Technology in computing education: Yet another bandwagon?

Joe Turner
Clemson University
Clemson, SC 29670 USA
turner@cs.clemson.edu

1 Introduction

The title of this talk was inspired by David Kay's paper "Bandwagons Considered Harmful" in the December 1996 *SIGCSE Bulletin* [2]. In an email message to him, I stated that his was a paper that needed to be written at least every five years to remind us of the folly of rushing to adopt the latest fad (or application) in education. Two quotes from that paper that are especially relevant to today's talk are "Innovations have the potential to do more harm than good" and "We must take great care that our enthusiasm for novel approaches not lead us to omit something vital".

The purpose of this talk is not to provide magic answers to the question that is posed in the title. Rather, I hope it will provoke thought and discussion by taking a somewhat irreverent look at what we are doing with technology in computing education. We will consider the nature of bandwagons and other pressures to use technology in computing education, some effects that technology has had on computing education, past and current activities in the use of technology in computing education, potential effects of technology in computing education, some hindrances in using technology in computing education, and trends and opportunities for the future.

2 Bandwagons

Examples of bandwagons that have affected computing education during the past 25 years are structured programming, Pascal, abstract data types, object-oriented programming (and, to a lesser extent, C++), and the current one: Java. There have been others, but these were chosen because they illustrate well the characteristics and results that are of interest here.

One of the characteristics of a "bandwagon" is that there is a rush to adopt something for such reasons as "Everybody is doing it so we should be doing it too" or "It will solve all of our problems". A second characteristic is that the object of the bandwagon is espoused with religious zeal. Experience tells us (or should tell us) that there is no silver bullet that will solve all of our problems, and that there is a down side to everything so we shouldn't rush into anything without a careful analysis of the disadvantages as well as the advantages of doing so. Similarly, religious zeal usually clouds a reasoned and rational approach to decision making.

It should be noted that except for Java, which is too new to evaluate completely, the examples listed here have had mostly significant and positive effects on computing education. However, none delivered what many, if not most, of its proponents promised initially. (Note that the word "proponents" as used here does not necessarily include the originators of the concepts.) What initially appeared to be a "silver bullet" to its proponents turned out to be only another useful tool after it was incorporated into our educational programs.

3 Pressures to use technology in computing education

There have been increasing pressures to use technology in computing education. Some of these are pressures to use technology in education in general, while others are more specific to computing education. Education at all levels is being pressured to be more efficient, and the use of technology is often looked upon as a panacea to improve efficiency. In general, the use of technology to improve the efficiency of education seems to be the "right" thing to do.

Using technology in computing education especially seems like the right thing to do. We often hear that we exemplify the "cobbler's children" syndrome: just as the cobbler's children had no shoes, our students' do not benefit from having the technology that we teach used to improve the efficiency and effectiveness of our teaching and their learning.

So everyone seems to expect us to use technology in computing education. Just as with assessment (another hot topic today), we are expected (and sometimes even directed) to use it; however, it isn't clear how we are supposed to use it or what kind of results we are expected to achieve by using it.

Of course, we want to use technology (or anything else that is beneficial and cost-effective) to improve the effectiveness of education for our students. The problem is that just using technology is not always an improvement, especially not if we consider the cost of using technology.

4 Some effects of technology on computing education

Let's consider briefly the current state of affairs relative to the effect that technology has had on computing education. We

ITiCSE'97 Working Group Reports and Supplemental Proceedings
© 1997 ACM 1-58113-012-0/97/0010 …$3.50

restrict ourselves here to those things that have been fairly widely used. More recent applications that have not had widespread adoption are considered later.

One obvious effect that technology has had on computing education is its effect on the content of our courses. As the technology changes, so do our courses, because part of what we teach is about the technology. But this is not an effect that is of interest here. We are interested in how technology has affected the way that we teach, in how it has affected the effectiveness of our academic programs. Some of the more evident effects of technology on the way that we teach and the way that students learn are noted in the remainder of this section.

Two effects of dramatic improvements in computing price/performance ratios and the increasing availability of powerful personal computing platforms are that more programming can be done by students in a shorter period of time and that closed labs are more feasible and useful than was the case several years ago. Personal access to a computing platform and to productivity tools such as editors and debuggers have made it feasible for students to implement programs much faster than was the case before these things were widely available, and the availability of multiple personal computing platforms in a single room has made closed labs both feasible and potentially beneficial. Having students produce more (or larger) programs is not really innovative; it is merely a natural result of the advances in computing hardware and software. Implementing closed labs is a significant change and might even be considered innovative, but it has only added structure to what is otherwise available without closed labs (even though the added structure can be valuable to many, even most, students).

Improvements in graphics displays and the computer components that drive them have resulted in an increase in the use of animation and visualization in computing education (and in education in other disciplines as well). The animation of program execution, algorithm execution, and computing concepts is very useful in teaching computing and for students to explore and learn on their own.

More recently, advances in computer networks have facilitated improvements in communication among teachers and students. Electronic mail, bulletin board discussions, electronic submission of student work, and the distribution of course materials are all examples of ways that course management has been made more efficient through the use of computing (and communication) technology.

But somehow when we look at all that has happened it doesn’t seem like very much. We have not been able to use technology to effect significant changes in the way that we teach or the way that students learn. There have been many improvements that are significant, and the work that has been done to apply technology in computing education has been valuable. But for the most part, the way that we teach and the way that students learn has not changed in a significant and substantial way because of the application of technology. What we have done so far is mostly to *automate existing processes*, and not to use technology to enable new processes.

5 Recent activities in using technology in computing education

Activity in applying technology in computing education seems to be increasing, or at least broadening. For example, the number of papers on technology in computing education at the annual SIGCSE Technical Symposium for the past four years is shown in Table 1, with the number of web-oriented papers in parentheses

Given that the numbers for ’95 are more typical for prior years than are the numbers for ’94, this illustrates that there has been not only increasing activity in using technology in computing education, but also that the nature of the applications is broadening.

Several factors contribute to the increased activity. The most obvious is the increasing capability and availability of personal computing, especially in terms of computing power, graphics capabilities, and networks. The effective use of graphics and animation is heavily dependent on the availability of good quality graphics, and the availability and accessibility of the Web has opened many new opportunities for applications in education.

However, the general increase in pedagogical activity in computing education also plays a role. A cursory look through the past several proceedings of most any computing education conference will reveal that the papers and panels have shifted from focusing mostly on the subject matter in courses or a program to including papers and panels on pedagogical aspects of teaching computing. The application of technology in education is mostly a pedagogical issue, so this increase in pedagogical activity has had a synergistic effect on the use of technology in education.

6 Some potential benefits of technology in computing education

It seems natural to expect that the use of technology in education has the potential for significant benefits. Indeed, some of the results mentioned previously have been considered beneficial in a large number of programs. Looking into the future for radical effects that the use of technology might have on computing education seems pointless (because it seems that such predictions are rarely accurate). But some previous uses of technology and some current applications that are being tested or used on a small scale show promise for significant future benefits.

Many of the past uses of technology in computing education fall into the general category of *learning aids*. This includes

Year	**Animation/ Visualization**	**Other**
‘94	6 (0)	0
‘95	2 (0)	0
‘96	4 (1)	4 (3)
‘97	4 (1)	3 (3)

Table 1. Number of papers on technology in computing education at SIGCSE Technical Symposia, 1994–1997
(number of web-oriented papers given in parentheses)

drill-and-practice and other computer-assisted instruction applications, and it also includes self-paced instruction and other computer-managed instruction applications. The use of animation and visualization also fall into this category. These applications have generally been around longer than others, beginning 15–25 years ago.

More recently, several applications of technology that can be categorized as *class management* have been developed. This category includes such things as the electronic submission of student work, the management of student project teams, and communication among teachers and students. Many of these applications have been enabled by advances in computer networks and in communication software.

Some recent efforts have also been directed towards the category of *testing and evaluation*. This includes on-line testing, performance-based testing, and self-paced testing. Improved security and authentication mechanisms are making the on-line administration of tests feasible, and it is also now feasible to have students produce programs as a performance-based test outside of a normal controlled-testing environment. This addresses many problems, for example, by helping students who normally avoid developing even elementary software development skills by "hiding" in teams or by getting around the difficulty of having students write significant code on a test (which is normally unreasonable). This category also includes on-line assessment and feedback.

Course materials delivery is the final category that is mentioned here. With the rapid advancements in the Web there is increasing use of the Web for materials delivery. Coupled with the other categories, facilities exist that now make it much easier to deliver courses entirely over the Web. This has been done in several instances, not only for distance education on a geographic basis, but as an alternative to class attendance for on-campus students as well. An interesting experiment was the CALOS project at the University of British Columbia [1], which compared the performance of three groups of students in a single course that was offered both in a traditional mode and on the web. One group of students had access only to the traditional classroom version of the course, a second group had access only to the web version, and a third group had access to both versions. The group that had access to both performed the best relative to predicted performance based on previous courses; the lecture-only and web-only groups performed at about the same lower level. Interestingly, a smaller proportion of the web-only students performed worse than expected than in either of the other two groups. The characteristics of the experiment and results prevent drawing any conclusions, but the suggestions are promising relative to the benefits of web-based courses.

7 Hindrances

So why are we not making faster progress in achieving substantial benefits from the use of technology in education? (Some might argue that we have achieved substantial benefits and that progress has been quite rapid; but from my perspective we haven't really seen anything revolutionary and the total effect of the evolution has not been as significant as might be expected — or at least hoped for.)

The primary hindrance to progress in the effective use of technology in computing education is that it usually requires a tremendous amount of work to develop technology-based materials. In itself, this would not necessarily be a significant deterrent; however, there are at least two reasons why it is: 1) The work is often not amortized over several offerings of a course because of the need to continuously update the materials and 2) The reward structure at most universities and many colleges is such that no recognition or benefit results from the work. Things are getting better in that more and more tools are becoming available to help reduce the amount of work that is required. But the tools don't always do what is really needed and there is always a learning curve in using a new tool.

A second hindrance is that, due to inadequate access to appropriate facilities or insufficient capabilities in existing facilities, it is not always convenient for students to use technology-based materials. I have been using HTML tutorials that I developed for the languages portion of my programming languages course for the past couple of years, and I find that at least a third of the students either print out a copy or buy a printed copy of the tutorials and never use the on-line version, even though their printed versions lack page numbers and, of course, hyperlinks. Another third of the students use a printed copy extensively but also make some use of the on-line version, and the remaining third use the on-line version exclusively. The tutorials are oriented toward teaching-by-examples, with the examples to be worked by the students as they read through each tutorial. In investigating the reasons why at least two-thirds of the students did not use the tutorials as intended, the reasons mostly fell into the two categories mentioned above: they didn't have ready access from their residence to good computing facilities, or the facilities to which they did have access lacked a screen that was large enough or processing capabilities that gave sufficiently good performance for the demands of simultaneously using a browser and a language-development environment.

We expect that continuing advances in hardware and software will bring solutions to these hindrances. However, there may be a more fundamental issue: Is the main deterrent to really significant advances the fact that we have been mostly trying to automate existing processes? Traditionally, initial applications of a new technology simply automate existing processes, providing no new functionality but performing existing processes faster and more accurately. For example, the initial uses of computers in accounting and banking tried to replicate existing procedures, producing the same paper reports and other documents that had traditionally been done manually using calculators or similar devices. These applications were mostly quite beneficial, improving the speed and accuracy of financial operations and documents, but they did not change the financial processes in any significant way.

A similar case can be made for most applications of technology in computing (and other) education. For example, algorithm animations are very useful for efficiently demonstrating how an algorithm works, and interaction with an algorithm can be an effective device for student learning. But it is an automation of what we do without using computer-based animation and, although the speed with which an effective explanation can be given is significantly improved with a good computer-based animation, no new capability is really provided.

8 The future

Innovation often comes when technology enables a new process or provides new functionality. For example, the initial automation of existing processes in banking was eventually followed by a complete revision of the way that banks do business, using the enabling technology of computers and communication networks to implement such things as electronic funds management and automated teller machines. We have not seen such innovations in computing education, but they may not be far away. John Stasko's paper at the 1997 SIGCSE Technical Symposium [3] reports on some results in further development of his animation system that made it accessible to students. Thus, students can produce their own algorithm animations, giving them a much deeper understanding than they would have if they passively observed an existing animation. Similarly, the work that is being done on web-based education could well develop to the point that self-paced learning via the web would become the norm rather than the exception, with fairly revolutionary consequences for education.

Many advances in computing have been accompanied by paradigm shifts that were essential to effective use (and teaching of) the advances. In the area of programming languages alone, paradigm shifts have been an essential component of effective instruction in Ada, C++, and Java. So it is likely that significant benefits from the use of technology in education will require paradigm shifts, for example, such as the shift required due to the introduction of self-paced education via the web. But we should not worry that a paradigm shift may be needed for significant benefits. After all ... shift happens!

So is technology in computing education yet another bandwagon? Perhaps. But that only means that we need to be careful that our use of technology in education does more good than harm. Thus we need to carefully and objectively evaluate our applications of technology in education and avoid using technology just because it is there or because we are pressured to do so. With continued effort and careful assessment we can look forward to significant improvements in the education of our students that will result from good uses of technology.

9 About the author

Joe Turner is a Professor of Computer Science at Clemson University (USA). He currently serves as Treasurer of ACM, as a Director of FOCUS (Federation on Computing in the United States), as US representative to IFIP TC-3, as Vice-chair of IFIP WG3.2, and as a member of the Board of Directors for the Association of Specialized and Professional Accreditors. His previous positions include service as President of the Computing Sciences Accreditation Board, Chairman of the ACM Education Board, a Director of the Computing Research Association, and a Director of the National Educational Computing Association.

References

1 Goldberg, M. W. CALOS: First results from an experiment in computer-aided learning for operating systems. *SIGCSE Bulletin 29*, 1 (March 1997), 48–52.

2 Kay, D. Bandwagons considered harmful. *SIGCSE Bulletin 28*, 4 (December 1996).

3 Stasko, J. T. Using student-built algorithm animations as learning aids. *SIGCSE Bulletin 29*, 1 (March 1997), 25–29.