The Emergence of Student Models from an Analysis of Ethical Decision Making in a Scenario-Based Learning Environment

Mike Winter[1] and Gord McCalla[1*]

[1] ARIES Laboratory, Department of Computer Science, University of Saskatchewan, SK, Canada

Abstract. Too often, professional ethics issues are trivialized in software engineering education. To begin to remedy this situation, we have built two interactive, adaptive learning *scenarios* that place students in the role of a software project manager confronting many critical project decisions, each with an ethical dimension. As students move through a scenario, making and justifying their decisions, their behaviour can be monitored and used both to adapt the scenario to each student as they proceed, and in post hoc analysis to identify different classes of ethical behaviour. In this paper we discuss five different classes of student behaviour that emerged from the analysis of protocols collected during the use of these scenarios in a third year undergraduate software engineering class. We speculate that the existence of these general *student models* can be used in several ways to further enhance the learning of ethics

1 Introduction

Professional ethics issues are often seen to be an irrelevant afterthought to students learning to become software professionals. Yet, the importance of preparing students to be ethical computer professionals in their practice, and to be aware of the social impact of their work, is widely acknowledged (eg. Appel, 1998; Gotterbarn and Riser, 1997). One of the main purposes of our research is to make ethics seem less 'academic' to computer science students. Key to this goal is the realization that making ethical decisions in the software development process isn't an afterthought to the process; it's an integral part of it. It is important for students to realize that connection and to understand they are not just technicians without moral responsibilities.

We have designed two goal-based scenarios (following the idea of Schank and the Community Partnering Development Team, *http://www.ils.nwu.edu/projects/cper/overview.html*) to be used by students to practise making ethical decisions in a realistic context. In each scenario, a student takes on the role of a manager of a large software project, and is then confronted with a series of difficult situations, each requiring him/her to make a decision with an ethical dimension. These

* We would like to express our gratitude to John Cooke and Judy Thomson who helped with the experiment, and acknowledge the Natural Sciences and Engineering Research Council of Canada for funding this research. We would also like to thank the Basser Department of Computer Science at the University of Sydney, Australia, who acted as sabbatical hosts for Gord McCalla during the time this paper was written.

scenarios were used in a third year software engineering course at the U. of Saskatchewan. From an analysis of the data collected during this use, several classes of student behaviour were identified. In this paper we describe these *student models* and speculate as to their potential uses.

2 The Ethics Scenarios

Two goal-based scenarios (accessible via the Web) have been developed. As students individually interact with a scenario, they move through different situations, each confronting them with a decision that must be made about the software development process. The ethical issues in each situation are confounded and contextualized by other issues such as looming deadlines, people management concerns, unexpected opportunities to further the project, etc. In each situation, students are able to consult the relevant section from the ACM Software Development Code of Ethics (http://www.computer.org/tab/seprof/code.htm), thus helping them to reify abstract Code concepts. The students' choices at each decision point determines the path they take through the scenario. Different choices lead to different consequences, in the short or long term, leading each individual to have a different experience in using the system. The goal is to give students a realistic feeling for the true complexities they will face in the real world of software development, if they are to maintain an ethical stance. As they make their way through a scenario, students are asked to justify their decision in each situation with a paragraph explaining why they did what they did. It forces them to reflect on the ambiguities inherent in each situation and to relate ethical issues to concrete reality, thus consolidating and deepening their understanding of the issues. At the end of the overall scenario there is a final outcome provided to the students based cumulatively on their previous decisions and a recap of all their decisions is provided for them to browse. Students are also given a chance to comment on the entire scenario, and add commentary on each of the choices they made earlier, should they wish.

In scenario one, students take on the role of being the new head of a small team of programmers and software engineers at Denodyne Systems, and are charged with guiding the team as it overhauls the company's MedXFile medical database system. From this starting premise, each student has to deal with a variety of crises as the development team struggles to release a finished product by the release date. Some of the crises include electronic harassment, a major yet rare bug, an angry user group, and the need to look through the project team's e-mail to try to stem an information leak. Scenario two (not shown) is structured similarly, but explores the issues that arise in building an embedded system application.

A truncated version of a typical student's interaction with scenario one is shown in Figure 1. The system's output is in plain text, with a bold label indicating each new situation; the student's input is italicized; and "….." is used to indicate that text has been left out of a situation in this paper for the sake of brevity. Only the initial setup, abbreviated versions of the first and second situations, and the start of the third situation are shown, to give a flavour for the style of interaction.

SETUP
You are the new head of a small group of programmers and software engineers at Denodyne Systems in Calgary, which is in the market of producing medical database products. ….. At your first management meeting, you are told the situation at Denodyne, and receive the following talk from Mr. D, the general manager of the company: "You know we're counting on you ….."

ILLICIT_CODE
….. After meeting with the entire team, you are approached by one of the team members. He says, " ….. I hang out with a few guys from Phrack, and we trade software and games and stuff, and one of them gave me a disk from work with some stuff on it. There was some source code and new structures on there from a version of their program that he had taken home to work on. This code could really help us in development ….. We could change it around a bit so that no one would ever know where it came from. It's done all the time. They would do the same to us if they could. You can't copyright an algorithm, so it's not illegal, and it's no different from using a system library or some code off of the net."

This could really give you a leg up on development; do you allow the code to be used in your project?

Choice: *NO*
Justification: *This is obviously unethical and I don't think it is worth it to risk the company's reputation. If the boss doesn't like it, I'm sure I can find a job somewhere else.*

READ_EMAIL2
You decide not to use the Phrack source code. You realize that while it may not be illegal, it probably isn't ethical to use someone else's intellectual property. You're also worried that using outside source code might cause more problems than it's worth to try and integrate it into your software. ….. A week later, Mr. D calls you into his office and presents you with an advertisement in the business pages of the local newspaper. He says "Look at this; somehow Phrack found out how we're coding the new features we're planning to implement. ….. Do what you have to do to find the person [responsible for the leak] and get rid of him or her."

Talking to your network manager later, he tells you that the company routinely logs all e-mail and internet use statistics for trouble-shooting purposes. It would be pretty likely that you could find out who was leaking the information if you went through the e-mail. …..

Do you look through the e-mail or not?

Choice: *NO*
Justification: *This is also unethical and workers would turn to hate me for invading their privacy. I tend to agree with them and would not like it if someone did this to me.*

SPYING4
You decide not to read through the e-mail ….. if you read through e-mail that's generally considered private, morale would go down even lower than it is. However, when Mr. D finds out about this, he is furious with you. …..

Figure 1. A Truncated Version of a Typical Interaction with Scenario One

Scenarios are structured in an underlying discrimination graph. Discrimination graphs seem to be both simple and well suited for rapidly prototyping new scenarios, but ultimately a structure such as a Bayes Net (as used by Mislevy and Gitomer, 1996, for example), might allow for more sophisticated analysis. The discrimination graph for scenario one is shown in Figure 2. Situations are represented by nodes. Each situation contains a textual description of an ethical dilemma. The situation normally starts with a synopsis of the reasons that may have motivated the student to decide as they did in the previous situation. The text then moves on to outline the new situation, at the end of which the student is asked to choose among (possibly several) alternatives, thereby opening up different paths in the graph. Branches in Figure 2 represent the alternative paths. To control unconstrained growth in the graph, a process of 'folding in' of some of the paths happens. Thus, different students may temporarily take a different path, but come back together again at some later point.

The path that a student follows is not always determined by their choice at the immediate decision point. Thus, in the last two situations of scenario one, a linear weight function that accesses all previous decisions (kept in a logfile for each user) is applied to choose a follow-up situation. It can reward unethical behavior in some ways. For example, if a student has chosen not to fix a major bug (the BUGFIX situation), they get a major boost in getting the product out on time. The weight function punishes unethical behaviour in other situations. For example, if a student has spied on the workers (the SPYING situation), the negative morale that results is counted against them. The weights and the threshold value are then cumulated to come up with a final decision about which follow up situation is appropriate. It would be easy, of course, to modify the weight function in order to explore the effects on students of different reward systems, thus allowing the creation of essentially "malevolent" or "benevolent" worlds that differently reward ethical vs. unethical behaviour.

3 The Experiment

We conducted an experiment in the use of the scenarios in a third year software engineering class (Cmpt. 370) at the University of Saskatchewan. In the experiment the 95 students in the class were asked to go through both scenarios, working on their own. The students' decisions, rationales, and end-comments were all recorded, analyzed, and later used to stimulate a classroom discussion. The discussion was extremely vigorous. It was focussed by the specific issues that emerged from the students' shared experience in using the scenarios, and it was animated by the different decisions they had made at various important junctures in the scenarios.

There were two different phases of analysis done. The first, used to facilitate in-class discussion, recorded how the students collectively responded to each ethical situation. The decisions made by the students in each of these situations were counted, and the justifications for each choice summarized. The numbers next to each decision point in Figure 2 indicate the percentage of students who took that path in the experiment. This material was used to encourage in-class discussion, for example around similarities and differences in student decisions and justifications in the two situations involving privacy (READ_EMAIL and SPYING). The second phase of analysis looked at each student's behaviour over a whole scenario, to see if standard patterns of behaviour, *student models*, occurred.

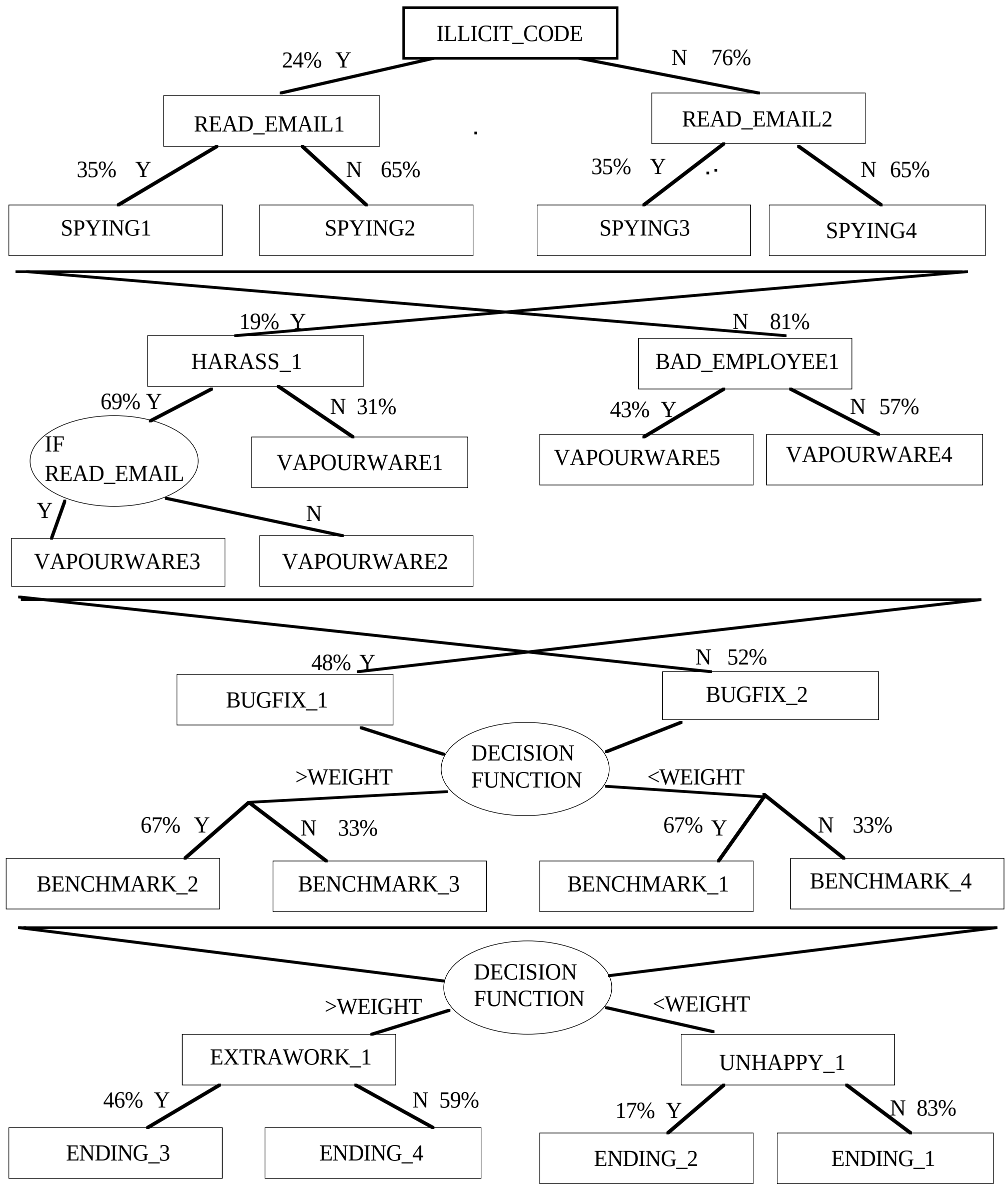

Figure 2. Discrimination Graph for Scenario One

4 The Student Models

The second phase of analysis categorized students on the basis of the paths they had taken through a whole scenario. Clear patterns only emerged from scenario one, so this became the focus of the analysis. Student decisions in HARASS_1 and the alternative BAD_EMPLOYEE1 were ignored; as were their decisions at the EXTRA_WORK and UNHAPPY alternatives. In each case, these situations partitioned the students into two distinct groups confronting two distinct situations, thus making it illogical to compare their decision making. Therefore, only the 6 situations that all students shared "in common" (ILLICIT_CODE, READ_EMAIL, VAPOURWARE, SPYING, BUGFIX, and BENCHMARK) were used. If any two students responded identically, they were put in the same group. Each group was further augmented by comparing the justifications of students in the group to the justifications of students who had taken similar paths (with one decision made differently). If the justifications were similar, the students with small differences in their paths were added to the main groups.

Overall, 91 of the 95 students in the class tried scenario one. Of these, 82 finished the scenario all the way through, 19 of them multiple times. For the students who tried the scenario more than once, we chose for analysis the chronologically earliest run through in which the students had fully justified their decisions, feeling this was most representative of their real feelings. We were able to categorize 67 of the students who completed the scenario into 5 different student models; the remaining 15 did not fit into any equivalence class. These students may, in fact, be part of other as yet undefined student models. However, we either did not find a consistency of behaviour among their decision making, or the numbers of students in that unknown model were not sufficient to form a critical mass. The descriptions of these student models follows.

Student model A: *straight and narrow*
Number of users: 21
Common paths taken: No/No/No/No/Yes/No; No/No/No/Yes/Yes/No

People in this model tended to make the 'safe' decisions, never taking any risks, or making any decisions that could be perceived as unethical. Figure 1 is an example of a straight and narrow student. Their behavior may be construed as being very concerned with ethical issues that involve the end users and workers in the software firm, but deficient in terms of the user's responsibility to the firm he or she is employed with. Although it should be noted that the students in this model often took the stance that they would do a certain action to facilitate good software development, they were reluctant to break any perceived behavioural taboos. They also thought that the bond of trust between themselves and the software developers was one of the most important facets of the software development process.

Student model B: mission critical dominant
Number of users: 13
Common paths taken: No/Yes/No/No/Yes/No; Yes/Yes/No/No/Yes/No

The most interesting characteristic of students in student model B is their understanding of the special nature of the 'mission-critical' elements of the software being developed. The most obvious place where this concern manifested itself was the VAPOURWARE situation where students were encouraged to claim the software would be done by its release date, even if this meant releasing it with bugs. Students in model B pointed out that rushing the development of the software to try to meet an impossible deadline was sure to lead to more problems in the long run, including potentially serious risks for end users of the system. An interesting situation where people in this group acted differently from others in similar student models was the READ_EMAIL situation. Everyone in this model was of the opinion that if an employee is using email on company time then the company has a right to look at it, but they then thought that increased surveillance of the employees (SPYING) was counterproductive and unethical.

Student model C: opportunistic
Number of users: 9
Common paths taken: Yes/No/No/Yes/Yes/No; Yes/No/No/Yes/Yes/Yes

The behavior of the main group of students in this model was similar to that of the second biggest sub-group in student model A. The major difference was in their handling of the ILLICIT_CODE situation. The students in this model were of an opportunistic mindset when confronting whether to use the source code from the rival company. The thought seemed to be that if the company was careless enough to hand over their important source code, the company was absolved of responsibility for using it. The urgency of the situation was also a deciding factor for many of the students. They were strongly opposed to any sort of violation of personal privacy of the workers (READ_EMAIL, SPYING). The other situation where they differed from those in student model A was in the VAPOURWARE situation. Where the students in model A generally claimed the release date would be met in order to show confidence in their workers, the students in student model C thought that a bit of constructive dishonesty was necessary to keep enough customers in place so that the company would stay afloat. The students in this group were more cutthroat than those in some of the other models. A small subgroup in this model also thought it necessary (and of minor ethical concern) to edit the testing logs to give the company an added competitive advantage (BENCHMARK). The majority of the students in this model tended to think that this action was very unethical.

Student model D: public relations dominant
Number of users: 14
Common paths taken: No/No/Yes/Yes/Yes/No; No/No/Yes/No/Yes/No

The students in this model are an interesting combination of those in student model C and student model E. Students in model D were reluctant to read workers' e-mail (in the READ_EMAIL situation) for reasons of privacy, but thought that increased surveillance (in the SPYING situation) was so essential to the completion of the project that they had to do it. Students in model D were not as driven by ethical concerns as in some of the other models, but were more concerned about the public relations consequences of their actions. It should be noted, however, that even though they assured their users of a successful early completion of the project, they all thought it was necessary to fix the major bug in the program (BUGFIX), and that it wasn't right to alter the test logs to give themselves a competitive advantage (BENCHMARK).

Student model E: business dominant
Number of users: 10
Common path taken: No/Yes/No/Yes/Yes/No

This student model originally seemed to be a subgroup of student model B, but the students' responses were of quite a different calibre from those of model B. The main difference in the two groups was the willingness of students in model E to release the software with a few bugs while reassuring the user group that the product would be out on time (VAPOURWARE). This didn't show the concern about the 'mission-critical' nature of the project that the students in model B did. It should be noted that most people in this group then chose to fix the major bug that was discovered (BUGFIX), but a significant subgroup did not. This group was also very business oriented. In their justifications, there were not that many concerns about the ethics of actions, but mainly about the business consequences of each move.

5 Use of the Student Models

The existence of such student models is interesting in its own right, but could also be useful in several ways. One of the main uses of these student models would be to enhance the system's decision making using stereotypes based on the student models, in a longstanding user modelling tradition going back at least to Rich (1979). For example, after the SPYING situation is presented, a user who has answered "No/No/No" would be unambiguously identified as a straight and narrow stereotype (model A). This person has the strength of being highly ethical (or at least behaving in such a way), but perhaps unrealistically so. Such a person might be confronted with further situations where a prior ethical decision has, itself, led to negative consequences. For example, a "type A" person refusing to read e-mail to catch a leak of project information to a competitor (READ_EMAIL), might be presented with a follow up situation where Denodyne loses the contract to that competitor, and all project members lose their jobs. Such challenges to a student's vested beliefs might lead them to a more subtle understanding of the issues, and the ability to identify and respond appropriately to ethical concerns they later meet in the real world.

Another use of the student models is for a class instructor to use them to categorize the students in his/her class. This can be done merely by observing common paths taken when using the scenarios (confirmed by a quick look at the qualitative information provided by the students). The most commonly occurring student models can then be the focus of the actual discussion in class, and the unethical or unrealistic aspects of each type of student explicitly discussed. Further, comparisons among the student models can be prepared in advance by the instructor to stimulate discussion. For example, it would be possible to point out to the students, should there be enough of them in each model, the way that the public relations dominant thinkers in model D are similar to and different from the opportunistic students in model C and the business dominant students in model E. All three classes of students thus see natural variations of their own behaviour patterns and gain a broader perspective.

Finally, the student models make it possible for students to take particular roles when interacting with a scenario. For example, students diagnosed as straight and narrow thinkers (model A) in their first pass through scenario one could take an opportunistic stance (model B) in a subsequent run through scenario one or in scenario two. This would broaden their experience and prepare them for a more complex real world than they imagine exists. It is possible in assigning roles to create artificial student models, for example the "spawn of Satan" who always behaves totally unethically. This would allow students who are already leaning towards unethical behaviour to be better acquainted with the ultimate consequences of being unethical. In fact, we postulate that many of the 19 students who used the scenario more than once in the experiment, may have assigned themselves different roles each time through.

6 Comparison to the Literature

The use of paper-based scenarios is a long-standing one in ethics teaching, and their use primarily in the classroom has proven to be effective (Stevens et al, 1993) (Ascher, 1986). In fact, the inspiration for some of the situations in the scenarios here was taken from Kallman and Grillio (1996), and various online sources. Perhaps the most similar to our work is the ethical board game developed by Lockheed Martin (http://www.lmco.com/exeth/dilbert.html) in which groups of employees form teams and then are presented with ethical scenarios and a range of responses. The teams then discuss the effects of all possible decisions and debate their merits.

Our research extends the use of scenarios by computationalizing and individualizing them for one-on-one student-system interaction. Technically oriented students are given a familiar technological environment in which they personally must make decisions that they recognize as important, but which have non-obvious consequences. This provides a realism and a personal relevance that is hard to achieve in a pure classroom approach or even in the group-based gaming approach of Lockheed Martin. Our approach will work best, however, with a combination of individual use and classroom follow-up, as we showed in the experiment.

Our analysis, inducing student models from observed student behaviour, is also non-standard. In other ethics experiments, subjects are given on paper several small scenarios with action plans, and are then asked to rank their approval of the action on a Likert scale of 'strongly disagree' to 'strongly agree', with five intermediate steps (Abratt et al, 1992)(Stevens et al, 1993). A more complex example is given in Reidenbach and Robin (1988) in which students rate various scenario/action pairs on a thirty point sliding scale with various ethical considerations divided into categories (i.e. deontological, utilitarian, etc.). Regression analysis is then performed on the responses, and the scores are compared across the groups in various categories. The most comprehensive attempt to measure moral development is in Colby and Kohlberg (1987) who describe a fifteen year long longitudinal study to categorize people into various moral 'stages'. This is done by doing an intensive qualitative analysis of the rationales subjects give for making various choices in response to ethical scenarios. This approach, while attractive for its depth of analysis, is impractical for measuring changes in ethical stance brought on by students using one or two goal-based scenarios over a short period of time.

Most of these methods use previously existing groups, and then rate their responses, but for our methods, we wanted to discover groups with similar ethical value systems based on observed actions in the scenarios. Our approach has allowed us to compare actual paths through the computational scenarios. This seems to be a more useful way of categorizing groups of people who think similarly, in particular because we can then reason about subtle differences in behaviour based on the actual decisions taken relative to the specific situations involving computer software. We also then set up the useful possibility of the system itself automatically classifying future students as they use the scenarios, not after the fact.

7 Conclusion

In this research we have shown that it is possible to design goal-based scenarios that allow students to actually experience the complexities of making ethical decisions in realistic situations. Frequently, unanticipated consequences result from decisions taken, based not just on one action but possibly the cumulative effect of many actions. Judging by the positive response of students who used the scenarios in an experiment carried out in a software engineering class and the enthusiastic follow-up discussion in class, the scenarios were a success in stimulating students to think more deeply about ethical issues and to realize their importance. As this paper has shown, the experiment also has allowed the delineation of several different classes of student behaviour, what we have called "student models", that can be used in a number of ways to further enhance the scenarios and their use.

The scenarios have been used again in a fourth year ethics class at U. of Saskatchewan in January, 1999, with similar enthusiasm on the part of the students. Preliminary analysis shows similar student models emerging, as well, although with fewer students taking a straight and narrow approach (model A) than in the first experiment. Further analysis will be carried out to confirm this. This analysis will also look at the data from both experiments in order to see if other student models emerge, in either scenario.

Overall, we believe this research shows that providing students with a realistic means of exploring abstract issues in a concrete setting allows the issues to be perceived by the students as both relevant and appropriately subtle and hard, not something trivial. It allows "soft" issues to be seen as integrated with technical concerns, and raises topics like ethics to "first class" status for technically oriented students.

References

Abratt, R. N., Higgs, D. and Nicola, S. (1992). An examination of the ethical beliefs of managers using selected scenarios in a cross-cultural environment. *Journal of Business Ethics*, 11: 29-35.

Appel, F. (1998). Including the social and ethical implications of computing in the computer science curriculum *ACM Computers and Society*, 28: 56-57.

Ascher, M. (1986). Ethical conflicts in the computing field - an undergraduate course. *ACM Computers and Society*, 16: 19-22.

Colby, A. and Kohlberg, L. (1987). *The Measurement of Moral Judgement: Volume 1, Theoretical Foundations and Research Validation*. Cambridge: Cambridge University Press.

Gotterbarn, D. and Riser, R. (1997). Ethical activities in computer science course: goals and issues. *ACM Computers and Society*. 27: 10-15.

Kallman, E. A. and Grillio, J. P. (1996). *Ethical Decision Making and Information Technology: An Introduction with Cases*. New York: Mc-Graw Hill.

Mislevy, R.J., and Gitomer, D.H. (1996). The role of probability-based inference in an intelligent tutoring system. *User-Modeling and User-Adapted Interaction*. 5: 253-282.

Reidenbach, E. R. and Robin, D. P. (1988). Some initial steps toward improving the measurement of ethical evaluations of marketing activities. *Journal of Business Ethics*. **7:** 871-879.

Rich, E. (1979). User modeling via stereotypes. *Cognitive Science*. 3: 329-354.

Stevens, R.E., Harris, J. O. and Williamson, S. (1993). A comparison of ethical evaluations of business school faculty and students: a pilot study. *Journal of Business Ethics*. 12: 611-619.