Int. J. Man–Machine Studies (1988) **29**, 613–624

Experimentation in computer science: an empirical view

JEFFREY MITCHELL AND CHARLES WELTY†

Department of Computer Science, University of Maryland, College Park, MD 20742, USA and † Department of Computer Science, University of Southern Maine, Portland, ME 04103, USA

(Received 6 March 1987 and in revised form 28 October 1987)

In many disciplines, scientific inquiry relies heavily on experimentation. Computer science is compared to other scientific disciplines in its use of experimentation by classifying articles in professional journals as experimental or non-experimental. The results of the classification suggest that experiments occur less frequently in computer science than in many other disciplines.

Introduction

There have been many calls for an increase in experimentation in computer science research (Feldman & Sutherland, 1979; McCracken, Denning & Brodin, 1979; Brooks, 1980). Most of these derive from the need for a strong, widely accepted research methodology which can discriminate between competing claims and support deductive theory. This paper reports an empirical study comparing the use of experimentation in computer science to its usage in other disciplines.

Hemple (1966) divides scientific inquiry into the empirical and the non-empirinal sciences. "··· [D]ependence on empirical evidence distinguishes the empirinal sciences from the non-empirical disciplines ··· whose propositions are proved without essential reference to empirical findings." The empirical sciences obtain evidence by induction (primarily observation and experimentation) while the non-empirical sciences depend on deduction (logic and proof). Computer science employs the methods of both the empirical and the non-empirical sciences.

Kaplan (1964) identifies experimentation as "··· a process of observation, to be carried out in a situation especially brought about for that purpose." He indicates that certain cultures, like that of ancient Greece, did not develop into successful modern cultures because they lacked experimentation and, hence, the ability to determine causation, even though they had highly developed powers of reason and observation. Bynum, Browne & Porter (1981) define an experiment as a "··· contrived set of observations, carried out under artificially produced and deliberately controlled, reproducible conditions." Central to these definitions is the notion that significant factors can be separated into independent and dependent variables.

A hypothesis is a formal statement which causally relates independent and dependent variables. An experiment is a controlled empirical test of a hypothesis. Experimental research is important because it provides a widely accepted and

Correspondence: J. Mitchell, Department of Computer Science, University of Maryland, College Park, MD 20742, USA.

0020-7373/88/110613 + 12$03.00/0

uniform method of testing hypotheses. Hypothesis testing produces empirical support for deductive theory and modelling. The synthesis of deduction and empiricism in computer science leads to a body of reliable knowledge about algorithms, data structures, systems, performance, and other important research topics.

In contemporary computer science, experiments are employed to test software systems, benchmark algorithms against various data sets, evaluate programmer behaviour, assess the effectiveness of new programming languages, evaluate human factors considerations in software systems, and measure casual users' ability to interact with systems.

In software engineering, for example, experiments are often employed in the testing phase of the software cycle. In this phase the performance of a system or prototype is verified experimentally against its expected behaviour as outlined in the software specification. Experimentation can demonstrate that a product meets or does not meet its specification (Beizer, 1984). Basili, Selby & Hutchens (1986) provide an overview of experimental research in software engineering.

Another example of the use of experimentation in computer science is in the analysis of the efficiency of algorithms. A great deal of non-empirical work has resulted in understanding bounds on the performance of certain algorithms. Experimentation enhances this work with empirical findings. For example, Wainwright (1985) presents refinements which increase the efficiency of the quicksort algorithm. In support of his modifications, he reports an experiment which compares the efficiency of seven algorithms. On the basis of the experimental findings he concludes that some variations of the quicksort algorithm are more efficient than others.

The non-empirical approach to computer science research yields theories, models, algorithms, and hypotheses which can be tested empircially. The interdependence of non-empirical and empirical research strategies is evident in computer science. For example, Yu and Chang (1984) close a deductive exposition of theories about query processing in a distributed environment with the hope "· · · that large scale experiments will be conducted to verify the usefulness of [their] ideas · · ·".

Computer scientists also use experimentation on physical systems. For example, Franta & Heath (1984) performed experiments on a testbed local area network. The results of their experiments showed that a round-robin protocol scheme actually reduced the fairness of access in a local area network. Their experiments aided in determining appropriate protocols for local area networks.

Much of the experimental work done in computer science involves human factors. For example, highly successful human factors experiments have been performed in the field of database systems. Shneiderman (1980) notes that "Database systems are an important area for experimental studies since the class of users may be an order of magnitude larger than the class of programmers." A widely cited and highly successful experiment led to the development of the database query language SQL (Structured Query Language). The experiment was reported by Reisner, Boyce & Chamberlin (1975). They compared two database query languages, SQUARE and SEQUEL. Subjects wrote queries in each of the languages. The queries were scored for correctness, and Reisner *et al.* found that subjects using SEQUEL were more proficient than those using SQUARE. Based on this experiment and further data analysis SQUARE was abandoned and SEQUEL enhanced. Today, SEQUEL, now known as SQL, is one of the most widely used database query languages (Date, 1986). The study by Reisner *et al.* (1975) demonstrates both the power of experimental analysis in computer science and the direct application of the findings of a computer science experiment to a commercial product. Mitchell & Daste (1987) provide a summary of experimentation in database systems research.

The study described below is an empirical attempt to benchmark experimentation in computer science against other disciplines and to determine whether experimentation is increasing or declining in computer science.

Frequency of experimentation

Rates of publication of experimental studies were determined by examining articles in the most important journals in computer science and various other disciplines. The methodology and findings of this study are described below.

METHODOLOGY

Articles in the most important professional journals from many disciplines were categorized as experimental or non-experimental. In order to conduct an empirical study these journals had to be chosen empirically. This study relied on Garfield's (1984) work published in the Institute for Scientific Information's *Journal Citation Reports* to empircially identify the most important journal in each field.

Garfield presented several rankings of scholarly journals. The rankings were based on how often an article in a journal is likely to be cited and how long after publication articles are likely to continue to be cited (half-life). He constructed his measures from entries in the Institute for Scientific Information Database, (the source of the *Science Citation Index*). In this study Garfield's "Impact Factor" is used as the operational criterion to determine which journal is most important in a discipline.

Impact Factor is a measure of how often an article in a journal is cited on average, controlling for the number of articles published annually in the journal. This measure identifies the professional journal which publishes the most frequently cited articles. Thus the journal which is of greatest importance to researchers in a discipline has the highest impact factor. *Computing Surveys* is the journal with the highest impact factor in computer science. The most important journals in each discipline were selected on the basis of this empirical criterion. While some may feel that a particular journal better represents a field or subfield, this study depends on the idea that the most important journal in a discipline as a whole is the most often cited journal.

Some of the journals selected include the word "review" in their titles. In many cases these journals really publish first reports of original research. In some cases the journals serve as a collection of papers which overview the research in a field. In either case it is expected that the content of the journal reflects the research work going on in the field.

Garfield (1984) presented an impact factor rank ordering of journals broken down by discipline (category). Table 1 displays the journal with the highest impact factor in the subset of Garfield's disciplines which are examined in this study.

TABLE 1

Journals with the highest impact factor score in each discipline. The column labelled CODE contains the labels used in the figures in this article. The column labelled DISCIPLINE contains Garfield's (1984) categorization of field names. The column labelled JOURNAL contains the name of the journal with the highest impact factor in the associated discipline

Code	Discipline	Journal
AG	Agriculture	*Journal of Agriculture & Food Chemistry*†
BIO	Biology	*Quarterly Review of Biology*
BS	Business*	*Administration Science Quarterly*
CH	Chemistry	*Chemical Reviews*
COM	Communication*	*Human Communication Research*
CS	Computer applications & cybernetics	*Computing Surveys*
ED	Education & educational research	*Review of Education Research*
CE	Engineering, civil	*Journal of Environmental Engineering*†
EE	Engineering, electrical & electronic	*Proceedings of the IEEE*
LG	Language & linguistics*	*Journal of Memory and Language*
HST	History	*American Historical Review*
IS	Information science & library science*	*Journal of ASIS*
BS	Management*	*Administrative Science Quarterly*
MA	Mathematics, applied	*SIAM Review*
MD	Medicine, general & internal	*New England Journal of Medicine*†
PHY	Physics, applied	*Applied Physics Letters*†
PS	Political science*	*American Political Science Review*
PSY	Psychology	*Psychological Review*

Source: Garfield, Eugene. "Journal citation reports: a bibliometric analysis of science journals in the ISI database," in the 1984 of *Science Citation Index*, and the *Social Science Citation Index*.

* Discipline names followed by an asterisk (*) were taken from the *Social Science Citation Index*, all others were taken from the *Science Citation Index*.

† A cross (†) after a journal name indicates that a random sample of the articles in the journal was evaluated instead of evaluating all the articles in the journal.

Articles published in 1985 in each of the journals listed in Table 1 were examined. Each article was evaluated and categorized as experimental or non-experimental. An article was considered experimental if it reported a causal study in which the researcher either exercised control over the independent variables while measuring dependent variables or cited at least one experimental report.

Article classification was done by the authors of this study. An article was classified as experimental if it included some theory or hypothesis that could be transformed into a formal statement relating independent and dependent variables. The experiment should have been replicateable (Considine, 1983) and may have employed standard tools (Campbell & Stanley, 1966). Only rarely were all these elements explicitly stated in an article reporting an experiment. Since it is difficult to measure objectively how important the cited articles are to the substance of a paper, papers that cited any experimental studies were classified as experimental. This method of classification conforms with the objective of measuring the *use* of experimentation, though it may inflate the usage measure slightly.

Articles which reported experiments were most commonly the initial report of an experiment. They included detailed descriptions of the methods and apparatus used in the experiment, as well as a discussion of the findings and a graphical or tabular presentation of the data. An example of an article which reports the results of an experiment is Vessey & Weber's (1986) report of an experimental comparison of three tools for the structural decomposition of a process.

Articles which cite reports of experiments often concentrate on a synthesis of the results of one or more experiments. These articles are more abstract and deal in model building or basic theory. They may include descriptions of new systems based on experimental findings. Articles citing reports of experiments sometimes propose new research questions or hypotheses. Jarke & Vassiliou (1985) provide an example of the type of article which cites and employs experimental research in synthesizing a new model. They build a "framework for choosing a database query language" based heavily on the results of experimental studies.

Journal articles which did not report or cite experimental research were classified as non-experimental. Articles in this category were case studies, reports of surveys, descriptions of new systems, presentations of theorems and proofs, reports of non-experimental field tests, etc. Only journal articles were considered; correspondence, viewpoints, book reviews, conference reports, news stories, and calls for papers were not included.

FINDINGS

Figure 1 shows a bar graph representing the percentage of all articles published in the leading journal in each discipline in order of amount of experimentation. The

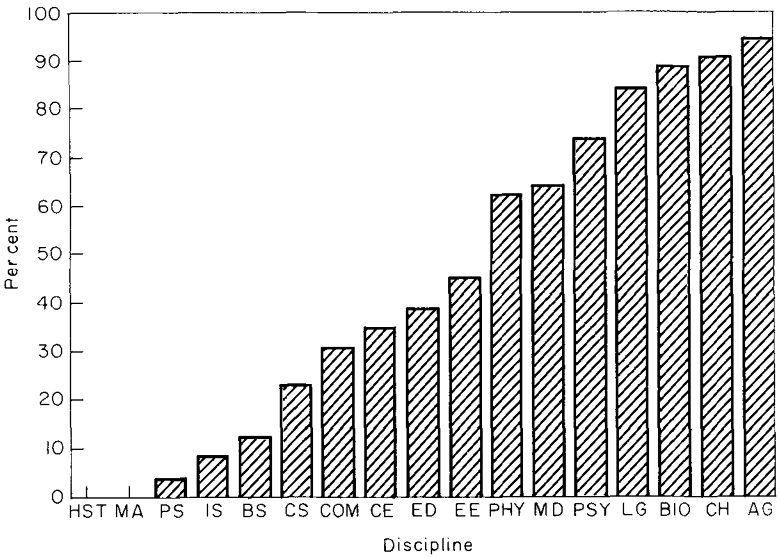

FIG. 1. Number of articles reporting or citing experiments represented as a percentage of all articles published in the most important journal in each of the selected disciplines in 1985 (see Table 1 for meaning of labels).

height of a bar reflects the percentage of articles published in 1985 which either report or cite experiments.

It is clear from Fig. 1 that computer science is at the low end of the spectrum, indicating that reports of experiments occur less frequently in computer science than in many other disciplines. In fact, only some of the social sciences and mathematics rank lower than computer science in their use of experimentation. Disciplines on which computer science is highly dependent, such as physics and electrical engineering, publish from 50% to 100% more experimental research than does computer science. It appears fair to conclude that computer scientists are doing (or at least publishing) less research that involves experimentation than are researchers in related disciplines.

In Fig. 1 a noticeable jump in the level of experimentation occurs between the disciplines of electrical engineering and physics. The natural break at this point divides the disciplines into those which publish more than 60% experimental articles and those disciplines which publish less than 46% experimental articles. None of the disciplines surveyed fall in the 46% to 60% range. The upper group of disciplines consists of disciplines in the natural, physical, and behavioural sciences, while the lower group consists of disciplines in the social sciences, humanities, engineering, mathematics, and computer science. The difference between these two groups is statistically significant ($F = 61{\cdot}957$, $P \leq 0{\cdot}0001$).

Discussion

The experimental emphasis of many disciplines suggests the importance of the experiment as a basic research tool. It is evident that experimentation has not penetrated computer science to the degree it has other disciplines. Feldman & Sutherland (1979) and McCracken *et al.* (1979) indicated that the crisis in experimental computer science was due primarily to a lack of funding. They use the term experimental to apply both to the construction of new systems and the measurement and testing of algorithms and systems. These authors called for more funds for experimental computer science. Feldman *et al.* (1979) also found that the professional papers in non-empirical areas required less time to produce than did empirical papers, leading scholars (in search of tenure) into non-empirical subfields. Lack of training in empirical methods may also play a role. In practice, the design of a new system is a complex task. Testing such a system can be at least as difficult. Resources, motivation, and time should be available for both system design and experimental testing. Further research is necessary to determine the appropriate level of experimental research in computer science, but it seems unlikely that the current level of experimentation is adequate given its higher usage in other scientific fields similar to computer science.

Publication of experiments

A survey of computer science journals identified those which publish the results of experiments on a regular basis. These journals are listed in Table 2. Many journals were initially inspected and then rejected either because they did not publish a significant number of experimental studies or because their primary interest was not

TABLE 2
Names and codes of the computer science journals evaluated in this study. Code is the label used for journal names on the figures. The Journal of the ACM *was included for comparison purposes*

Code	Journal
CACM	*Communications of the ACM*
CS	*Computing Surveys*
HCI	*Human Computer Interaction*
TSE	*IEEE Transactions on Software Engineering*
TSMC	*IEEE Transactions on Systems, Man, and Cybernetics*
IJMMS	*International Journal of Man–Machine Studies*
JACM	*Journal of the ACM*
SIGGHI	*SIGCHI Bulletin* (Computers and human interaction)

computer science—for example, fewer than 40% of the articles in *Human Factors* (in 1985) dealt with topics directly related to computer science. The articles published in each of the selected journals were classified as experimental or non-experimental. The percentage of experimental articles published in 1985 was computed. The results of this classification are presented in Figure 2.

Figure 2 shows that there is a wide range in the proportion of experimental studies published in computer science journals. Proportionally more of the articles published in the *International Journal of Man–Machine Studies* report or cite experiments than do articles in any of the other journals. While computer science journals published as much as 40% experimental work, on average less than 20% of the articles published in the selected computer science journals reported or cited experimental work in 1985.

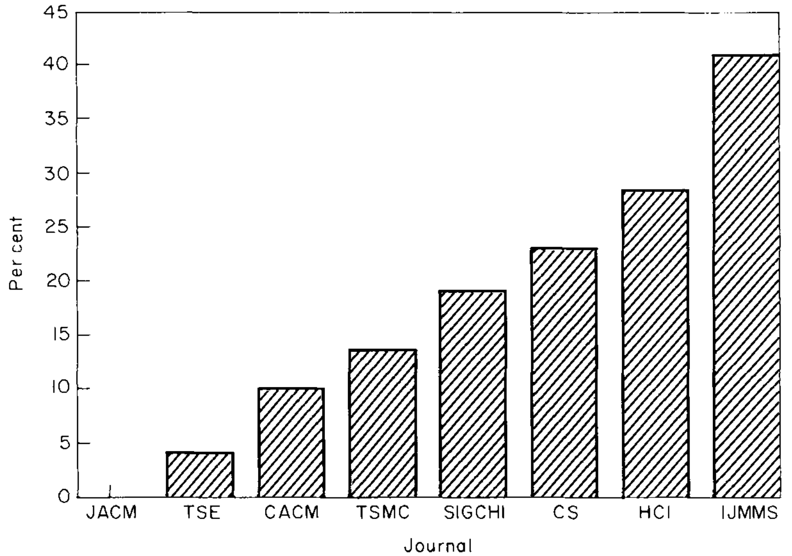

FIG. 2. Number of articles reporting or citing experiments represented as a percentage of all articles published in various computer science journals in 1985 (see Table 2 for meaning of labels).

The highest proportion of reports of computer science experiments in 1985 were in journals closely associated with the human factors subfield. Articles in the *International Journal of Man–Machine Studies* tended to be concise descriptions of human factors experiments. The second most experimental journal, *Human Computer Interaction,* also concentrates on studies in the human factors area. The other journals reported a variety of applications of experimentation to a cross section of the computer science subfields including human factors.

Evidence of increasing experimentation

Time series trends were established for articles published in the past decade in four important computer science journals in order to determine whether experimentation is increasing or declining in computer science. Articles published between January 1, 1976 and December 31, 1985 in *Communications of the ACM, Computing Surveys, IEEE Transactions on Software Engineering,* and *International Journal of Man–Machine Studies* were evaluated and categorized as experimental or non-experimental. These journals were selected from the computer science journals listed in Table 2 because they were published each year in the last decade. The four journals showed two different trends which are presented in Fig. 3.

Figure 3 shows the percentage of articles which report or cite experiments for each of the last ten years. The upper trend line (squares) represents the proportion of experimental research published in *International Journal of Man–Machine Studies* during the decade. The lower line (triangles) represents the average proportion of experiments published in *Computing Surveys, Communications of the ACM* and

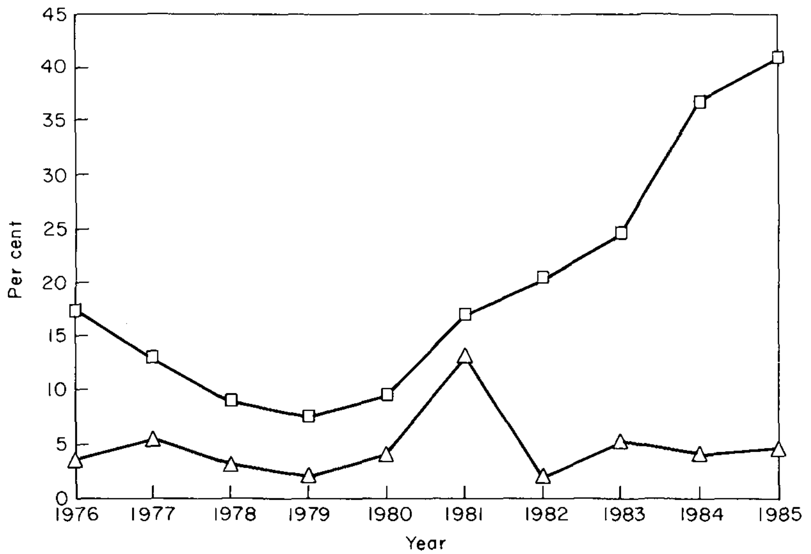

FIG. 3. Number of articles reporting or citing experiments represented as a percentage of all articles published in the *International Journal of Man–Machine Studies* (squares) each year from 1976 to 1985, compared to the average percentage (triangles) of articles reporting or citing experiments in *Communications of the ACM, Computing Surveys,* and *IEEE Transactions on Software Engineering* during the same period.

IEEE Transactions on Software Engineering between 1976 and 1985. The data for these three mainstream computer science journals were combined because their individual trends are very similar. The figure shows an annual increase in the percentage of experimental articles published in *International Journal of Man–Machine Studies* over the last 7 years in contrast to the trend represented by *Computing Surveys, Communications of the ACM* and *IEEE Transactions on Software Engineering.* These mainstream journals generally publish less than 10% experimental articles, though an occasional issue is devoted to experimental work. For example, in 1981 *Computing Surveys* published a special issue on human factors which included several experimental studies.

Figure 3 indicates that while the use of experimentation is growing in the human factors area, mainstream computer science still publishes approximately the same proportion of experimental studies as they have for each of the past 10 years. Publication of experimental work in *International Journal of Man–Machine Studies* grew at a rate of almost 6% ($b = 5{\cdot}82$, $r = 0{\cdot}98$) each year over the past 7 years, while percentage of experimental articles published in the other journals remained essentially constant ($b = 0{\cdot}55$, $r = 0{\cdot}42$) during the decade.

It appears that the crisis in experimental computer science addressed by Feldman & Sutherland (1979) and McCracken *et al.* (1979) is still in effect. The publication of experimental human factors articles hit a low point in 1979 and has since been on a consistent upswing. The rate of publication of experimental articles in the mainstream publications remained low and erratic throughout the decade.

General trends in experimentation

The seven-year trends seen in Fig. 3 suggest that experimentation is becoming more frequent in the human factors area while there has been no real change in mainstream computer science. In order to gain another perspective on this data and to determine which of these two trends better reflects the computer science discipline, it was decided that all computer science articles from 1977 to 1984 should be categorized. Owing to the impossibility of reading every article, the keyword index of *ACM Guide to Computing Literature* was examined.

The *ACM Guide to Computing Literature* contains indices (author, keyword, category, etc.), which reference entries in each volume of *Computing Reviews. Computing Reviews* is composed of short reviews of articles published in the most important professional journals in the discipline of computer science. The Keyword Index is an alphabetical listing of keywords out of context (KWOC) and keywords from article titles. The Keyword Indices in the 1977 to 1984 volumes of *ACM Guide to Computing Literature* were examined, and the number of entries which included keywords referring to experimentation were counted. The number of titles were counted for each of the following keywords: experimental, experimentally, experimentalists, experimentation, experimenter, experimenters, and experimenting. This count yielded a time series trend for the use of keywords relating to experiments (see Fig. 4).

Figure 4 presents the trend line from 1977 to 1984 of the number of titles which refer to experimental work in the Keyword Index of *ACM Guide to Computing Literature.* Reasons for the dramatic increase in the use of experimental keywords in

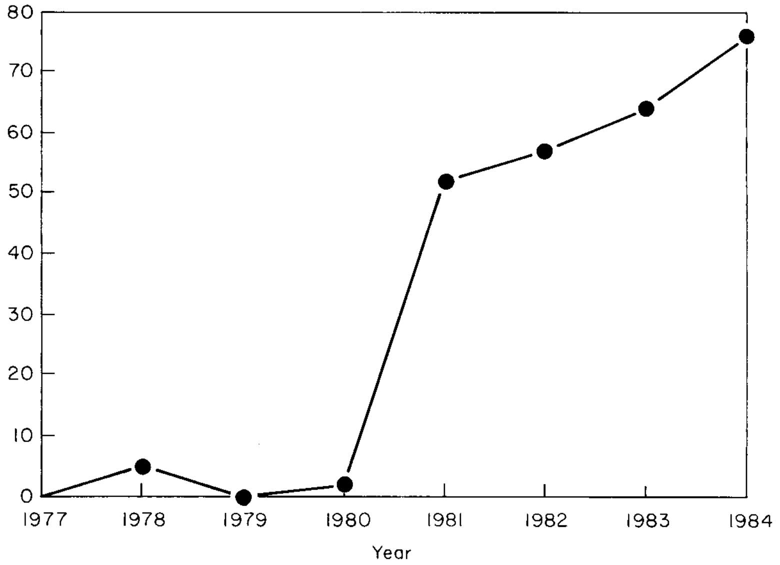

FIG. 4. Number of titles referring to experimentation in the Keyword Index of the *ACM Guide to Computing Literature* each year from 1977 to 1984.

titles between 1980 and 1981 are unknown. The *ACM Guide to Computing Literature* did not report any change in its policy or its method of generating the Keyword Index.

It is conceivable that researchers began to focus more on experimentation after the the Feldman and Sutherland (1979) report recommended more support for computer science experimentation, but statistical controls suggest otherwise.

The post-1980 trend in the *ACM Guide to Computing Literature* shows a growth rate of about eight titles per year ($b = 7{\cdot}9$, $r = 0{\cdot}98$). When control is instituted for the number of titles in each index each year the slight annual increase in the number of experimental titles disappears. This means that the increase in the number of experimental entries is an artifact of the tendency to index more articles in the *ACM Guide to Computing Literature* each year than in the year before. These findings suggest that while experimentation is increasing in the human factors area, and some experimentation is done generally, on the whole the level of experimentation is not increasing in computer science.

The Keyword Index is composed of words which appear in the titles of journal articles. Articles which reported experiments but which did not include one of the experimental terms in their title were not classified as experimental in the data presented in Fig. 4. Hence Fig. 4 probably underestimates the use of experimentation in computer science. This underestimation should be constant from year to year. The time series analysis is included here as an alternative view of the data presented in Fig. 3. The data in Fig. 3 were derived by examining the actual content of articles in order to establish the trend. Except for the increasing trend towards human factors experimentation, both figures show little change in the levels of experimentation over the past decade.

Conclusion

The empirical finding that less experimentation is employed in computer science than in other "hard" sciences suggests an important question: what level of experimentation is appropriate in computer science?

It is probably impossible to determine the optimal mix of experimental and non-experimental research. Our finding that experimental research is lacking in computer science is supported in the literature (Feldman & Sutherland, 1979; McCracken *et al.*, 1979; Brooks, 1980; Basili *et al.*, 1986). The 1985 volume of *IEEE Transactions on Software Engineering* was rescanned to see which papers could be enhanced with an experimental component, either in the paper itself or in a subsequent paper. *IEEE Transactions on Software Engineering* was chosen because it is a mainstream journal with a low rate of publication of experimental articles.

Some of the authors who published non-experimental papers in the 1985 *IEEE Transactions on Software Engineering* recognized the need for experimentation explicitly. In one case the authors state that, "[m]uch experimentation will be needed in combining these quality attributes into a single measure." (Yau & Callofello, 1985). Other authors make claims that are not supported by experimentation, but which are amenable to experimentation. For example: "[s]ince both the query facility and the host language are very simple, programmers do not have any difficulty using them." (Roussopoulos & Yeh, 1985). Most commonly no such statements are made, but in some cases experimentation would be appropriate.

This subjective rescanning of the 1985 *IEEE Transactions on Software Engineering* found that approximately 25% of the papers published could have contained, or led to, experimental work; some may have. If this experimental work were done and found publishable by *IEEE Transactions on Software Engineering* it would seem that 25% of *IEEE Transactions on Software Engineering* papers would be experimental instead of the 4% shown earlier (see Fig. 2).

The problem has several components: (1) the time required to do experimentation precludes authors in search of publication (and tenure) from performing experiments; (2) journals do not necessarily require experimental confirmation of results, even where it may be appropriate in an article; (3) experimental follow-up articles may not be as publishable as the original article; and (4) researchers in computer science may not have sufficient training in experimentation.

With the broad applications experimentation can have in computer science, and the strong empirical support experiments give deductive theorists, it is surprising that more experimentation is not done in computer science.

Suggestions made by David Hill and the referees have enhanced this paper.

References

Basili, V. R., Selby, R. W. & Hutchens, D. H. (1986). Experimentation in software engineering. *IEEE Transactions on Software Engineering*. **SE-12,** 733–743.

Beizer, B. (1984). Software performance. In C. R. Vick & C. V. Ramamoorthy, *Handbook of Software Engineering*. New York: Van Nostrand Reinhold.

Brooks, R. (1980). Studying programmer behavior experimentally: the problem of proper methodology. *Communications of the ACM*. **23,** 207–213.

BYNUM, W. F., BROWNE, E. J. & PORTER, R. (Eds) (1981). *Dictionary of the History of Science*. Princeton: Princeton University Press.

CAMPBELL, D. T. & STANLEY, J. C. (1966). *Experimental and Quasi-Experimental Designs for Research*. Chicago: Rand McNally College Publishing.

CONSIDINE, D. M. (Ed.) (1983). *Van Nostrand's Scientific Encyclopedia* (6th Edition, Volume 1). New York: Van Nostrand Reinhold.

DATE, C. J. (1986). *An Introduction to Database Systems* (4th Edition). Reading, MS: Addison-Wesley.

FELDMAN, J. A. & SUTHERLAND, W. R. (1979). Rejuvenating experimental computer science. *Communications of the ACM*, **22**, 497–502.

FRANTA, W. R. & HEATH, J. R. (1984). Measurement and analysis of HYPER channel networks. *IEEE Transactions on Computers*, **C-33**, 249–260.

GARFIELD, E. (1984). Journal citation reports: a bibliometric analysis of science journals in the ISI data base. In GARFIELD, E. (Ed.) *Sciences Citation Index*. Philadelphia: Institute for Scientific Information.

HANSON, N. R. (1971). *Observation and Experimentation: a Guide to Philosophy of Science*. London: George Allen and Unwin.

HEMPLE, C. G. (1966). *Philosophy of Natural Science*. Englewood Cliffs: Prentice-Hall.

JARKE, M. & VASSILIOU, Y. (1985). A framework for choosing a database query language. *Computing Surveys*. **17**, 313–340.

KAPLAN, A. (1964). *The Conduct of Inquiry*. San Francisco: Chandler Publishing.

MCCRACKEN, D. D., DENNING, P. J. & BRODIN, D. H. (1979). ACM executive committee position on the crisis in experimental computer science. *Communications of the ACM*. **22**, 503–504.

MITCHELL, JEFFREY & DASTE, JACQUELYN, (1987). Human factors and database management systems research. *Proceedings of the ACM South Central Regional Conference*, Lafayette, Louisiana.

REISNER, P., BOYCE, R. & CHAMBERLIN, D. (1975). Human factors evaluation of two database query languages—Square and Sequel. *Proceedings of the National Computer Conference*, **44**, 447–452.

ROUSSOPOULOS, NICOLAS & YEH, RAYMOND T. SEES—A Software Testing Environment Support System. *IEEE Transactions on Software Engineering*, **SE-11**, 355–366.

SHNEIDERMAN, B. (1980). *Software Psychology: Human Factors in Computer and Information Systems*. Cambridge: Winthrop.

VESSEY, I. & WEBER, R. (1986). Structured tools and conditional logic: an empirical investigation. *Communications of the ACM*, **29**, 48–57.

WAINWRIGHT, R. L. (1985). A class of sorting algorithms based on quicksort. *Communications of the ACM*, **28**, 396–402.

YAU, STEPHEN S. & CALLOFELLO, JAMES S. (1985). Design stability measures for software maintenance. *IEEE Transactions on Software Engineering*, **SE-11**, 849–855.

YU, C. T. & CHANG, C. C. (1984). Distributed query processing. *Computing Surveys*, **16**, 403–430.