SOCIAL RESPONSIBILITY AND COMPUTER EDUCATION

Norman R. Nielsen
Stanford University

The issue of responsibility on the part of the computer professional is one that has blossomed very recently. Members of the computing field have become very much aware of some of the issues, even if there isn't any consensus on how to handle them. The ACM debates on standards of professional conduct and on the desirability of a professional society such as the ACM taking formal positions on social issues are illustrative. So are the activities of the computer professionals against the ABM. Society itself is also becoming aware of these issues. Only last year the State of California began actively considering the possibility of licensing computer professionals. The fact that a whole session of this Symposium is being devoted to social aspects of computing is yet another indication of the growing interest and concern in this area.

Although there appears to be agreement about the need for concern in this area, there is no consensus on what could (or should) be done by the faculty member who is teaching computer related or computer based courses. The situation has the characteristics of a dilemma. On the one hand, the faculty member is in the position of teaching facts to his pupils. By and large, positions on social issues, while based upon facts, are heavily influenced by personal philosophies, beliefs, and emotions. Two rational individuals can, and frequently do, come to opposing views on the same issue. Accordingly, it is inappropriate for the faculty member to mix in his personal viewpoints as part of the regular material he dispenses as part of the course. On the other hand, the faculty member, based upon his knowledge and experience, does have an obligation to speak out and make his views known.

Although it would be nice to resolve the foregoing dilemma at this time, that is something that must be left for the future. However, even though this particular issue can be avoided, the whole range of social issues can not. Consequently, it is the position of this paper that the faculty member should try to prepare the student to make better decisions for himself rather than try to persuade him to take particular stands on particular issues.

In the final analysis the student, as he makes his way in the world, is going to make his own decisions on social questions; these decisions should not, and indeed can not, be made for him. Thus, if the teacher can impart to his students an awareness of the consequences of using the tools and concepts which he is covering in the classroom, the student will be in a better position to make his own judgments. In this sense, then, it is the faculty member's task to make the student's education a more meaningful one. For if that education has given him an awareness of the implications of computing, the student will then be in a position to make more enlightened decisions on those computing issues that impact society.

Thus, the real question to be addressed concerns the means by which one can seek to gain an acceptance of responsibility on the part of the student, the means by which an awareness of the broader consequences of computing tools can be imparted. Because of the newness of the concern in this area, there is little to go on. In particular there is no "definitive course of action" which can be laid down and advocated to faculty members. However, this should not be an excuse for inaction.

At Stanford we have been trying to introduce some of these broader considerations through minor modifications to the curriculum. This paper sets forth what has been done in two courses given by the Computers and Information Systems group in the Graduate School of Business. It is hoped that these illustrations will serve not only to show what can be done but also to stimulate some thought and spark some subsequent action on the part of the reader.

Professional Responsibility

There are two general types of responsibility between which it might be desirable to distinguish. The first is what might be termed professional responsibility, while the second might best be termed societal responsibility. Essentially, professional responsibility is concerned with the attitude or approach taken by an individual when he is solving problems. The concern is with whether the resulting product will be "up to standard". Admitedly, there is not always agreement about the professional standards that should be applied. However, the concern in this area has a longer history, and hence there is somewhat more agreement on what is included. Every faculty member in the area of computing has undoubtedly wrestled at one time or another with the problem of getting the student to internalize the idea that a good programming job must include good documentation. In other words, a professional programming job will by definition include proper and adequate documentation.

The technique that has been applied at Stanford is essentially one of causing the student to look at the implications of the uses of the tools being discussed. In other words, what happens if a non-professional approach is taken? First, as is usually done, the student is given some clear indications of the problems that can result. However, as follow-up, the student is shown a problem situation in which certain professional standards had been sacrificed for one or another "good" or "powerful" reasons. In this way, the point is brought home as to the real world pressures for sacrificing standards that all too often must be faced. Finally, the student is placed in a decision-making or other active role where he must work with the results that stemmed from such an improperly performed project. This exercise serves to indicate clearly both the difficulties engendered by such a non-professional approach and the consequences which must be suffered as a direct result. Such a graphic illustration helps bring about an understanding of why the professional standards were developed in the first place. With such an understanding of both the rationale and the consequences, the student is much more likely to take a proper approach, rather than to "skip the documentation because that's a drag".

An illustration of how this approach can be applied may be seen in the course entitled "System Simulation". This is a one quarter, four-unit course, focusing upon the fundamentals of simulation. It covers preliminary system analysis, language selection, model construction, data collection, programming, debugging, validation, experimental design, and analysis of results (including system modification and redesign). As such this course is like many other standard tool courses with the student being taught "how to do it". However, the above type of approach is also used, so that the student gains some appreciation for the applicational implications of the tools which he is learning to use. Consider a few examples of this process.

One of the many teaching tools that is used is the so-called case method. The case is nothing more than a "word problem" which has been expanded to several pages with background data to provide a realistic "setting". In addition a fair amount of additional data (some of which is extraneous) has been thrown in. This gives the student some practice in problem formulation as well as analysis. Such cases have long been used in the simulation course to illustrate applications of simulation techniques. However, such cases can also be used to focus attention upon the implications of the manner in which the simulation tools are applied.

One case concerns the simulation of an open-pit mining operation. A simulation model had been constructed by the company to analyze mining operations. However, a number of flaws are evidenced in the approach. For example, it would appear that the modeler had been "a guy with a technique in search of a problem". Thus, some of the more relevant aspects of the mine's operation had been overlooked. It would also appear that a number of perhaps unjustified assumptions had been introduced solely to make the programming task more tractable. These are symptoms of some of the non-professional approaches that are all too often found in practice. Of course in the case, as in the real world, there are "reasons" why things were done the way in which they were.

The student, however, is faced with the problem of coming in after the fact and having to deal with this simulation. He must make judgments concerning the disposition of the model. Can any decisions be made from it? If so, which ones? Should the model be scrapped, modified, or replaced with a new effort? There are other questions of a less technical nature. How did the misfortune occur, and how can the reoccurence of a similar development be prevented? What was the impact of the poor simulation model upon the personnel at the mine in terms of future willingness to attempt simulation analyses, to make decisions based upon simulated results?

By having to deal with the aftermath of an improperly performed simulation study, the student develops a much greater feeling for and appreciation of the impact and consequences that can stem from the more technical decisions which he might make concerning a simulation study. He comes to realize that some of the decisions that at first glance appear to be local in nature do indeed have a much broader impact. The experience of looking at the situation from the other side of the coin, from the viewpoint of the user, as opposed to the developer, can be very educational.

Another case that is used concerns the problem of ambulance dispatching in a large city. Here too there are problems in the design of the model stemming from political pressures for a "quick" result and from a poor experimental design. However, this case also provides an opportunity to introduce the idea of extending one's analysis to include some broader social concerns. In the dispatching problem, one is faced with the unstated trade-off between minimizing the cost of the ambulance service and minimizing the ambulance service's response time to calls for assistance (thereby reducing human suffering and loss of life.) Since the two goals are at cross-purposes, the problem solution must include an implicit equating of response time and cost. In other words, a compromise must be found between the good of the ambulance service and the good of the society. In this way the student sees how such broader and sometimes "fuzzy" social issues can be handled in the analysis. Thus, he sees that the use of technical tools need not preclude the treatment of social considerations.

Still another approach is offered by a third case which relates to the scheduling of barge tows on a river. A simulation model has already been programmed for the student and is running in a time-sharing mode. The student then uses this program to analyze the operations of the barge company and to make recommendations on scheduling policies. The catch here is that there is a great deal of variability in the simulation results from run to run. Hence, the student who takes the problem of run design lightly is quite likely to make some bad decisions. Again, as with the other cases, the goal is to bring home to the student the consequences stemming from decisions to take a shortcut. By placing the problem in a realistic setting, the student not only sees that wrong answers were produced for a particular case but more importantly, he sees what the costs of those wrong answers were. This leads to a better appreciation for the real consequences that can stem from what appear to be more technical decisions.

Although the case type of problem offers many advantages over the straight "problem statement" type of formulation, it still falls short of realism. For this reason, another technique is also used in this course. This involves the use of an actual project drawn from the outside world. This permits the student to become acquainted with the pressures and difficulties that are associated with such problems. Hearing about these pressures from someone does not have nearly the impact that actually experiencing them has upon the student. The project is essentially a start-to-finish problem solving exercise. The general problem area is picked out by student teams. Each team must trim their problem down to size and come up with a specific statement of the problem to be solved. Following this are the usual steps of model formulation, programming, etc. However, mixed in with this are now some of the real pressures that must be faced in actual projects. These include time limitations, personnel constraints, computer time budgets, computer memory limitations, etc. There is also the overriding pressure to get results, for part of the project is to report findings back to the real world managers or staff responsible for the area studied. In other words, the student has heard about the problems that he will face, and he has seen the consequences that can result if they are not handled appropriately. Then he is given the opportunity to make the trade-offs and face the consequences in his own project.

The above approaches serve to illustrate how responsibility considerations can be introduced into the curriculum. However, it should be realized that none of the techniques described were brought into the simulation course solely to emphasize professional responsibility, for the various cases and the project have other functions as well. Such approaches would have been used in the normal course of teaching the desired material on simulation. What is to be emphasized is that for very little additional effort, the points to be made about professional responsibility can be made with a stronger and more lasting impact. There need not be any large financial or personnel resource commitment to this task. Clearly, better materials can be developed, and perhaps this paper will stimulate that type of activity. But even without such development activity, there is a lot that can be done to emphasize professional responsibilities, and this can often be done within the framework of the existing curriculum. It is particularly in this latter area of curriculum change that this paper is seeking to stimulate activity.

Societal Responsibility

The second type of responsibility to be considered is what was termed above as societal responsibility. This reflects the concern for the broader impact of computing upon those not immediately connected with computing, upon the general society. The same type of approach can be taken as for professional responsibility - trying to bring the student to see the implications and consequences of the applications of the tools which he is learning to use. Concern for some of the broader implications of computing is much more recent, and there is correspondingly much less agreement on just what should be attempted. Hence, the following illustration is only that - illustrative and not prescriptive. However, it will provide an indication of the kinds of things that can be done to bring social considerations into a course.

The course in question is entitled "Management and the Computer", and is a one quarter, three unit course offered to all of the masters candidates. This type of course can be viewed as the typical introductory computing course for the non-scientist. (In this particular instance, the students are largely prospective managers who will be responsible for the way in which computing techniques will be applied in the world of business and government in the years ahead. Hence, the inclusion of social considerations is of particular relevance. However, the same ideas are also relevant for students in the humanities, arts, and sciences). The course covers an introduction to computers and how they operate, concepts of hardware, software, etc. It then goes on to teach programming skills. The last portion of the course deals with the application of computing techniques to problems.

From this description one can see that there is nothing unusual about the course. It is in many ways like other introductory courses. However, the emphasis in this course has been shifted slightly to include considerations of the impact of computing decisions upon both organizations and society.

By way of illustration, a case about a life insurance company and a new application for its marketing department becomes a vehicle for discussing the whole issue of where decisions about the implementation of new applications should be made. The student is led to consider the advantages of having the more technically oriented personnel at the computer facility make such decisions. Yet this has such obvious disadvantages that he is led to consider letting the non-technical people in the using departments make such decisions. Yet this too can have significant and undesirable impacts upon the operation of the firm and of the computer department. In this way the student is drawn into a full examination and consideration of the various factors involved in deciding upon the applications to be run, on the manner in which they will be implemented, etc. In this way the student gains an appreciation for the consequences or implications of the various ways in which decisions can be made with regard to the use of the computer in a firm.

Although cases are helpful tools for stimulating thought, they are by no means a necessity. Papers or articles can also be used for this purpose. For example, an article on vote counting is sufficient to bring out a number of broader issues that should be raised in connection with that operation. The advantages of computerized processing in terms of speed and accuracy serve as the incentive for developing this type of computer application. Yet, some of the difficulties encountered provide for bringing in the non-computer considerations that can play an even more important role (e.g., misalignment of names on the voting booth template, ill-defined procedures for collecting and handling ballots). This can lead into a discussion of how you might determine what the computer is really doing. The possibilities for fraud in the counting and the difficulty of discovering such activities serve to make the whole question of security take on much more meaning. The number of reported counting foul-ups (all apparently without fraudulent overtones) highlights the difficulty of developing even simple systems.

Through discussions such as these, the student very quickly comes to realize that computers are not automatically an answer to problems - even where speed, accuracy, and repetitiveness mark the problem. As a result the student is more likely in the future to consider what some of the implications and ramifications might be before an automatic "let the computer do it" is sounded.

When the course comes to data bases there is another opportunity to bring in consideration of social consequences. Looking at things from the point of view of a firm or of a government or even of a specific application, the student can readily see the advantages of developing such files and of using computer processing power to maintain and interrogate those data bases. As his excitement mounts, the student sees ways to combine data bases for even greater benefit, etc.

However, at this juncture one can introduce a consideration of the technical problems of data base security, back-up, error checking and correction of new entries, etc. This sets the stage for an excursion into some of the non-technical problem areas and into some of the implications for society of such data bases. One can look at the problems of preventing misuse of a data base or of preventing unauthorized applications from becoming authorized. A look at how improper inputs or system quirks can haunt an innocent person for years, can make a discussion of operational procedures, indemnification, and responsibility much more meaningful.

It is only when the student can view both the positive and negative features of a particular type of computer processing and when he can see the implications of both the advantages and the disadvantages, that he is in a position to make more informed decisions about the course which computing should take. If the student has been taught to think and to make good decisions for himself. The felt need of faculty members for "selling" particular positions in the classroom will diminish, as the student will be able to arrive at the "right" decisions (whatever they might be) by himself. And, after all, that is what education is all about - to teach the student to think.

Conclusion

This paper has tried to illustrate, through the example of two particular courses, the kinds of things that can be done in everyday classes to introduce some concern for the potential impact of computing tools. Admittedly, the above approaches are not a model to be copied directly by all or even by many, for every faculty member must tailor his course to fit his students and their needs. However, the courses described are indicative of the kinds of things that can be done using ordinary curricula and materials. Introducing the computer to management students probably offers a more fertile ground for these kinds of changes, since many of these students will soon be in positions of decision making or policy making authority. However, the simulation course does illustrate an application in a more tool oriented environment.

The important point to realize is that the tools can not be taught in isolation. It is also necessary to teach the applications, uses, and misuses of those tools. Once the student has been made aware of the broader implications of the uses of computing tools, he will be in a much better position to make enlightened value judgments on issues relating to the application of computing techniques in our society. Since these types of considerations can gradually be worked into courses without the need for major curriculum revision or for the development of large quantities of new materials, faculty members should be able to begin introducing some of the broader social considerations into their courses without delay.