Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®

The Journal of Systems and Software 70 (2004) 165–176

The Journal of Systems and Software

www.elsevier.com/locate/jss

Research in computer science: an empirical study

V. Ramesh [a,*], Robert L. Glass [b,1], Iris Vessey [a,2]

[a] Department of Information Systems, Kelley School of Business, Indiana University, Bloomington, IN 47405, USA
[b] Computing Trends, 1416 Sare Rd., Bloomington, IN 47401, USA

Received 20 May 2002; received in revised form 7 August 2002; accepted 3 November 2002

Abstract

In this paper, we examine the state of computer science (CS) research from the point of view of the following research questions:

1. What topics do CS researchers address?
2. What research approaches do CS researchers use?
3. What research methods do CS researchers use?
4. On what reference disciplines does CS research depend?
5. At what levels of analysis do CS researchers conduct research?

To answer these questions, we examined 628 papers published between 1995 and 1999 in 13 leading research journals in the CS field. Our results suggest that while CS research examines a variety of technical topics it is relatively focused in terms of the level at which research is conducted as well as the research techniques used. Further, CS research seldom relies on work outside the discipline for its theoretical foundations. We present our findings as an evaluation of the state of current research and as groundwork for future CS research efforts.

Keywords: Topic = Computing research; Research Approach = Evaluative-Other; Research Method = Literature analysis; Reference Discipline = Not applicable; Level of Analysis = Profession

1. Introduction

Computer science is a well-established discipline that is represented in almost all institutions of higher education. [3] As part of their faculty responsibilities, computer scientists conduct research in several different areas, such as artificial intelligence, databases, distributed systems, etc. Research is published in journals dedicated to fostering research in those specific areas. Thus, it is not surprising that most papers that examine the nature of research within computer science tend to focus on specific areas of computer science (see, for example, Gruman, 1990; Rice, 1995; Wegner and Doyle, 1996; Gallopoulous and Sameh, 1997) or even subareas, for example, heterogeneous databases (Sheth and Larson, 1990) or data modeling (Hull and King, 1987), rather than on the discipline as a whole. From a broader perspective, we also find articles that address the nature of computer science research at a country level, e.g., Ramamritham (1997) on India and Estivili-Castro (1995) on Mexico. With the exception of studies by Glass (1995) and Tichy et al. (1995), however, very few studies have examined the nature of research in the field as *a whole*. And even these studies have a relatively narrow focus in that they examine only commonly researched topics and/or the research methods used.

Our objective in this study is to provide a detailed characterization of computer science research, along the

* Corresponding author. Tel.: +1-812-855-2641; fax: +1-812-855-4985.
E-mail addresses: venkat@indiana.edu (V. Ramesh), rlglass@acm.org (R.L. Glass), ivessey@indiana.edu (I. Vessey).
[1] Tel.: +1-812-337-8047.
[2] Tel.: +1-812-855-3485.
[3] Throughout this paper, CS is an abbreviation for Computer Science unless the context explicitly states otherwise (e.g., where CS is used to represent Computer System (in Unit/Level of Analysis) or Case Study (under Research Methods)).

0164-1212/$ - see front matter © 2003 Elsevier Inc. All rights reserved.
doi:10.1016/S0164-1212(03)00015-3

dimensions identified above, by examining articles published in major computer science journals from 1995–1999. Our interest in this study goes beyond topic and research methods and includes other ways of characterizing research such as research approach, which identifies the way in which a research study is conducted, the level of analysis, which identifies the object that is studied, and reference discipline, which identifies the theoretical foundation of the research.

2. The current study

This section describes the classification scheme used to characterize CS research in this study. It also presents details regarding the journals examined and the classification process.

2.1. Classification scheme

Given that our objective was to characterize research in computer science, our first task was to identify a classification scheme that would enable us to capture the richness of CS research. We found that traditional classification schemes such as the ACM computing classification scheme (ACM CCS, 1998) characterize research only along one dimension, i.e., topic. Classification schemes in related disciplines such as information systems, e.g., ISRL categories (Barki et al., 1988), also tend to focus on topic with research method as a secondary consideration. However, researchers often wish to know how the findings of studies of interest were obtained (i.e., the research approach and research method used). In addition, the level at which a study is conducted is also of interest to researchers; a study might, for example, focus on an abstract concept (AC) such as a data model, or a computing element (CE) such as an algorithm, or it might focus on a system, a project, or an organization. Finally, the origin of the study's theoretical base, the reference discipline, is also interesting to researchers because it may suggest richer conceptualizations of the phenomena of interest.

Because none of the existing classification schemes was sufficiently rich in the desired dimensions, we developed a multi-faceted classification system that characterizes research along the five dimensions outlined above. The classification system was comprehensive in a further way; it was developed to describe research in three computing-related disciplines: computer science, software engineering, and information systems (see Vessey et al., 2001). Thus, some of the categories in our scheme may be less relevant to mainstream CS research. For brevity, the classification system is presented with the results of our study in Tables 3–7. Below, we present a brief description of how the classification system was developed.

2.1.1. Classifying topic

To ensure that our list of topics was sufficiently broad to include all areas of computing research, we used several sources of topics from the general discipline of computing, viz., the ACM computing reviews classification scheme (ACM CCS, 1998), the categories in Barki et al. (1988), and the topic areas identified by Glass (1992). In particular, we used the classification scheme proposed by Glass (1992) as the starting point for arriving at the high-level categories shown in Table 3 because its stated objective of presenting a comprehensive set of topics in the fields of computer science, software engineering, and information systems best fit our completeness criterion.

The overall classification scheme, which is shown in Table 3, divides the topics of the computing research field into several major categories:

- Problem-solving concepts
- Computer concepts
- Systems/software concepts
- Data/information concepts
- Problem-domain-specific concepts
- Systems/software management concepts
- Organizational concepts
- Societal concepts
- Disciplinary issues

Each of these categories, is further divided into several subordinate categories.

2.1.2. Classifying research approach

We also categorized the research techniques used. We divided those techniques into research approach, the overall approach undertaken in performing the research, and research method, the more detailed techniques used. In this section, we discuss research approach.

Surprisingly, there is very little information in the field to aid in the classification of research techniques. We used Morrison and George's (1995) categorization of research approaches as a starting point for determining the research approaches to be examined in this study. Based on an analysis of articles in both software engineering and information systems between 1986 and 1991, they characterized the four major research approaches as descriptive, developmental, formulative, and evaluative. These correspond roughly to the scientific method categories of: observe, formulate, and evaluate (Glass, 1995). We included developmental in the descriptive category because such research primarily involved describing systems.

We further subdivided these categories to reflect a rich set of research approaches. Table 4 shows the categories used to classify research approach in this study. The descriptive approach has three subcategories. Subcategory descriptive-system (DS) is based on Morrison and George's descriptive category and is used to capture papers whose primary focus is describing a system. Descriptive-other (DO) was added to capture those papers that used a descriptive approach for describing something other than a system, for example, an opinion piece. We added descriptive-review (DR) as a subcategory into which we categorized papers whose primary content was a review of the literature.

The formulative research approach was subcategorized into a rich set of possible entities being formulated, including processes/procedures/methods/algorithms (all categorized under FP), and frameworks and guidelines/standards (FF and FG, respectively). In all, there are six subcategories of the formulative research approach.

Our evaluative categories are based on the three alternative "evaluative" epistemologies identified by Orlikowski and Baroudi (1991): positivist (evaluative-deductive in our system), interpretive (evaluative-interpretive), and critical (evaluative-critical). We added an "Other" category here to characterize those papers that have an evaluative component but that did not use any of the three approaches identified above. For example, we classified papers that used opinion surveys to gather data (as opposed to questionnaires that used established research instruments) under evaluative-other.

2.1.3. Classifying research method

Research method describes the specific technique used in a given study. While the choice of research approach narrows the set of possible applicable research methods, there is typically a one-to-many relationship between a given research approach and method. Hence, in addition to research approach, we also coded the detailed technique used by a study.

Unlike research approach, where there were few candidate categories from which to choose, in the case of research method, there were numerous classifications from which to choose. Recall that, while the objective of this paper is to characterize the nature of research in computer science, the categories and taxonomies used in this paper were intended to cover the whole of the computing field, including computer science.

Arguably, the computing discipline most concerned with research method is Information Systems where many prior publications have identified a number of commonly used methods (see, for example, Alavi and Carlson, 1992; Farhoomand and Drury, 2000). These articles identify, for example, laboratory experiments (using human subjects), field studies, case studies, and field experiment. Several other research methods have also been identified; for example, conceptual analysis (or conceptual study), literature review (Lai and Mahapatra, 1997), instrument development (Alavi and Carlson, 1992), and exploratory survey (Cheon et al., 1993).

Some studies have examined research methods specific to a software engineering context. Both Zelkowitz and Wallace (1997) and Harrison and Wells (2000) proposed a number of research methods similar to those identified in the information systems studies cited above. In addition, we are aware of two papers that address research methods in both computer science and software engineering. Glass (1995), for example, suggested a fairly simplistic approach, derived from prior literature, which categorized methods as scientific, engineering, empirical, and analytical. Tichy et al. (1995) conducted a more general survey of articles in CS journal and conferences and found that CS research was lacking in its use of experimental methods.

To assist in the categorization of the CS component of computing research, we added the following categories to the above list: conceptual analysis/mathematical (CA/M) and mathematical proof to facilitate the classification of papers that utilize mathematical techniques; Simulation, to allow categorization of papers that utilized simulation as their primary research method; and concept implementation for papers whose prime research method was to demonstrate proof of a concept by building a prototype system. We also added the category laboratory experiment (software) to characterize those papers that, for example, compare the performance of a newly-proposed system with other (existing) systems. It is important to note that not all of the research methods included in Table 5 are appropriate for computer science research.

2.1.4. Classifying unit/level of analysis

Level of analysis refers to the notion that research work may be conducted at one or more of several levels; for example, at a high level, the research may be technical or behavioral in nature. Example of technical research would be focused on the computing system (CS), computing element (CE, representing a program, component, algorithm, or object) or abstract concept level (AC, e.g., graph-based representations). An example of behavioral research is the Watts Humphrey work on Team Software Process (http://www.sei.cmu.edu/tsp/tsp.html), which would be categorized as GP for Group/Team, and his Personal Software Process work, which would be categorized as IN for individual (http://www.sei.cmu.edu/tsp/psp.html). Some research work is done at the level of the profession (PRO), of which this paper is an example, as are those papers referenced in the introduction that address CS research in a particular country, while others may be conducted within an enterprise at the organizational (OC) level. Table 6 presents the levels of analysis used in this study.

2.1.5. Classifying reference discipline

By reference discipline, we mean any discipline outside the CS field that CS researchers have relied upon for theory and/or concepts. Generally, a reference discipline is one that provides an important basis, such as theory, for the research work being conducted. Various researchers have characterized the reference disciplines used in research (see, for example, Swanson and Ramiller, 1993; Westin et al., 1994). Swanson and Ramiller (1993) identified computer science, management science, and cognitive science, organizational science, and economics as four key reference disciplines for information systems. Barki et al. (1988) also include behavioral science, organizational theory, management theory, language theories, artificial intelligence, ergonomics, political science, and psychology, while Westin et al. (1994) further identified mathematics/statistics and engineering. The reference discipline categories presented in Table 7 represent a comprehensive aggregation of the categories addressed in prior research, i.e., some of our categories subsumed one or more of the categories outlined above. The management category, for example, subsumes organizational theory and management theory. Similarly, artificial intelligence is subsumed within computer science.

2.2. Journal and article selection

For our study to truly reflect the field of computer science, we needed to ensure that we evaluated a representative sample of research articles. We began with the ACM and IEEE journals that Geist et al. used in their 1996 study of faculty productivity and eliminated two software engineering journals (*ACM Transactions on Software Engineering and Methodology and IEEE Transactions on Software Engineering*) as well as *IEEE Transactions on VLSI*, which does not appear in the list of IEEE Computer Society publications. [4] Table 1 presents the 13 journals examined.

We used a sampling approach that enabled us to select approximately 500 articles for evaluation. We wanted to ensure that, as a group, the two primary publication outlets, IEEE and ACM Transactions were reflected equally in the sample set. [5] Based on the number of articles published during the years 1995–1999, inclusive, we selected 1 in 10 articles from the IEEE journals and 1 in 3 articles from ACM journals. This approach resulted in approximately 309 articles in IEEE journals and 286 articles in ACM journals, as well as 33 articles from a joint IEEE/ACM publication (*Transactions on Networking*). Table 2 presents raw data for the number of articles examined in each of the journals during the five-year period.

Table 1
The journals examined

Journal title	Abbreviation
IEEE Transactions on Computers	COMP
Journal of the ACM	JACM
IEEE Transactions on Knowledge and Data Engineering	KDE
IEEE Transactions on Pattern Analysis and Machine Intelligence	PAMI
IEEE Transactions on Parallel and Distributed Systems	PDS
ACM Transactions on Human–Computer Interaction	TOCHI
ACM Transactions on Database Systems	TODS
ACM Transactions on Graphics	TOG
ACM Transactions on Information Systems	TOIS
ACM Transactions on Modeling and Computer Simulation	TOMCS
IEEE/ACM Transactions on Networking	TON
ACM Transactions on Programming Languages and Systems	TOPLAS
IEEE Transactions on Visualization and Computer Graphics	VCG

2.3. The classification process

Two of the three authors of this paper independently classified each article using just one category in each of the five characteristics. Hence the coding reflected the primary focus of the research. Following the individual codings, the first author resolved differences by reexamining the article and choosing a final coding that was typically one of the two original codings.

Agreement varied among categories. For example, high levels of agreement were achieved for research method and reference discipline coding (close to 90%), while coding of level of analysis and topic was somewhat more problematic (70% and 75% agreement, respectively). Disagreement occurred most often when a paper could legitimately have been coded in more than one way. Original agreements across all categories averaged around 80%.

3. Findings

In the following section the study findings are presented by research question; that is, we address the topics, research approaches, research methods, levels of analysis, and reference disciplines that CS researchers use, in turn. Tables 3–7 summarize the findings. Although this study was designed to characterize research in the CS *discipline*, it is also interesting to examine differences in the journals themselves. Hence, in each of the sections, we also highlight the findings by journal.

[4] *ACM Transactions on Software Engineering and Methodology* (TOSEM) and *IEEE Transactions on Software Engineering* (TSE) were not included in this study because they are primarily software engineering journals and were therefore examined in our analysis of the software engineering literature. This analysis is reported in Glass et al. (2002).

[5] An alternative distribution scheme based on the amount of research published could also have been used here. However, we did not want our sample to be overwhelmed by publications in IEEE Transactions, which publish more issues per year and more articles per issue than ACM Transactions.

Table 2
Numbers of publications examined by journal and year

	Overall	COMP	JACM	KDE	PAMI	PDS	TOCHI	TODS	TOG	TOIS	TOMCS	TON	TOPLAS	VCG
1995	141	21	20	10	17	16	4	10	7	9	6	4	13	4
1996	128	20	16	10	17	14	7	3	7	7	6	8	9	4
1997	135	18	16	10	19	14	6	7	7	6	9	9	11	3
1998	122	17	15	8	19	12	5	5	6	7	8	6	11	3
1999	102	15	11	8	17	10	3	3	4	6	8	6	8	3
Totals	628	91	78	46	89	66	25	28	31	35	37	33	52	17

Table 8 presents the data by journal for each of the categories examined. While some of the results for the discipline as a whole and the journals are somewhat predictable, some are fairly surprising.

3.1. Findings for topic

Table 3 shows that research in computer science is spread evenly among the five categories: computer concepts (28.67%), problem-domain-specific concepts (21.50%), systems/software concepts (19.11%), data/information concepts (15.45%), and problem-solving concepts (14.65%). Two other categories, systems/software management concepts, and organizational concepts, are represented minimally, while two categories, societal concepts and disciplinary issues are not represented at all.

The leading sub-category was computer graphics/pattern analysis within the problem-domain-specific concepts category. Twenty percent of articles were devoted to this category, while 17.68% were devoted to inter-computer communication (part of computer concepts), which includes such topics as networking and distributed systems. Other notable topics were computer/hardware principles/architecture at 10.19% (again part of computer concepts) and database/warehouse/mart organization at 8.44% (part of data/information concepts), while papers focusing on mathematics/computational science (part of problem-solving concepts) were next at 6.69%.

Table 8 (Panel A) presents the topics by journal. The results show that most journals tended to have a single dominant topic as suggested by their title. These topics, then, broadly define the sub-fields that make up the discipline of computer science. We found that 2 or 3 of the 13 journals typically focused on the same topic area. For example, the principal topic category in *Journal of the ACM* and *ACM Transactions on Modeling and Computer Simulation* was problem-solving concepts; in *IEEE Transactions on Computers*, *IEEE Transactions on Parallel and Distributed Systems*, and *IEEE/ACM Transactions on Networking* it was computer concepts; in *ACM Transactions on Computer–Human Interaction* and *ACM Transactions on Programming Languages and Systems*, it was systems/software concepts, in *IEEE Transactions on Knowledge and Data Engineering*, *ACM Transactions on Database Systems*, and *ACM Transactions on Information Systems*, it was data/information concepts, and in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *ACM Transactions on Graphics*, and *IEEE Transactions on Visualization and Computer Graphics*, it was problem-domain-specific concepts.

3.2. Findings for research approach

Table 4 shows the primary research approaches used by CS researchers. Formulative was by far the dominant research approach representing 79.15% of the papers assessed, followed by evaluative and descriptive approaches, which were virtually equivalent at 10.98% and 9.88%, respectively.

Examination of the sub-categories of research approach shows that FP, a multifaceted subcategory that includes formulating processes, procedures, methods, or algorithms is the most important of the formulative subcategories. Approximately half of computer science research (50.55%) fell into this category. The next largest category was FC (e.g., formulating a concept such as a data model), at 17.04%. Papers whose primary focus was evaluation using techniques other than deductive, interpretive, or critical approaches (evaluative-other) were third at 9.87%.

Table 8 (Panel B) shows the primary research approaches by journal. The data shows that FP (formulate-process, method, or algorithm) was the most important research approach in 12 of the 13 journals examined while formulate-concept (FC) was the second most important approach (in 8 out of those 12 journals). *ACM Transactions on Computer–Human Interaction* was the only journal in which the formulative research category did not dominate. Instead, 40% of the papers in that journal were devoted to evaluative studies (evaluative-deductive and evaluative-other at 20% each), with a further 20% devoted to system descriptions (DS). Other journals with significant numbers of evaluative studies were *ACM Transactions on Programming Languages and Systems* (21.15%) and *Journal of the ACM* (20.51%).

Our results suggest, therefore, that the focus in most areas of computer science research is primarily on formulating things.

Table 3
Findings for computing topics

1.0	*Problem-solving concepts*	14.65%
1.1	Algorithms	5.57%
1.2	Mathematics/computational science	6.69%
1.3	Methodologies (object, function/process, information/data, event, business rules,...)	–
1.4	Artificial intelligence	2.39%
2.0	*Computer concepts*	28.67%
2.1	Computer/hardware principles/architecture	10.19%
2.2	Inter-computer communication (networks, distributed systems)	17.68%
2.3	Operating systems (as an augmentation of hardware)	0.80%
2.4	Machine/assembler-level data/instructions	–
3.0	*Systems/software concepts*	19.11%
3.1	System architecture/engineering	0.48%
3.2	Software life-cycle/engineering (including requirements, design, coding, testing, maintenance)	–
3.3	Programming languages	3.82%
3.4	Methods/techniques (including reuse, patterns, parallel processing, process models, data models...)	3.82%
3.5	Tools (including compilers, debuggers)	5.25%
3.6	Product quality (including performance, fault tolerance)	1.75%
3.7	Human–computer interaction	3.18%
3.8	System security	0.80%
4.0	*Data/information concepts*	15.45%
4.1	Data/file structures	1.91%
4.2	Data base/warehouse/mart organization	8.44%
4.3	Information retrieval	3.98%
4.4	Data analysis	0.64%
4.5	Data security	0.48%
5.0	*Problem-domain-specific concepts (use as a secondary subject, if applicable, or as a primary subject if there is no other choice)*	21.50%
5.1	Scientific/engineering (including bio-informatics)	0.48%
5.2	Information systems (including decision support, group support systems, expert systems)	0.64%
5.3	Systems programming	–
5.4	Real-time (including robotics)	0.16%
5.5	Computer graphics/pattern analysis	20.22%
6.0	*Systems/software management concepts*	0.32%
6.1	Project/product management (including risk management)	0.32%
6.2	Process management	–
6.3	Measurement/metrics (development and use)	–
6.4	Personnel issues	–
7.0	*Organizational concepts*	0.32%
7.1	Organizational structure	–
7.2	Strategy	–
7.3	Alignment (including business process reengineering)	–
7.4	Organizational learning /knowledge management	–
7.5	Technology transfer (including innovation, acceptance, adoption, diffusion)	0.16%
7.6	Change management	–
7.7	Information technology implementation	–
7.8	Information technology usage/operation	–
7.9	Management of "computing" function	0.16%
7.11	IT impact	–
7.11	Computing/information as a business	–
7.12	Legal/ethical/cultural/political (organizational) implications	–
8.0	*Societal concepts*	–
8.1	Cultural implications	–
8.2	Legal implications	–
8.3	Ethical implications	–
8.4	Political implications	–
9.0	*Disciplinary issues*	–
9.1	"Computing" research	–
9.2	"Computing" curriculum/teaching	–

Table 4
Findings for research approach

Descriptive:		9.88%
DS	Descriptive-system	4.14%
DO	Descriptive-other	5.10%
DR	Review of literature	0.64%
Evaluative:		10.98%
ED	Evaluative-deductive	1.11%
EI	Evaluative-interpretive	–
EC	Evaluative-critical	–
EO	Evaluative-other	9.87%
Formulative:		79.15%
FF	Formulative-framework	2.39%
FG	Formulative-guidelines/standards	0.64%
FM	Formulative-model	5.73%
FP	Formulative-process, method, algorithm	52.55%
FT	Formulative-classification/taxonomy	0.80%
FC	Formulative-concept	17.04%

Table 5
Findings for research method

AR	Action research	–
CA	Conceptual analysis	15.13%
CAM	Conceptual analysis/mathematical	73.41%
CI	Concept implementation (proof of concept)	2.87%
CS	Case study	0.16%
DA	Data analysis	0.16%
DI	Discourse analysis	–
ET	Ethnography	–
FE	Field experiment	–
FS	Field study	0.16%
GT	Grounded theory	–
HE	Hermeneutics	–
ID	Instrument development	–
LH	Laboratory experiment (human subjects)	1.75%
LR	Literature review/analysis	0.32%
LS	Laboratory experiment (software)	1.91%
MA	Meta-analysis	–
MP	Mathematical proof	2.39%
PA	Protocol analysis	–
PH	Phenomenology	–
SI	Simulation	1.75%
SU	Descriptive/exploratory survey	–

Table 6
Findings for level of analysis

SOC	Society	–
PRO	Profession	0.32%
EXT	External business context	–
OC	Organizational context	–
PR	Project	–
GP	Group/team	–
IN	Individual	1.91%
CS	Computing system	5.57%
CE	Computing element—program, component, algorithm	53.34%
AC	Abstract concept	38.85%

Table 7
Findings for reference discipline

CP	Cognitive psychology	0.80%
SB	Social and behavioral science	–
CS	Computer science	89.33%
SC	Science	0.96%
EN	Engineering	–
EC	Economics	–
LS	Library science	–
MG	Management	–
MS	Management science	–
PA	Public administration	–
PS	Political science	–
MA	Mathematics	8.60%
OT	Other	0.32%

3.3. Findings for research method

Table 5 presents the primary research methods used by CS researchers. Conceptual Analysis/Mathematical (CA/M) (73.41%) was the primary research method with conceptual analysis (not using mathematical techniques) next at 15.13%. Categories such as laboratory experiment (using human subjects), laboratory experiment (software), simulation, and concept implementation are also represented, although none reached double-digits.

Table 8 (Panel C) shows the findings for research method by journal. CA/M was the most important research method in all journals except *ACM Transactions on Computer–Human Interaction (TOCHI)*. The figures ranged from a low of 37.14% in *ACM Transactions on Information Systems (TOIS)* to a high of 90.32% in *ACM Transactions on Graphics* (TOG). In *TOCHI*, which published no studies using CA/M, the leading research methods were conceptual analysis (40%) and laboratory experiment (36%). Concept implementation as a research method was primarily used in TOCHI (16%) and TOIS (11.43%). *TOIS* was also the only journal in which comparative studies of systems (laboratory experiment (software)) was used as the primary research method (14.29%).

3.4. Findings for level of analysis

Table 6 presents the levels of analysis used by CS researchers. It shows that, similar to research approach and research method, CS research was also relatively focused in terms of levels of analysis. The most dominant level of analysis was the Computing Element (CE) category (53.34%), which relates to algorithms, methods, and techniques, e.g., a scheduling algorithm for a crossbar switch. The Abstract Concept (AC) category, which relates to concepts such as the definition of global predicates in the context of distributed computations, was the next largest at 38.85%. Finally, 5.57% of the papers focused on the computing system (CS) level. Two other categories (individual (IN) and profession (PRO))

Table 8
Representation by topic (Panel A), representation by reasearch approach (Panel B), representation by research method (Panel C); representation by level of analysis (Panel D), representation by reference discipline (Panel E)

	Overall	COMP	JACM	KDE	PAMI	PDS	TOCHI	TODS	TOG	TOIS	TOMCS	TON	TOPLAS	VCG
Panel A:														
Prob-solving	14.65%	12.09%	44.87%	28.26%	5.62%	6.06%	–	–	3.23%	2.86%	51.35%	–	1.92%	11.76%
Computer	28.66%	69.23%	26.92%	2.17%	–	68.18%	–	–	–	–	37.84%	96.97%	7.69%	–
Systs/SW	19.11%	16.48%	11.54%	6.52%	–	22.73%	80.00%	3.57%	6.45%	17.14%	5.41%	3.03%	88.46%	–
Data/Info	15.45%	1.10%	16.67%	60.87%	4.49%	1.51%	8.00%	96.43%	–	54.29%	2.70%	–	1.92%	–
Prob-domain	21.50%	1.10%	–	2.17%	89.89%	1.52%	8.00%	–	90.32%	17.14%	2.70%	–	–	88.24
Sys/SW Mgt	0.32%	–	–	–	–	–	–	–	–	5.71%	–	–	–	–
Org'al	0.32%	–	–	–	–	–	4.00%	–	–	2.86%	–	–	–	–
Societal	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Disc issues	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Panel B:														
DO	5.10%	5.49%	–	2.17%	5.62%	–	–	7.14%	3.23%	8.57%	24.32%	3.03%	7.69%	5.88%
DR	0.64%	–	–	4.35%	1.12%	–	4.00%	–	–	–	–	–	–	–
DS	4.14%	3.30%	–	6.52%	2.25%	1.52%	20.00%	3.57%	–	11.43%	10.81%	–	1.92%	11.76%
ED	1.11%	–	–	–	–	–	20.00%	–	6.45%	–	–	–	–	–
EI	–	–	–	–	–	–	–	–	–	–	–	–	–	–
EO	9.87%	7.69%	20.51%	6.52%	6.74%	3.03%	20.00%	7.14%	–	8.57%	10.81%	9.09%	21.15%	–
EC	–	–	–	–	–	–	–	–	–	–	–	–	–	–
FC	17.04%	23.08%	29.49%	17.39%	13.48%	7.58%	8.00%	17.86%	16.13%	8.57%	18.92%	3.03%	28.85%	–
FF	2.39%	–	1.28%	2.17%	8.99%	–	–	3.57%	–	8.57%	–	–	1.92%	–
FG	0.64%	3.30%	–	–	–	–	4.00%	–	–	–	–	–	–	–
FM	5.73%	6.59%	1.28%	13.04%	5.62%	7.58%	4.00%	–	3.23%	17.14%	2.70%	12.12%	–	–
FP	52.55%	50.55%	46.15%	47.83%	56.18%	80.30%	16.00%	50.00%	70.97%	37.14%	32.43%	72.73%	38.46%	82.35%
FT	0.80%	–	1.28%	–	–	–	4.00%	10.71%	–	–	–	–	–	–
Panel C:														
AR	–	–	–	–	–	–	–	–	–	–	–	–	–	–
CA	15.13%	19.78%	–	36.96%	4.49%	12.12%	40.00%	10.71%	3.23%	34.29%	16.22%	24.24%	7.69%	23.53%
CAM	73.41%	69.23%	89.74%	52.17%	89.89%	81.82%	–	78.57%	90.32%	37.14%	72.97%	69.70%	86.54%	70.59%
CI	2.87%	2.20%	–	4.35%	–	–	16.00%	3.57%	–	11.43%	5.41%	–	3.85%	5.88%
CS	0.16%	–	–	–	–	–	–	–	–	–	2.70%	–	–	–
DA	0.16%	–	–	–	–	–	–	–	–	2.86%	–	–	–	–
ET	–	–	–	–	–	–	–	–	–	–	–	–	–	–
FE	–	–	–	–	–	–	–	–	–	–	–	–	–	–
FS	0.16%	–	–	–	–	–	4.00%	–	–	–	–	–	–	–
GT	–	–	–	–	–	–	–	–	–	–	–	–	–	–
HE	–	–	–	–	–	–	–	–	–	–	–	–	–	–
ID	–	–	–	–	–	–	–	–	–	–	–	–	–	–
LH	1.75%	–	–	–	–	–	36.00%	–	6.45%	–	–	–	–	–
LR	0.32%	1.10%	–	–	1.12%	–	–	–	–	–	–	–	–	–
LS	1.91%	1.10%	–	–	1.12%	1.52%	4.00%	7.14%	–	14.29%	–	–	1.92%	–
MP	2.39%	4.40%	10.26%	–	3.37%	–	–	–	–	–	–	–	–	–
PA	–	–	–	–	–	–	–	–	–	–	–	–	–	–
SI	1.75%	2.20%	–	6.52%	–	4.55%	–	–	–	–	2.70%	6.06%	–	–
SU	–	–	–	–	–	–	–	–	–	–	–	–	–	–

Panel D:														
SOC	–	–	–	–	–	–	–	–	–	–	–	–	–	–
PRO	0.32%	–	–	2.17%	1.12%	–	–	–	–	–	–	–	–	–
EXT	–	–	–	–	–	–	–	–	–	–	–	–	–	–
OC	–	–	–	–	–	–	–	–	–	–	–	–	–	–
PR	–	–	–	–	–	–	–	–	–	–	–	–	–	–
GP	–	–	–	–	–	–	–	–	–	–	–	–	–	–
IN	1.91%	–	–	–	–	–	40.00%	–	6.45%	–	–	–	–	–
AC	38.85%	56.04%	42.31%	34.78%	44.94%	12.12%	24.00%	35.71%	22.58%	42.86%	45.95%	36.36%	55.77%	–
CS	5.57%	4.40%	–	10.87%	2.25%	–	20.00%	3.57%	–	17.14%	16.22%	–	7.69%	11.76%
CE	53.34%	39.56%	57.69%	52.17%	51.69%	87.88%	16.00%	60.71%	70.97%	40.00%	37.84%	63.84%	36.54%	88.24%
Panel E:														
CP	0.80%	–	–	–	–	–	20.00%	–	–	–	–	–	–	–
MA	8.60%	1.10%	41.03%	–	7.87%	–	–	–	6.45%	2.86%	18.92%	3.03%	5.77%	–
SB	–	–	–	–	–	–	–	–	–	–	–	–	–	–
CS	89.33%	98.90%	57.69%	97.83%	92.13%	100.00%	72.00%	100.00%	90.32%	97.14%	81.08%	96.97%	94.23%	82.35%
EC	–	–	–	–	–	–	–	–	–	–	–	–	–	–
IS	–	–	–	–	–	–	–	–	–	–	–	–	–	–
MG	–	–	–	–	–	–	–	–	–	–	–	–	–	–
MS	–	–	–	–	–	–	–	–	–	–	–	–	–	–
OT	0.32%	–	–	–	–	–	8.00%	–	–	–	–	–	–	–
SC	0.96%	–	1.28%	2.17%	–	–	–	–	3.23%	–	–	–	–	17.65%

Table 1 presents the journal titles; The shaded areas represent the highest percentage and 2nd highest percentage, respectively.

were below 2%, while the five categories of societal, organizational context, external business context, project, and group/team were not represented.

Table 8 (Panel D) presents level of analysis by journal. The data shows that CE was the primary level of analysis in 8 of the 13 journals. The figures ranged from a low of 51.69% in *IEEE Transactions on Pattern Analysis and Machine Intelligence* to a high of 88.24% in *IEEE Transactions on Visualization and Computer Graphics (VCG)*. Further, AC was the primary level of analysis in four journals ranging from 42.86% to 56.04%, while Individual was the primary level of analysis in *ACM Transactions on Computer–Human Interaction (TOCHI)*. In addition, *TOCHI* and *ACM Transactions on Graphics (TOG)* were the only journals to publish articles that used a non-technical level of analysis (i.e., levels of analysis other than AC, CS [6] or CE) with 40% of the articles in *TOCHI* and 6.45% of the articles in *TOG* focusing on the individual level.

3.5. Findings for reference discipline

Table 7 presents the reference disciplines used by CS researchers. The results suggest that, for the most part, CS research does not rely on other fields for its fundamental theories and/or concepts. Of the papers examined, Computer Science itself was the reference discipline in 89.33% of the cases. The only other discipline that emerged was mathematics (8.60%). There were trivial instances of papers that relied on cognitive psychology (0.80%) and science (0.96%).

Table 8 (Panel E) presents the breakdown of reference discipline by journal. Not surprisingly, computer science was the primary reference discipline in all journals, ranging from a low of 57.69% in *Journal of the ACM (JACM)* to a high of 100% in *IEEE Transactions on Parallel and Distributed Systems*. Mathematics was a major reference discipline in *JACM* with 41% of the articles using concepts directly from that discipline. Only two journals did not have mathematics as their second most important reference discipline (*TOCHI* and *VCG*). Cognitive psychology emerged as a major reference discipline in *TOCHI* (20%) and Science in *VCG* (17.65%).

4. Discussion and implications

In this study, we sought to analyze the characteristics of computer science research according to five research characteristics all of which are recorded in the literature as being important aspects of any research study. We first provide a brief summary of the key findings, followed by a discussion of the some of the limitations of our study.

[6] Computer System.

CS research is fairly evenly distributed across five major topic areas: problem-solving concepts, computer concepts, systems/software concepts, data/information concepts and problem-domain-specific concepts. The leading category is computer concepts, with problem-domain-specific concepts (principally computer graphics and pattern analysis) second. As would be expected, there is very little work in the area of systems/software management concepts (two papers), one paper on organizational concepts, and no papers that examined societal concepts or disciplinary issues.

In terms of both research approach and research method, CS research tends to be quite focused. The "formulate" research approach category accounts for almost 80% of the research with a majority of papers being devoted to formulating a process, method, or algorithm. The preferred research method is conceptual analysis based on mathematical techniques.

With regard to levels of analysis, CS research falls primarily into the CE or AC categories confirming that the mission of CS is to conduct research that is focused on technical levels of analysis. As would be expected, very little research focused on the society or profession categories.

With respect to reference disciplines, our data shows that CS research seldom relies on research in other disciplines and in the rare instances that it does, it relies primarily on mathematics.

Table 9 presents a summary of the most important research characteristics in each of the 13 journals. The data indicate that while CS research addresses a diverse range of topics, there is a high degree of consistency in terms of the research approaches, research methods, and levels of analysis used to study these topics. Further, across the 13 journals studied, *ACM Transactions on Computer–Human Interaction* is a clear outlier. It is, for example, the only journal not to have FP (formulate process, method or algorithm) as the predominant research approach and CA/M as the predominant research method. From the viewpoint of level of analysis, CE dominates AC by eight journals to four. It is, however, interesting to note that each of the four journals in which AC is dominant focuses on one of the major topic categories; the only topic category that is not the focus of one or more of the journals we studied is problem-domain-specific concepts.

Note that we used our classification system to record the keywords describing this paper (following the abstract). The paper is classified as follows: (a) the topic is computing research (9.1); (b) the research approach is EO (evaluative-other) because our paper is about evaluating CS research; (c) the research method is LR (literature review/analysis); (d) the level of analysis is the profession (PRO); and (e) the reference discipline is none because we did not use concepts from other disciplines in performing the study. We encourage authors in the future to use our classification system not only to select keywords but also to write abstracts. Such a practice would aid researchers to assess the relevance of published research to their own endeavors.

A study of this nature is not without limitations. The first limitation stems from the choice of journals. The results of our study reflect the nature of computer science research to the extent that these journals are representative of the field. While there are many other magazines, and high-quality research conferences that publish CS research articles, we chose to analyze only articles published in journals because of the traditional and enduring role that journals play in the development of academic disciplines. A second potential limitation arises from the fact that we coded only a sample of the articles published in the selected journals. Given, however, that we used a systematic sampling procedure, we have no reason to believe that the results are biased. A final limitation arises from the subjective nature of the coding process. We attempted to reduce the subjectivity by using two independent coders who revisited the articles to resolve any disagreements. The relatively high-level of raw agreements suggests that articles were indeed coded in a consistent manner.

Table 9
Summary of characteristics of journals

Journal	Principal topic	Research approach	Research method	Level of analysis	Reference discipline
TOMCS	Problem-solving	FP	CA/M	AC	CS
JACM	Problem-solving	FP	CA/M	CE	CS
COMP	Computer	FP	CA/M	AC	CS
PDS	Computer	FP	CA/M	CE	CS
TON	Computer	FP	CA/M	CE	CS
TOIS	Data/information	FP	CA/M	AC	CS
TODS	Data/information	FP	CA/M	CE	CS
KDE	Data/information	FP	CA/M	CE	CS
PAMI	Problem-domain-specific	FP	CA/M	CE	CS
TOG	Problem-domain-specific	FP	CA/M	CE	CS
VCG	Problem-domain-specific	FP	CA/M	CE	CS
TOPLAS	Systems/software	FP	CA/M	AC	CS
TOCHI	Systems/software	DS, ED, EO	CA	IN	CS

5. Conclusion

We examined 628 papers (over a 5-year period) in 13 leading research journals in the CS field from 1995 to 1999 to answer questions regarding the nature of CS research. We characterized CS research in terms of the topics, research approaches, research methods, levels of analysis, and reference disciplines used. Our results suggest that CS research focuses on a variety of technical topics, using formulative approaches to study new entities that are either computing elements or abstract concepts, principally using mathematically-based research methods.

The results from our study should be of value to both researchers and doctoral students engaged in computer science research. For example, our study provides a characterization of the types of articles that computer science journals publish. Researchers can use this knowledge to make choices when deciding on a target journal for their research. Our results can also be used to provide insights into areas of CS research that are receiving little research attention. For example, in terms of research approaches, our results clearly suggest that insufficient emphasis is being placed on the use of evaluative methodologies. However, while our results clearly support Tichy et al.'s (1995) claim regarding the lack of focus on evaluation in CS research, Fletcher (1995) cautions that the use of experimental methods may not always be appropriate in computer science, a caveat that should be kept in mind.

Further, funding organizations such as NSF could use the findings of our research to provide focused calls for proposals aimed at fostering research in particular areas or using particular approaches/methods. It is important to note, however, that any interpretation of gaps represented in or findings must take into account the fact the classification scheme was developed to cover a broader scope than computer science, alone, by also including the disciplines of software engineering, and information systems. Hence, for example, while our results clearly show that there is a lack of emphasis on organizational aspects of computing, that is the focus of IS researchers (see Vessey et al., 2002) and does not necessarily represent opportunities for CS researchers.

We hope that our evaluation of the state of current CS research fosters future CS research efforts.

References

ACM CCS: ACM Computing Classification System [1998 version], http://www.acm.org/class/1998, 1998, accessed 11/06/01.

Alavi, M., Carlson, P., 1992. A review of MIS research and disciplinary development: implications for deans/administrators. Journal of Management Information Systems 8, 45–62.

Barki, H., Rivard, S., Talbot, J., 1988. An information systems keyword classification scheme. MIS Quarterly 12 (2), 299–322.

Cheon, M.J., Grover, V., Sabherwal, R., 1993. The evolution of empirical research in IS: a study in IS maturity. Information and Management 24 (5), 107–119.

Estivili-Castro, V., 1995. Computer Science research in Mexico. IEEE Computer 28 (8), 56–62.

Farhoomand, A.F., Drury, D.H., 2000. A Historiographical Examination of Information Systems. Communication of the AIS Vol. 1, Article 19, 2000.

Fletcher, P., 1995. The role of experiments in Computer Science. Journal of Systems and Software 30 (1–2), 61–163.

Gallopoulous, E., Sameh, A., 1997. CSE: content and product. IEEE Computational Science and Engineering 4 (2), 39–43.

Geist, R., Chetuparambil, M., Hedetniemi, S., Turner, A.J., 1996. Computing research programs in the US. Communications of the ACM 39 (12), 96–99.

Glass, R.L., 1992. A comparative analysis of the topic areas of Computer Science, Software Engineering, and Information Systems. Journal of Systems and Software (November), 277–289.

Glass, R.L., 1995. A structure-based critique of contemporary computing research. Journal of Systems and Software (January).

Glass, R.L., Vessey, I., Ramesh, V., 2002. Research in software engineering: an analysis of the literature. Information and Software Technology 44, 491–506.

Gruman, G., 1990. Conference tries to define computing-research agenda. IEEE Software 7 (1), 127–132.

Harrison, R., Wells, M., 2000. A Meta-Analysis of Multidisciplinary Research. In: Papers from The Conference on Empirical Assessment in Software Engineering (EASE), pp. 1–15.

Hull, R., King, R., 1987. Semantic database modeling: survey, applications and research issues. ACM Computing Surveys 19 (3), 201–260.

Lai, V.S., Mahapatra, R.K., 1997. Exploring the research in information technology implementation. Information and Management 32, 187–201.

Morrison, J., George, J.F., 1995. Exploring the Software Engineering component in MIS research. Communications of the ACM 38 (7), 80–91.

Orlikowski, W., Baroudi, J.J., 1991. Studying information technology in organizations: research approaches and assumptions. Information Systems Research 2 (1), 1–28.

Ramamritham, K., 1997. Computer Science research in India. IEEE Computer, 40–47.

Rice, J.R., 1995. Computational science and the future of computing research. IEEE Computational Science and Engineering 2 (4), 34–41.

Sheth, A., Larson, J., 1990. Federated database systems for managing distributed, heterogeneous, and autonomous databases. ACM Computing Surveys 22 (3), 183–236.

Swanson, E.B., Ramiller, N., 1993. Information systems research thematics: submissions to a new journal 1987–1992. Information Systems Research 4 (4), 299–330.

Tichy, W.F., Lukowicz, P., Prechelt, L., Heinz, E., 1995. Experimental evaluation in computer science: a quantitative study. Journal of Systems and Software (January).

Vessey, I, Ramesh, V., Glass, R.L., 2001. Research in Computing Disciplines: A Comprehensive Classification Scheme, Indiana University: Working Paper (TR 107-1), http://www.kelley.iu.edu/ardennis/wp/ (accessed 02/08/02), 2001.

Wegner, P., Doyle, J., 1996. Strategic directions in computing research. ACM Computing Surveys 28 (4), 565–574.

Westin, S.M., Roy, M., Kim, C.K., 1994. Cross-fertilizations of knowledge: the case of MIS and its reference disciplines. Information Resources Management Journal 7 (2), 24–34.

Zelkowitz, M.V., Wallace, D., 1997. Experimental validation in software engineering. Information and Software Technology 39, 735–743.

Ramesh Venkataraman, Ph.D., is an Assistant Professor of Information Systems and Ford Motor Company Teaching Fellow in the Department of Accounting and Information Systems at Kelley School of Business, Indiana University. He has published over 25 papers in leading journals, books, and conferences. His areas of expertise are in database modeling and design, systems design and development, heterogeneous databases, and groupware systems.

Robert L. Glass is president of Computing Trends, publishers of *The Software Practitioner*. He has been active in the field of computing and software for over 45 years, largely in industry (1954–1982 and 1988–present), but also as an academic (1982–1988). He is the author of over 20 books and 70 papers on computing subjects, Editor Emeritus of Elsevier's *Journal of Systems and Software*, and a columnist for several periodicals including *Communications of the ACM* (the "Practical Programmer" column) and *IEEE Software* ("The Loyal Opposition"). He was for 15 years a Lecturer for the ACM, and was named a Fellow of the ACM in 1998. He received an honorary Ph.D. from Linkoping University in Sweden in 1995. He describes himself by saying "my head is in the academic area of computing, but my heart is in its practice."

Iris Vessey is Professor of Information Systems at Indiana University's Kelley School of Business, Bloomington. She received her M.Sc., MBA, and Ph.D. in MIS from the University of Queensland, Australia. Her research into evaluating emerging information technologies from both cognitive and analytical perspectives has been published in journals such as *Communications of the ACM*, *Information Systems Research*, *Journal of Management Information Systems*, and *MIS Quarterly*. In recent years, her interests have focused on managerial issues associated with the management and implementation of enterprise systems. She currently serves as Secretary of the Association for Information Systems and of the International Conference on Information Systems and is an AIS Fellow.

Behavior Research Methods & Instrumentation
1983, Vol. 15(2), 222-227

Research methods for computer applications

JO W. TOMBAUGH
Carleton University, Ottawa K1S 5B6, Canada

This paper identifies issues and techniques useful to psychologists interested in applied research for computer applications. Applied research includes guideline research, product development evaluation, and field trial evaluation. Suggestions for improved guideline research involve screening the problem to determine its potential usefulness and screening the design to make sure it involves sufficiently representative tasks, subjects, and materials for the desired generality. Product development evaluation requires techniques for testing users that are fast, flexible, and inexpensive, such as "user trials" of products in ergonomics and "user edits" for testing documentation. Field trials of office automation or videotex systems lead to decisions about further introduction of those systems. A methodology for planning the research component of field trials, "evaluation assessment," provides a set of steps to use in deciding which design and measures should be collected.

Psychologists can serve as professional communicators who summarize and interpret relevant psychological literature for system designers (Macdonald-Ross, 1978). They can also contribute both conceptual and empirical methods to the area (Barnard, Hammond, Morton, Long, & Clark, 1981). Conceptual tools are analytic methods developed to help system designers make appropriate design decisions (e.g., Moran, 1981b). Empirical methods, the focus of this paper, are those that involve data collection in the process of dealing with computer applications.

Three types of applied research are covered in this article: guideline, product development, and evaluation research. Guideline research tends to be centered in universities or in human-factors research laboratories financed by government and large manufacturers. It is similar to traditional psychological research, even though the strategies that lead to the development of useful guidelines are not well understood (Barnard et al., 1981; Moher & Schneider, 1982).

With product development and evaluation research, psychologists must find workable solutions to specific problems rather than discover general principles. Questions are concerned with whether (as opposed to why) a particular procedure works. The controlled experiment is often inadequate and inappropriate for this type of research. Instead, methods are needed for gathering empirical evidence on such questions quickly and efficiently. Product testing methods in ergonomics (Rennie, 1981) and education (Sanders & Cunningham, 1973), as well as requirements analysis methods for computer systems (Ramsey & Atwood, 1979), are particularly useful. Sending a psychologist into an applied research setting with knowledge of only traditional experimental designs is comparable to sending a carpenter to build a house with only a hammer and saw.

This research was supported by a grant from the National Science and Engineering Research Council of Canada (Strategic Grant G0365).

The third type of applied research is evaluation research. Major computer systems are typically tested in field trials before being widely implemented. Information gathered in these tests ranges from questions of equipment reliability, to user satisfaction, to measuring changes in user activity and productivity. An awareness of the difficulties in field research (Cook & Campbell, 1979; Rutman, 1980) is important to these kinds of endeavors. Methodological considerations associated with each of these three approaches are considered in turn.

APPROACHES TO DEVELOPING GUIDELINES

Ideally, guidelines research will lead to general principles that aid system designers. As Moran (1981a, p. 5) has stated, "An applied psychology of the user, like any applied science, is first of all a science. This means that it not only collects data, but also attempts to formulate theories to explain the data . . . . Its role is to provide the designer with a systematic way of thinking about user behavior and methods for dealing with it."

A number of writers in this area (Barnard et al., 1981; Moher & Schneider, 1982; Moran, 1981a; Ramsey & Atwood, 1979) have stressed the need to develop research techniques that lead to the development of useful theories and guidelines. "Although the literature contains numerous references to the *use* of experimental methods, there are few references on investigations into the methodology itself" (Moher & Schneider, 1982, p. 67). Recent reviews in a number of applied areas—typography (Macdonald-Ross, 1978), technical writing (Anderson, 1980; Wright, 1978), learning from text (McConkie, 1978)—also consider the methodologies employed to answer practical questions in these applied fields to be inadequate. Laboratory research concerning a single variable, using simplified stimulus material and simplified performance criteria, are often encountered. The question addressed is often "competitive" (e.g., Are menu-driven systems better than command-driven systems?) instead of "conditional" (e.g., Under what conditions is a menu-driven system better than a command-driven system?). Overly simplified research of this type lacks credibility because of the lack of generality of the results and the failure to provide information about interactions.

To encourage conditional questions and to avoid the difficulties of single-variable research, many writers suggest using a systems or framework approach in which the characteristics of user, task, performance measures, and other situational factors are considered jointly (e.g., Benbasat, Dexter, & Masulis, 1981). The difficulty with this approach is that it always identifies more potential interacting variables than can be manipulated in a single study, or even a single research program. As noted by Barnard et al. (1981, p. 89), "any ergonomic approach which attempts to formulate guidelines or design principles on the basis of highly selective parametric studies is likely to have inherent limitations." Thus, there are two challenges for the applied researcher: (1) deciding what variables to manipulate and measure (selecting the problem), and (2) insuring that the results are sufficiently general to be useful to practitioners (designing the experiment).

Selecting the Problem

An applied researcher is motivated in part by a desire for his results to be directly useful (Marx & Hillix, 1963). Thus, the problem to be studied must be selected with care. Advice found in the literature focuses on the origin of the problem. Problem sources include theory, user errors, user/computer incompatibility, expert-practitioner, and naturalistic observation, as described below.

Theory. Moran (1981a) has suggested that theory should guide the formation of problems. Decisions concerning variables to study and measure are based on the theory. Card, Moran, and Newell (1980, p. 33) express this approach well: "Our motives for studying the manuscript-editing task are both to extend the theory of man as a symbolic information processing system and to apply the theory of human information processing in practical task domains. The consideration of tasks with real application is important as a check on the often-noted tendency for psychological research to become paradigm-bound, as a way of testing the power of theoretical ideas against the complexity of real-world behavior, and as a means of fostering progress in the application domain." The work by Mayer (e.g., Mayer, 1981) on teaching computer programming and by Sebrechts, Deck, and Black (1983) on manuals for word processing are excellent examples of this approach.

User errors. Chapanis (1965) suggests the use of the critical incident technique, with case studies of particularly poor and good implementations to identify or determine the important factors. Similarly, Norman (1982) analyzes performance errors to derive system design rules. For example, "mode errors" suggest a need for improved system feedback to the users.

User/computer incompatibility. Barnard et al. (1981) state that identified cognitive incompatibilities between human users and computers are a valid source of research ideas. Since one such incompatibility is less flexibility in understanding input by the computer than by the human operator, they choose to study dialogue to develop optimal ways of narrowing this gap.

Expert-practitioner. Anderson (1980) and Mcdonald-Ross (1978) suggest that research should focus on the intuitions of skilled and experienced practitioners, such as successful system designers.

Naturalistic observation. McConkie (1978) argues that empirical, descriptive studies of how the potential end users naturally process information should be the starting point for research.

The problem sources listed above would appear to be determined to a large extent by the experiences and backgrounds of the researchers. All have been used successfully by the investigators who recommend them. Yet using one of these methods as a starting point for problem formation is certainly not sufficient to guarantee utility. The appropriateness of the problem, not its origin, will ultimately determine its applicability. Consequently, criteria for screening a selected problem for appropriateness are needed. Tombaugh and Dillon (Note 1) developed three research criteria after a first effort at guideline research failed. This experiment will be briefly described, followed by an explanation of the criteria and the manner in which the study failed to meet them.

The origin of the experiment resulted from observation of a design flaw in an early prototype of a videotex system. User input was not echoed on the display screen. A field experiment was performed to demonstrate to system designers the importance of feedback (echoed input) for new videotex users. Amount of delay and presence/absence of feedback were manipulated: Delay was incorporated because absence of feedback was expected to cause inappropriate responding under longer delays. Analysis of the results supported this prediction, but it was not particularly useful. System designers already knew that keypresses should be echoed; lack of feedback was a minor problem compared to some others for the users. So, the experiment made no contribution to psychological theory.

The screening criteria developed after this experiment were generalizations of the three problems briefly stated above. Problems should be screened for their reality to users, utility to designers, and contribution to psychological theory. First, reality to users should be assessed by asking whether the experimental problem is truly a problem to the user. This is particularly relevant when the researcher has had little or no contact with the user group or the computer application selected for study. An hour or 2 of observation can often focus the problem. The feedback experiment failed this test because the new users worked so slowly with the system that only delays longer than the system naturally provided resulted in increased responding during delay. In fact, users had many difficulties that influenced performance much more than lack of feedback. Watching a few new users on the system before designing the experiment would have led to a reformulation of the problem.

Utility is a criterion that focuses on whether the results will be useful to designers. This is very difficult for the researcher to assess. Observation of a design flaw, such as failure to provide feedback, is not sufficient evidence. The reason for the existence of the flaw must also be considered. Whenever existing guidelines and common practice show that most designers would recognize an observed flaw as bad practice, the experiment will not be very useful. For the feedback experiment given above, this was the greatest miscalculation. The signs that should have been noted before deciding to conduct the experiment were as follows: (1) It is rare to find a computer system in which user responses are not echoed, and (2) no designer would argue that feedback is undesirable. In fact, before the experiment was completed, a new prototype of the videotex system was released that provided user feedback. The early prototype had omitted that feature because of implementation problems and time constraints, not inappropriate design.

Finally, the theory criterion is concerned with whether the study will advance understanding of cognitive processes. If it does, it will stand on its own merit as a contribution to the psychological literature, whether or not it is useful to designers. The videotex experiment failed this test because it had no manipulations designed to increase understanding of feedback mechanisms. It utilized, but did not expand, knowledge of psychological principles. Thus, a retrospective examination of the experiment using the criteria based on users, designers, and theory indicates that the experiment should not have been attempted.

Designing the Experiment

Even when the problem addressed in guideline research is of interest to designers, there is a possibility that the experimental findings will not have wide application. Design and implementation decisions can limit the usefulness of the results. How to make research sufficiently general to be credible for the applied setting is not well understood. Barnard et al. (1981, p. 90) call for "an integration of different types of empirical evidence and different analytic methods." This approach is called "triangulation" by Denzin (1978) and is used to increase the credibility of research findings. The rationale for this approach is that different methodological approaches (e.g., naturalistic observation and controlled experimentation) and different types of data (e.g., verbal reports and performance measures) have different strengths and weaknesses, thus complementing each other when used jointly (Ericsson & Simon, 1980; Patton, 1980).

Moher and Schneider (1982) note that task selection and subject selection need to be carefully considered. The task or tasks used in the research should approach the complexity of those tasks to which the research is to be generalized. It should not be overly simplified. Comparison of computer-initiated dialogue (e.g., menu selection) and user-initiated dialogue (e.g., command language) using a task with only three commands will not be satisfactory. In addition, subject selection based on who is available should be replaced by selection from the population or populations of interest. If the subjects available are computer novices and the research question sensibly concerns experts, the outcome will not have general interest.

Finally, important extraneous variables should be identified and controlled in a manner that will not limit generality of the results. More use of control by randomization or by systematic variation rather than control by holding constant may sometimes be effective. As an example, assume that research is to be conducted to compare two different ways of preparing user manuals (e.g., structured format vs. narrative). If a single version is prepared for each method (identical except for structure), a finding in favor of one type of manual might be a special case. However, if different manuals are produced in both structured and narrative form and one form is generally superior, the results will be taken much more seriously.

Choice of task, subjects, and materials affect the validity of the research design (Campbell & Stanley, 1963). Traditional psychological research emphasizes internal validity, the extent to which a relationship between two variables is shown to be causal. To be taken seriously by practitioners, researchers must attempt to increase the external validity of their designs, the extent to which the results generalize to other settings, persons, and operations. The challenge is to increase external validity without a corresponding sacrifice of internal validity.

PRODUCT DEVELOPMENT EVALUATION

It is time for psychologists to assume a greater role in the production, implementation, and evaluation of computer systems. There is increased awareness that computer systems designed for naive and discretionary users will not be successful unless they are reasonably easy to learn and use. As a result, there are increasing demands that usability criteria for computer systems be established by designers in the same manner that performance criteria are now established for the system itself. In addition to specifying minimum response time and minimum down time, the maximum training time and the acceptable error rate for the intended users may soon be specified (Bennett, 1979). Software tools are being developed to separate the task of diaglogue design from the actual design of a system's functions (Feldman & Rogers, 1982; Roach, Hartson, Ehrich, Yunten, & Johnson, 1982). Thus, a "dialogue specialist" with little understanding of the technical specifications of the system can work on developing, testing, and improving the interface between user and system independently of the programmers and designers. The shortage of experienced computer scientists and engineers dictates that their talents should be concentrated on tasks for which they are trained. The design of equipment and software must be their major concern.

The methodologies needed for product testing are considerably different from those needed for guideline research. Instead of emphasizing carefully controlled experimentation to determine causal variables and their interaction, the empirical work should be designed to find out whether or not users are able to use the prototype as intended, and if not, to identify the revisions that need to be made. It is imperative that the methods be fast, flexible, and inexpensive.

A confusion between the type of research required to provide guidelines and the type needed to test a prototype has led psychologists given an opportunity to participate in product evaluation to undertake research that is too slow and limited for the development process. For example, Wagner (1980), who is in charge of producing manuals for IBM, rejects testing their manuals with users because of the methods he has observed: "The typical test is very specific and geared to a very small part of a manual . . . . It is not cost-effective to test more than a small percent of the manual . . . . Statistically significant test scores require too many test subjects" (p. 99).

Because the need for systematic and efficient methods of obtaining user feedback is great, there is a growing emphasis upon developing appropriate methodologies. For example, some researchers are developing systematic methods for testing manuals and other forms of instructions. Winbush and McDowell (1980) describe a systematic approach based on the use of a test coordinator, an individual charged with assessing the usability of manuals. This highly structured approach involves three parts: (1) manual reading, in which subjects first identify sections they find difficult and then take a written test on the document content, (2) procedural testing, in which subjects attempt to actually perform tasks described in the manual in a simulated environment, and (3) attitudinal questionnaire administration to determine the degree to which users find the manual to be satisfactory.

Less quantitative and structured methods, which take less time and may be equally effective, are also available for testing computer systems. For example, Atlas (1981) defines a "user edit" in the following manner: "The user edit is based on a very simple idea: Find someone who knows nothing about your machine and have him work with it, using only your manual as a guide; his errors and hesitations should tell you where the weak points are. To get more detailed information, it is also a good idea to ask users to talk while they work, telling what they are trying to do, what they are looking for, what gives them trouble, and what they suggest to make the manual better" (p. 28). Testing takes place with a very small number of users. Often, modifications are suggested after testing only one or two users. These methods are gaining increased recognition as being practical in the time required for system development and for leading to improvement in systems. Buxton and Sniderman (1980) refer to the process of test/modify/test as design interation. It is similar to the "user trial" proposed for product evaluation in ergonomics (Johnson & Baker, 1974; Rennie, 1981; Thompson & Rath, 1974).

Two methodological notes should be kept in mind in product development evaluation. First, Marcel and Barnard (1979) report that there are major differences in the reading patterns of subjects asked to explain what instructions meant without using the machine (or a simulated version) and those who actually tried to use it. The first group systematically read through the instructions. The second group tended not to use the instructions in order but to jump around, and these users often tried things on the machine without referring to the instructions. Lewis and Mack (1982) noted a similar problem with secretaries using a tutorial for a word processor. They tended to jump around in the manual, even though it had a carefully planned order. Early testing is valuable to alert designers that a particular dialogue or manual is too complex for the user, but finding that users are able to follow the instructions in this form does not guarantee that they will follow them properly when actually using a system. Thus, to get an accurate impression of how a manual will be used, testing should also involve actual interaction with the machine as soon as possible. There is a growing emphasis by designers on having the equipment and solftware available to design inexpensive and easily modified prototypes (Green, 1982).

Second, Dillon and Tombaugh (1982) note that user testing in a laboratory environment was excellent for insuring that users could follow the instructions. However, a test in the actual environment in which the device was to be used showed that the intended users (who had expressed satisfaction in the laboratory) did not like the material when it was presented in the field. Thus, it may be that user tests to measure user satisfaction will require testing in the user's natural environment (i.e., field trials). This does not mean that the laboratory work on clarity should be abandoned. Arndt (1977) notes the limitations of laboratory testing but sees them as needed preliminary work before testing in a naturalistic setting can be started. The field trial that starts with a system that can be used but is not particularly well liked can more easily be modified to a user's satisfaction than one that is not understandable to the user at the time it is introduced in the field.

In summary, techniques are being developed to insure that computer prototypes are usable by the group for whom they are intended. These techniques can be employed with minimum time and effort. However, there are few data indicating whether findings from user trials are generally reliable or highly dependent on intuitions and perceptiveness of an experienced researcher. As noted by Rennie (1981, p. 163), "overall there have been comparatively few papers published on consumer product evaluation. One reason for this may be that although ergonomics studies are being conducted in the area, few researchers feel their work would be of general interest and may feel their methods would not stand up to rigorous scientific examination." However, as qualitative evaluation methods gain in rigor (Patton, 1980) and as videotape and tape recorders increase the accuracy of observation, it should be possible to gain a direct estimate of the reliability and validity of the data collected.

FIELD EVALUATION

There is a growing emphasis upon evaluation of field trials when new technical products are introduced into a working environment. For example, the Canadian Department of Communications is organizing multimillion dollar field trials of integrated office equipment to evaluate the impact of the equipment upon a variety of outcome measures. In addition, field trials involving videotex systems are being conducted in a variety of locations (e.g., Aldrich, 1982).

Although the term "evaluation" is applied to this type of methodology, as well as to that presented in the last section, there are important differences in the two types of research. The distinction is similar to that between formative and summative evaluation in the literature on social program evaluation (e.g., Rutman, 1980). Formative evaluation is research in service of an application. Data are collected to improve the service. For example, the user edit is used to improve a document. Summative evaluation is research in judgment of an application. A decision on whether or not to introduce office automation in other offices might depend on the outcome of a field trial. The preferred research method for summative evaluation is the true experiment, applied in a field study. This is difficult research to do well. The pioneering methodological paper of Campbell and Stanley (1963) has more recently been considerably expanded and updated by Cook and Campbell (1979) and represents the type of research called for in field trials. A knowledge of experimental and quasiexperimental designs and statistical treatment of quasiexperimental designs is of great benefit to those interested in evaluation of the introduction of new systems. For example, if routine records of an office keep track of number of dollars earned on a monthly basis, analysis of the interrupted time-series design (Cook & Campbell, 1979) will indicate whether the introduction of office automation changed either the level or the rate of growth in the earnings.

Evaluation assessment is another methodology that is of great value to the researcher involved in planning the evaluation of a field trial (Rutman, 1980; Wholey, 1977). Policy makers often state the expected outcomes of a new program in vague and general terms; "greater productivity" and "a better working environment" are typical examples. Unless the researcher can find acceptable operational definitions for such terms, the evaluation will not be effective. Evaluation assessment provides a list of steps for the researcher to follow in deciding on measures and the experimental design. For example, documents about the application (e.g., promotional material, requirement specifications, minutes of meetings) are used to develop a "rhetorical application model" that lists the various components of the application and their intended effects. For example, an office system might include a management information system to improve the quality of decision making, a messaging system to improve office communication, and a word processing system to increase productivity. The researcher will then interview the policy makers to find out if all of the components and effects of interest are identified and to attempt to determine measures for the expected effects. The interviews might lead the researcher and policy maker to agree that "improved office communication" will be represented by increased use of memos between managers and staff and by decreased union grievances. Typically, no measure is found for some effects and they are eliminated from the research design. For example, "improved quality of decision making" might fall into this category. The final experimental design is then determined, based upon the defined measures and the availability of appropriate control groups. While utilization of evaluation assessment procedures does not eliminate the possibility of poorly utilized evaluation, it does insure that the researcher is well acquainted with the application and increases the communication between researcher and policy makers.

In summary, psychologists working in applied research on the development of computer systems may be called upon to participate in a large variety of data collection activities for different reasons and with different criteria for success. Exposure to the traditional methods of human-factors research, to field research techniques, and to formative research techniques will provide the flexibility the psychologist needs to select the appropriate tools for each type of research task.

REFERENCE NOTE

1. Tombaugh, J. W., & Dillon, R. A. *A field experiment with a videotex system.* Unpublished manuscript, Carleton University, 1979.

REFERENCES

Aldrich, M. J. World review of private videotex. *Proceedings of Videotex '82.* Middlesex, England: Online, Ltd., 1982.

Anderson, P. V. The need for better research in technical communication. *Journal of Technical Writing and Communication,* 1980, **10**, 271-282.

Arndt, J. Laboratory studies and the laboratory paradigm of man: Confessions of an uneasy consumer researcher. *Journal of Consumer Policy,* 1977, **1**, 9-12.

Atlas, M. A. The user edit: Making manuals easier to use. *IEEE Transactions on Professional Communication,* 1981, **PC-24**, 28-29.

Barnard, P. J., Hammond, N. V., Morton, J., Long, J. B., & Clark, I. A. Consistency and compatibility in human-computer dialogue. *International Journal of Man-Machine Studies,* 1981, **15**, 87-134.

Benbasat, I., Dexter, A. S., & Masulis, P. S. An experimental study of the human/computer interface. *Communications of the ACM,* 1981, **24**, 752-762.

Bennett, J. L. The commercial impact of usability in interactive systems. In Infotech State of the Art Report, *Man/computer communication* (Vol. 2). Maidenhead, England: Infotech International, 1979.

Buxton, W. A. S., & Sniderman, R. Iteration in the design of the human-computer interface. In *Proceedings of the 13th Annual Meeting of the Human Factors Association of Canada.* Toronto: Human Factors Association, 1980.

Campbell, D. T., & Stanley, J. C. *Experimental and quasi-experimental designs for research.* Chicago: Rand McNally, 1963.

Card, S. K., Moran, T. P., & Newell, A. Computer text-editing: An information-processing analysis of a routine cognitive skill. *Cognitive Psychology,* 1980, **12**, 32-74.

Chapanis, A. Words, words, words. *Human Factors,* 1965, **7**, 1-17.

Cook, T. D., & Campbell, D. T. *Quasi-experimentation. Design and analysis issues for field settings.* Chicago: Rand McNally, 1979.

Denzin, N. K. *The research act.* New York: McGraw-Hill, 1978.

Dillon, R. F., & Tombaugh, J. W. Psychological research on videotex. *Behavior Research Methods & Instrumentation,* 1982, **14**, 191-197.

Ericsson, K. A., & Simon, H. A. Verbal reports as data. *Psychological Review,* 1980, **87**, 215-251.

Feldman, M., & Rogers, G. Toward the design and development of style-independent interactive systems. In *Proceedings of Human Factors in Computer Systems.* Washington, D.C: Association for Computing Machinery, 1982.

Green, M. Towards a user interface prototyping system. In *Proceedings of Graphics Interface '82.* Toronto: National Computer Graphics Association of Canada, 1982.

Johnson, E. M., & Baker, J. D. Field testing: The delicate compromise. *Human Factors,* 1974, **16**, 203-214.

Lewis, C., & Mack, R. Learning to use a text processing system: Evidence from "thinking aloud" protocols. In *Proceedings of Human Factors in Computer Systems.* Washington, D.C: Association for Computing Machinery, 1982.

MacDonald-Ross, M. Graphics in texts. In L. Shulman (Ed.), *Review of research in education* (Vol. 5). Itasca, Ill: Peacock, 1978.

Marcel, T., & Barnard, P. Paragraphs of pictographs. The use of non-verbal instructions for equipment. In P. Kolers, M. Wrolstad, & H. Bouma (Eds.), *Processing of visible language* (Vol. 1). New York: Plenum, 1979.

Marx, M. H., & Hillix, W. A. *Systems and theories in psychology.* New York: McGraw-Hill, 1963.

Mayer, R. E. The psychology of how novices learn computer programming. *Computing Surveys,* 1981, **13**, 121-141.

McConkie, G. W. Learning from text. In L. Shulman (Ed.), *Review of research in education.* Itasca, Ill: Peacock, 1978.

Moher, T., & Schneider, G. M. Methodology and experimental research in software engineering. *International Journal of Man-Machine Studies,* 1982, **16**, 65-87.

Moran, T. P. An applied psychology of the user. *Computing Surveys,* 1981, **13**, 1-11. (a)

Moran, T. P. The command language grammar: A representation for the user interface of interactive computer systems. *International Journal of Man-Machine Studies,* 1981, **15**, 3-50. (b)

Norman, D. Steps toward a cognitive engineering: Design rules based on analysis of human error. In *Proceedings of Human Factors in Computer Systems.* Washington, D.C: Association for Computing Machinery, 1982.

Patton, M. Q. *Qualitative evaluation methods.* Beverly Hills: Sage, 1980.

Ramsey, H. R., & Atwood, M. E. *Human factors in computer systems: A review of the literature.* Englewood, Col: Science Applications, 1979.

Rennie, A. M. The application of ergonomics to consumer product evaluation. *Applied Ergonomics,* 1981, **12**, 163-168.

Roach, J., Hartson, H. R., Ehrich, R., Yunten, T., & Johnson, D. DMS: A comprehensive system for managing human-computer dialogue. In *Proceedings of Human Factors in Computer Systems.* Washington, D.C: Association for Computing Machinery, 1982.

Rutman, L. *Planning useful evaluations. Evaluability assessment.* Beverly Hills: Sage, 1980.

Sanders, J., & Cunningham, D. A structure for formative evaluation in product development. *Review of Educational Research,* 1973, **43**, 217-236.

Sebrechts, M. M., Deck, J. G., & Black, J. B. A diagrammatic approach to computer instruction for the naive user. *Behavior Research Methods & Instrumentation,* 1983, **15**, 200-207.

Thompson, C. W. N., & Rath, G. J. The administrative experiment: A special case of field testing or evaluation. *Human Factors,* 1974, **16**, 238-252.

Wagner, C. Quality control methods for IBM computer manuals. *Journal of Technical Writing and Communication,* 1980, **10**, 93-102.

Wholey, J. S. Evaluability assessment. In L. Rutman (Ed.), *Evaluation research methods: A basic guide.* Beverly Hills: Sage, 1977.

Winbush, B., & McDowell, G. Testing: How to increase the usability of computer manuals. *Technical Communication,* 1980, **4**, 20-22.

Wright, P. Feeding the information eaters: Suggestions for integrating pure and applied research on language comprehension. *Instructional Science,* 1978, **7**, 249-312.