods in Educational Technology Research and Development* and tens of scholarly articles.

Bendrojo lavinimo mokyklos informatikos programų vertinimų metodologinė apžvalga

Justus J. RANDOLPH

Šiame straipsnyje pateikiamos 29 informatikos programų, kurios buvo analizuojamos moksliniuose straipsniuose iki 2005 metų, vertinimų metodologinės apžvalgos, analizė ir rezultatai. Programos apima nuo darželio iki dvyliktos klasės informatikos mokymo kursą. Programas analizuojančių straipsnių turinį sudarė stambių mokslo duomenų bazių paieška, internetas svetainės ir užklausos, pateiktos informatikos mokslo tyrėjams, laikantis griežtų, iš anksto parengtų įtraukimo kriterijų. Šie pranešimai buvo užkoduoti atsižvelgiant į jų demografines, programų ir vertinimo savybes bei vertinimo išvadas. Buvo nustatyta, kad didesnė informatikos programų dalis buvo siūloma Šiaurės Amerikos vidurinės mokyklos mokiniams. Daugiausia dėmesio buvo skiriama tarpininkų nuostatoms (suinteresuotiems asmenims), informatikos programų teisiniams aspektams, moksliniams pasiekimams ir viduriniame, ir aukštajame moksle nagrinėti. Buvo naudojamos anketos, duomenų bazės, įvairūs šaltiniai, standartizuoti testai, mokytojo ar tyrėjo sudaryti testai.