Contemporary Usage of CASE Tools in U. S. Colleges and Universities

Susan J. Chinn
School of Business
University of Southern Maine
Portland, ME 04104, USA
schinn@usm.maine.edu

Scott J. Lloyd
Eric Kyper
College of Business Administration
University of Rhode Island
Kingston, RI 02881, USA
sjlloyd@uri.edu; ekyp3036@postoffice.uri.edu

ABSTRACT

CASE tools have been incorporated into Information Systems curriculums for years. Curriculum guidelines in both disciplines call for the use of CASE tools. This paper describes the findings of a study of U.S. college and university information technology programs to determine in which classes CASE was taught, the extent to which the tools were being used, the degree of coverage of tools' functional aspects, and reasons why some academics were not using CASE. The results, which confirm continued CASE usage in academia in accordance with the guidelines, present implications for both the quality of the tools as perceived by academics, as well as their continued use. We note that, at present, there is no academic consensus on choice of a CASE tool, and that the drawing features of CASE tools seem to be used more heavily than other more complex and powerful capabilities. Based on the survey results we conclude that schools should reevaluate their teaching curricula and model curriculum guidelines to justify the importance of teaching and using CASE tools.

Keywords: CASE Tools, Systems Analysis and Design, Teaching Methodology, CASE Tool Selection, Model Curriculum Guidelines

1. INTRODUCTION

CASE (computer-aided software engineering), refers to the development of information systems using automated software tools to support the phases of software development from project selection to maintenance (Hoffer, George, and Valacich, 2002). The tools themselves are typically categorized by functionality as: upper CASE, middle CASE (often part of upper CASE), and lower CASE, as well as integrated CASE. Upper CASE tools are used for planning and business modeling; they support early iterations of project selection, planning, and analysis. Such tools usually include diagramming capabilities and business rule documentation. Middle CASE tools focus on the documentation of new and existing systems in the analysis and design stages, including data or object modeling, data dictionaries, and analysis tools (e.g., consistency checking). Lower CASE tools are used to implement, test, and maintain the system and include code, form, and report generators, and other utilities. Many CASE tools combine these functionalities and are classified as integrated CASE tools, and would include project management capabilities as well (McLeod and Jordan, 2002).

CASE tools have been incorporated into Information Systems curriculums for years. Model curriculum guidelines, as described below for Information Systems (MIS/CIS) programs, incorporate tool evaluation, selection, and usage in courses relating to systems analysis and design and software engineering. In at least one set of guidelines, CASE usage is explicitly advocated. This paper describes the findings of a study to determine in which classes CASE was taught, the extent to which the tools were being used, the degree of coverage of tools' functional aspects, and reasons why some academics were not using CASE.

Competency in the use of CASE tools is a priority stated in model curriculum guidelines for schools offering MIS/CIS.

The IT Dean's Council, whose mission includes defining key knowledge areas for IT, calls for a deep level of understanding of systems development tools and techniques as being essential for undergraduate programs in MIS/CIS and other related programs (Landry, Pardue, Longenecker, and Feinstein, 2003). In the business area, the IS 2002 model curriculum guidelines suggest using CASE both in the Analysis and Logical Design course (IS 2002.7) and in the Physical Design and Implementation with DBMS course (IS 2002.8) at the undergraduate level. The analysis and design course guidelines stress that while automated tools are not a substitute for understanding business processes, CASE can reinforce them, and can help to ensure correct use of a methodology. CASE usage is specifically mentioned in course goal 75, to develop logical designs, and is implied in course goal 73, to give students exposure to commercial program products. At the design and implementation level, the guidelines state that automated tools may be used to design and implement systems. Goal 93 suggests that students develop skills in using code generators, while goal 95 implies the use of CASE when designing conceptual and logical data models, converting the models to a physical DBMS, and for test data generation (Gorgone, Valacich, Topi, Feinstein, and Longenecker, 2003). At the graduate level, the Model Curriculum and Guidelines, a joint project of the ACM and AIS, again recommend use of current tools in MSIS 2000.2, Analysis, Modeling, and Design (Gorgone and Gray, 2000). Although CASE is not specifically mentioned, the implication here is that this course would use a CASE tool.

The balance of the paper is organized as follows: in the next section, we summarize the literature on using CASE tools in academia. We then describe the method for collecting data from a web-based survey, and discuss the results. We also draw some conclusions and raise questions for continued usage of CASE.

2. LITERATURE REVIEW

Research focusing on CASE tools in information systems and computer science education has tended to be largely descriptive – that is, presentations of various successful deployments in analysis and design projects. Researchers found that student projects which used CASE tools tended to produce higher quality software projects (Granger and Pick, 1991; Mynatt and Leventhal, 1990; Sidbury, Plishka, and Beidler, 1989). Students also received the benefit of being exposed to commercial tools (Seburn, 1997), as well as to an integrated approach to systems development (Granger and Little, 1996; Grove, 1997). Positive results were experienced despite drawbacks such as funding for software, learning curves for faculty and students, poor documentation or user interfaces, hardware and networking problems, and compatibility with other software (Granger and Little, 1996; Mynatt and Leventhal, 1990; Seburn, 1997; Sidbury et al, 1989). It was also noted that CASE was often introduced in the software engineering or analysis and design courses, which are both content- and project-intensive, without additional time devoted to learning the tool itself (Grove, 1997). The learning curve issue may be fairly serious; Gill and Hu (1998) reported that although MIS/CIS majors should be expected to have a high degree of competence in using CASE, faculty rated themselves as weak in that competence and that therefore CASE tool usage in MIS/CIS programs was below what might be expected or suggested by the curriculum guidelines. Some faculty members have anecdotally cited similar issues in the decision to use CASE tools, such as the learning curve, funding, maintenance, and accessibility of the tools to students.

One study presented some interesting pitfalls to using CASE in teaching software engineering. Students using a CASE tool for a project believed that their models were correct simply because the tool accepted their model even though their methodology was suspect (Eriksen and Stage, 1998). Students also focused on aspects of the project supported by the CASE tool while neglecting those that were not supported. However, it was easier to iterate through designs and produce more documentation. The researchers found support in a university setting for the premise set forth by the Capability Maturity Model in a professional setting (Humphrey, 1989) -- CASE tool usage should be introduced only after a certain level of competency with the underlying methodology has been achieved.

3. METHOD

We directly contacted via email 304 college professors in MIS/CIS departments in the United States found on the ISWorld list of IS departments (http://juliet.stfx.ca/~rmackinn/infosys/isgroups.html) with a background in analysis and design, and from the ACM Crossroads web site (http://www.acm.org/crossroads/resources/list.html). The authors attempted to contact all professors that conceivably teach analysis- or design-related or software engineering courses as described in the curriculum guidelines. All colleges and universities were contacted without regard to the presence of a graduate program. We asked them to select a link on the email to a web-based survey hosted at our university site. We received 68 responses for a response rate of 22%; however, because only seven responses were received from Computer Science programs associated with math departments in Colleges of Arts and Sciences, these were dropped from the analysis. This yielded a final sample of 61 useable responses for a response rate of 20%. The survey is included as Appendix I.

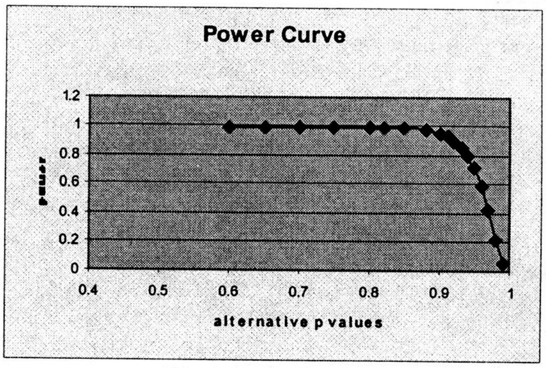


Figure 1: Power Curve

With only a 20% response rate one must be concerned with the validity of the sample and the sampling process. Toward that end we attempted to demonstrate that the sample is representative of the test population. The distribution of the sample was compared to the distribution of the general population of American Association of Collegiate Schools of Business (AACSB) to determine the successfulness of a representative stratified sample. In our sample 100% of the business schools surveyed had a CIS/MIS program, while 96% of the AACSB population had a MIS/CIS program. This is not a significant difference at the 0.01 level. Likewise, 66.2% of the schools in the sample reported having a graduate program, while 76% of the population offered graduate programs. In this case the sample falls within a 95% confidence interval. Thus we can conclude no response bias and a valid representative sample (Freedman, Pisani, and Purves, 1978) Figure 1 shows a power curve for our test and sample size. Power for an alternative p value of .96 is .59 which demonstrates that if significant differences exist between our sample and the overall population we should have been able to detect them.

A limitation to the research concerns the pitfalls of collecting data by using Web surveys. Web surveys are particularly subject to the problems of coverage and non-response (Couper, 2000). Coverage error, as described by Couper, results from an inconsistency between the target population and the frame population. In our case the target population was all academics teaching systems analysis and design, software engineering, or equivalent courses. However, the frame population was two published lists from the ACM and ISWorld, neither of which could be said to be comprehensive. Therefore the coverage error occurs because we could not obtain a complete list with email addresses. Attempts to contact providers of such a list were not successful. However, since the sample is not significantly different than the AACSB population, coverage bias is not an issue in this survey. Non-response error is also a factor in Web surveys. Again, according to Couper (2000), non-response error occurs when not all the individuals contacted are willing or able to submit their responses to the survey. We did have several instances where willing participants could not complete the survey due to technical problems. We also had some returned messages (e.g., no such email address). Those who were contacted and did not respond may also have had concerns about spam, privacy, or simply lacked the time and interest.

4. RESULTS

Survey participants were asked to respond to questions concerning the discussion and use of CASE tools in their courses. The CASE tools referred to in the survey are outlined in Table 1.

These CASE tools were chosen so as to be representative of what universities are likely to use or discuss. Each tool (except for UML) is actually a suite of tools that covers a range of features, typical of integrated CASE. Common to each tool is application design and database design functionality. This is important because (especially at the undergraduate level) faculty are unlikely to teach CASE tool usage for specific functions such as business intelligence systems design or strategic planning design. UML is included because it is a standard associated with many CASE tools and is likely to be discussed or taught separately from a specific tool. With the exception of UML, each of these products is primarily marketed towards large corporations working on enterprise level projects. There are many other CASE tools that could have been included in this table but for simplicity we chose common tools that have had fairly large market shares and provide functionality that is likely to be covered in a MIS/CIS program.

CASE Tool	Core Features
Excelerator	Primarily used for low front end application design
Oracle Developer	Suite of CASE tools for business intelligence, database, and data warehouse design, and application development
Power Designer	Suite of tools for application modeling through UML, business process modeling techniques and traditional database modeling techniques
Visible Analyst	Suite of tools for strategic planning, data modeling, UML modeling, and process modeling
Rational Rose	Suite of products including functionality for data modeling and application design
UML	Unified Modeling Language: A standard for modeling software artifacts

Table 1: CASE Tools Included in Survey

Most respondents indicated that they currently used Oracle Developer (18%), Visible Analyst (19%), or Rational Rose (22%). Past use of Oracle's Developer, Visible Analyst, and Rational Rose were reported at 34%, 32%, and 22% respectively. This decrease in usage may reflect the change in methodology as object-oriented concepts have been more widely introduced into systems analysis and design courses in the past few years. In 1999 it was suggested that by 2003 42% of all development would be object-oriented (Orenstein, 1999). Interestingly, respondents also reported a drop in UML as well, from 19.7% to 14.7%. Faculty members have had to learn both new concepts and adapt to new CASE tools in support of the new development paradigm. Therefore, the drop in usage is consistent with previous research which reported learning curves for faculty (Gill and Hu, 1998, Granger and Little, 1996).

No respondent listed Excelerator as a tool currently used; however, several reported past use. Power Designer was cited in past/current usage by only one respondent. In the "Other" category, respondents listed using Microsoft Visio, Together, Sterling Cool, MS Project, Oracle Designer, Erwin, and Embarcadero's "Describe."

Eighty-two percent of the sample discussed the role of CASE tools, while 67% actually used the tools in the courses. As expected, CASE tools were discussed most frequently in systems analysis and design courses, followed by database courses, and were also used most frequently in these same courses. Table 2 shows the percentages of both undergraduate and graduate classes where CASE tools were discussed, and those in which they were actually used. Respondents were allowed to answer multiples times for each category.

We asked respondents not using CASE tools to select from a list some reasons that they were not used; in addition, we provided an open-ended area for them to respond to the question. Our choices were:

- No clear industry preference for a particular tool
- Not used enough in industry
- Expense of the tools
- Does not add anything to the academic experience
- Not enough time in class to adequately present the information
- Other (please specify)

The reason cited most often for non-use was the class time constraint, followed by the belief that the tools did not add any value to the academic experience, and finally by the expense of the tools. The sole response under "Other" was that the tools were "not useful."

Classes which discuss CASE	**% Undergraduate**	**% Graduate**
Introduction to CS/MIS	36.0	26.2
Application Development	27.8	22.9
Systems Analysis and Design	67.2	52.5
Database	39.3	31.1
Project Management	22.9	18.0
E-commerce	6.6	6.6
Other (e.g., semester or field project, decision support systems)	3.3	1.6
Classes which use CASE	**% Undergraduate**	**% Graduate**
Introduction to CS/MIS	1.6	1.6
Application Development	16.4	14.7
Systems Analysis and Design	52.5	45.9
Database	29.5	26.2
Project Management	6.6	3.3
E-commerce	3.3	3.3
Other (e.g., semester or field project, decision support systems)	4.9	3.3

Table 2: Percentages of Classes That Discuss and Use CASE Tools

Use of the current CASE tool in the classroom varied widely, from being used primarily as a drawing tool to being used throughout the systems development process. However, these results might depend on the course in which CASE was being used. Of the respondents using CASE, 67.5% also noted that the same tool was used in more than one class, but not necessarily to provide continuity. None of the respondents taught classes strictly about CASE tools, and there was no interest in offering such a class (only 1 positive response). Table 3 shows the extent to which CASE tools were used; the responses are based on a Likert scale where 1 = CASE usage throughout the entire systems development process, 2 = CASE usage throughout most of the systems development process, 3 = CASE usage throughout some of the systems development process, 4 = little CASE usage in the systems development process, and 5 = CASE usage primarily as a drawing tool. The table shows that the tendency is towards lighter usage, especially as a drawing tool, rather than for a complete systems implementation.

Extent used	1=entire process	2	3	4	5=drawing
	21.2%	9.1%	24.2%	24.2%	21.3%

Table 3: Extent of CASE Usage

Table 4 presents the percentages of the respondents' ranking of how heavily different functional aspects of the tool were used across courses in their programs. If a tool was used in more than one course, respondents were asked to select the rating that would apply to the course in which the tool was used the most. The responses are again based on a Likert scale where 1 = lowest coverage, 2 = light to moderate coverage, 3 = moderate coverage, 4 = moderate to heavy coverage, and 5 = highest coverage. For example, a response of 1 under "Report generation" meant that this aspect of the tool was not covered across any courses where the tool was used.

The results show no clear consensus about which tools are used or how they are used. This may be due to some of the weaknesses of these tools (Sharma and Rai, 2000). One may hypothesize that they have value in several areas but that some of the features require more time to master than is possible in a semester. Further examination of the results shows that heavy users were more likely to use the CASE tools for drawing diagrams than using the other features, which may lead to the conclusion that drawing is one of the most accessible and easily used features of any of these tools.

Some CASE tools come with tutorial materials, but only 23% of the respondents were using them. When asked to explain why they were not being used, some of the respondents stated that the software was so easy to use that the tutorials were unnecessary; others stated that they spent class time explaining the tools, or had created their own learning materials; still others said that the materials were too lengthy.

Upper-CASE functionality	1= Lowest	2	3	4	5=Highest
Data flow diagramming	27.6	6.9	17.2	20.7	27.6
ER Diagramming	16.1	3.2	16.1	25.8	38.7
Workflow diagramming	25	20.8	12.5	25	16.7
Object oriented diagramming	31	0	20.7	13.8	34.5
Data dictionary	22.2	11.1	22.2	14.8	29.6
Business rule adoption	33.3	16.7	25	12.5	12.5
Analysis tools	7.7	38.5	11.5	23.1	19.2
Lower-CASE functionality	**1= Lowest**	2	3	4	**5=Highest**
Code generation	47.8	21.7	0	17.4	13
Form generation	26.1	13	13	34.8	13
Report generation	27.3	4.5	13.6	40.9	13.6
Document generation	23.8	9.5	14.3	33.3	19
Import and export utilities	38.1	23.8	19	14.3	4.8

Table 4: Percent Coverage of Functional Aspects of the Most-Used CASE Tool

5. DISCUSSION AND FUTURE RESEARCH

From the above results, we can infer that many academics are using CASE tools in their curricula, which supports the model curriculum guidelines. Not surprisingly, the results show that analysis and design and database courses use CASE tools the most. The choice of tool has also changed in favor of object-oriented tools, which is consistent with the shift in methodology in the field. The results also show a range of usage of the tools, from primarily being used for diagramming, to complete integration throughout the systems development process. Few respondents appear to be using the tutorial materials. However, if we can infer from the results that non-coverage of some functional aspects means that the tools are deficient in those areas, then perhaps these results should signal a message to the software development community that CASE tools seem to fall short in many areas, such as object-oriented diagramming.

It might be expected that CASE tool usage learned in an academic setting would lead to increased application of CASE in real-world development environments. Although it may be possible to conclude that the use of any specific tool may not lead to the use of that tool in industry, students will still have benefited from the exposure to development methodologies associated with the tool. Businesses that use CASE tools for large system development projects report a high level of satisfaction with them and conclude that their use is essential (McNurlin and Sprague, 2004). Limayem, Khalifa, and Chin (2004) confirmed this through a survey that examined the relationship between use of CASE and its effect on systems development projects.

As the reader can see by the results, a large percentage of colleges and universities are using CASE tools in their classes; however, there is no clear choice of which tools are being used or how they are used. Does this lack of consensus among academics also suggest that we, ourselves, are not yet convinced of the worth of these tools despite evidence to the contrary? Schools should continually reevaluate their courses to justify the inclusion of these tools, or make a more pervasive case for their use and reach agreement of their necessity in the curriculum. In addition, model curriculum guidelines in MIS/CIS programs need to be re-evaluated. If we are preparing students for real world applications, can we rationalize the time and effort spent teaching CASE? Or, does academia require more support from industries that do use CASE in order to improve the tool selection and education process?

6. REFERENCES

Couper, Mick P. (2000), "Web Surveys: A Review of Issues and Approaches," Public Opinion Quarterly, Vol. 64, No. 4, pp. 464-494.

Eriksen, Lars B., and Jan Stage (1988), "A Qualitative Empirical Study of CASE Tool Support to Method Learning," Information and Software Technology, Vol. 40, No. 5-6, pp. 339-345.

Freedman, D., R. Pisani, and R. Purves (1978), Statistics. Norton, New York.

Gill, T. Brandon, and Qing Hu (1988), "Information Systems Education in the USA," Education and Information Technologies, Vol. 3, No. 2, pp. 119-136.

Gorgone, John T., Gordon B. Davis, Joseph S. Valacich, Heikki Topi, David L. Feinstein, and H. E. Longenecker (2003), "IS 2002: Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems" [special issue], The Data Base for Advances in Information Systems, Vol. 34, No. 1.

Gorgone, John, and Paul Gray (2000), "MSIS 2000: Model Curriculum and Guidelines for Graduate Degree Programs in Information Systems: Report of the Joint ACM/AIS Task Force on Graduate IS Curriculum" [pdf file], Communications of the AIS, Vol. 3, Article 1, February 2000. Retrieved August 5, 2005 from http://cais.aisnet.org/contents.asp

Granger, Mary J., and Joyce C. Little (1996), "Integrating CASE Tools into the CS/CIS Curriculum," SIGCSE Bulletin, Vol. 28, No 5, pp. 130-132.

Granger, Mary J., and Roger A. Pick (1991), "The Impact of Computer-Aided Software Engineering on Student Performance," SIGCSE Bulletin, Vol. 23, No. 1, pp. 62-72.

Grove, Ralph F. (1997), "A Proposal for Integrated Software Engineering Education," SIGCSE Bulletin, Vol. 29, No. 2, pp. 11-16.

Hoffer, Jeffrey A., Joey F. George, and Joseph S. Valacich (2002), Modern Systems Analysis and Design, 3rd ed., Prentice Hall, New York.

Humphrey, W. S. (1989), "Case Planning and the Software Process," Technical Report CMU/SEI-89-TR-26, Software Engineering Institute. Pittsburgh, PA.

Landry, Jeffrey P., Harold J. Pardue, Herbert E. Longenecker, and David F. Feinstein (2003), "A Common Theme for IT Degree Programs," Communications of the ACM, Vol. 46, No. 11, pp. 117-120.

Limayem, Moez, Mohamed Khalifa, and Wynne Chin (2004), "CASE Tools Usage and Impact on System Development Performance," Journal of Organizational Computing & Electronic Commerce, Vol. 14, Issue 3, pp. 153-175.

McLeod, Raymond, Jr. and Eleanor Jordan (2002), Systems Development: A Project Management Approach, Wiley, New York.

McNurlin, Barbara and Ralph Sprague (2004), Information Systems Management in Practice, 6th ed. Prentice-Hall, New York.

Mynatt, Barbee T., and Laura M. Levanthal (1990), "An Evaluation of a CASE-Based Approach to Teaching Undergraduate Software Engineering," SIGCSE Bulletin, No. 1, pp. 48-52.

Orenstein, David, (1999), "Java, Visual Basic Seen as Languages of the Future," Computerworld, Vol. 33, No. 13, March 29, 1999, p. 6.

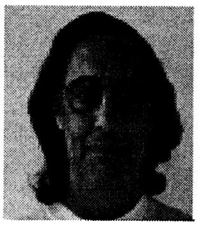
Sharma, Srinarayan, and Arun Rai (2000), "CASE Deployment in IS Organizations," Communications of the ACM, Vol. 43, No. 1, pp. 80-88.

Seburn, Mark J. (1997), "Iterative Development and Commercial Tools in an Undergraduate Software Engineering Course," SIGCSE Bulletin, Vol. 29, No. 1, pp. 306-309.

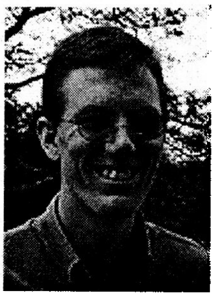
Sidbury, James R., Richard Plishka, and John Beidler (1989), "CASE and the Undergraduate Curriculum," SIGCSE Bulletin, Vol. 21, No. 1, pp. 127-130.

AUTHOR BIOGRAPHIES

Susan J. Chinn is an Assistant Professor of MIS at the University of Southern Maine. She holds a Ph.D. from Kent State University, and an M.S. from Virginia Commonwealth University. She spent several years in industry as a programmer/analyst and consultant. Dr. Chinn has taught systems analysis, systems design, expert systems, programming languages, and introductory undergraduate and graduate MIS survey courses. Her research interests include information technology in nonprofit organizations, organizational use of the Internet, and tools and techniques for systems analysis and design.



Scott J. Lloyd is an Associate Professor of MIS at The University of Rhode Island and received his degree in Management Information Systems from Kent State University. He teaches in the area of databases, knowledge management and system design. Dr. Lloyd has had his research published in The Journal of Database Management, Industrial Management, Midwest Decision Sciences Proceedings, Simulators International and Modeling and Simulation. He serves on the Editorial board of The Journal of Database Management and as ad-hoc reviewer of The Journal of Electronic Commerce in Organizations. He has been Track Chair of Computers and Information Systems for North East Decision Science Proceedings. Since his appointment to the URI faculty he has published in Industrial Mathematics, Software Engineering in the 21st Centaury, Journal of Electronic Commerce in Organizations, North East Decision Sciences Proceedings, National Decision Science Proceedings and Information Resources Management Association National Proceedings and was the co-editor of Software Engineering in the 21st Century. Lloyd also worked in industry as an MIS consultant, DBA and research technologist. His research is in the areas of database management, optimization of file access for database management systems, knowledge management, data mining and system analysis and design.



Eric Kyper is a Ph.D. student in management information systems at the University of Rhode Island. He has taught classes in the use of basic and advanced computing applications and introductory information systems. His research has appeared in Applied Economic Letters, Advances in Business and Management Forecasting, and proceedings of AMCIS, DSI, IRMA and several other conferences. Currently he serves as a reviewer for Informing Sciences. Current research interests focus mainly on the area of data mining.

APPENDIX I.
Survey Measurement Instrument

Q1: Please select your department type: □ CS (Computer Science) □ MIS (Management Information Systems)
Q2: Does your college offer a Master's program in your area? Yes/No
Q3: Does your college offer a Ph.D. program in your area? Yes/No
Q4: Do you discuss CASE tools in your CS/MIS program? Yes/No
If Yes, please go to question 7. If No, please go to question 5.
Q5: Please select the three reasons you do not use CASE.
Rank them as 1-Most important, 2-next important, etc.

	Expense of the tools.
	No clear industry preference for a particular tool.
	Not used enough in industry.
	Does not add anything to the academic experience.
	Not enough time in the class to adequately present the information.
	Other (please specify)

Q6: Please rank order the three most important reasons you do not discuss CASE.
Rank them as 1-Most important, 2-next important, etc.

	Does not add anything to the academic experience.
	Not enough time in the class to adequately present the information.
	Other (please explain)

Q7: In which classes do you <u>discuss</u> the use of CASE tools? Please select all that apply.

Introduction to CS/MIS		Undergraduate		Graduate
Application development		Undergraduate		Graduate
Systems analysis and design		Undergraduate		Graduate
Database		Undergraduate		Graduate
Project management		Undergraduate		Graduate
E-commerce		Undergraduate		Graduate
Other (please specify)		Undergraduate		Graduate

Q8: Do you use CASE tools in your CS/MIS program? Yes/No
Q9: In which classes do you <u>use</u> CASE tools? Please select all that apply.

Introduction to CS/MIS		Undergraduate		Graduate
Application development		Undergraduate		Graduate
Systems analysis and design		Undergraduate		Graduate
Database		Undergraduate		Graduate
Project management		Undergraduate		Graduate
E-commerce		Undergraduate		Graduate
Other (please specify)		Undergraduate		Graduate

Q10: Which tools <u>have</u> been used in the CS/MIS program in the past 5-10 years?

Excelerator	
Oracle Developer	
Power Designer	
Visible Analyst	
Rational Rose	
UML	
Other (please specify)	

Q11: Which tools are <u>currently</u> being used in the CS/MIS program?

Excelerator	
Oracle Developer	
Power Designer	
Visible Analyst	
Rational Rose	
UML	
Other (please specify)	

The following questions apply to the current tool used most heavily in the CS/MIS program.

Q12: For each CASE tool attribute below, rate the degree of functionality of the current tool that is used the most in your CS/MIS program.

Upper CASE functionality	**Lowest**				**Highest**
	1	2	3	4	5
Data flow diagramming					
ER diagramming					
Workflow diagramming					
Object-oriented diagramming					
Data dictionary					
Business rule documentation					
Analysis tools (consistency checking, etc.)					
Lower CASE functionality	**Lowest**				**Highest**
	1	2	3	4	5
Code generation					
Form generation					
Report generation					
Document generation					
Import and export utilities					

Q13: Is a tutorial available for the tool? Yes/No

Q14: Do you use the tutorial for the tools? Yes/No
If Yes, please go to question 16. If No, please go to question 17.

Q15: Why is the tutorial not used for the tool?

Tutorial has errors in it	
Tutorial is confusing	
Tutorial takes too long to complete	
Other (please specify)	

Q16: Select the number which closest represents the extent to which the CASE tool is used in any analysis and design class.
1= Entirely throughout the system design process
5= Primarily as a drawing tool

Extent used	1	2	3	4	5
Select one:					

Q17: Is the same CASE tool used in more than one class? Yes/No
If Yes, please go to question 18. If No, please go to question 19.

Q18: If the CASE tool is used in more than one class, do you use the tool to provide continuity from class to class?
Yes/No

Q19: Is any class strictly about the use of CASE tools? Yes/No

Q20: Would you like to offer a class strictly in the use of CASE tools? Yes/No

Q21: What is the approximate number of students in your CS/MIS undergraduate program?

Q22: If you have a graduate CS/MIS program, what is the approximate number of students?

iscap
Information Systems & Computing
Academic Professionals

EDSIG
Serving Information Systems Educators

STATEMENT OF PEER REVIEW INTEGRITY

All papers published in the Journal of Information Systems Education have undergone rigorous peer review. This includes an initial editor screening and double-blind refereeing by three or more expert referees.

ISSN 1055-3096