Integrating Computer Ethics into the Computing Curriculum: A Framework for Implementation

A Barnard, C de Ridder, and L Pretorius
University of South Africa, Pretoria, South Africa

E. Cohen
Leon Kozminski Academy of Entrepreneurship and Management, Warsaw, Poland

barnaa@unisa.ac.za driddc@unisa.ac.za pretol@unisa.ac.za

Eli_Cohen@acm.org

Abstract

The advent of the Information Age and global connectivity has placed ethics center stage in the use of Information and Communication Technologies (ICT). As the drive towards the establishment of a so-called IT profession gains momentum, ethical conduct and codes of ethics have recently been formulated and introduced formally. Initiatives in this regard can be attributed to, among others, the ACM and the IEEE. Of particular significance is the ImpactCS Project commissioned by the joint taskforce of the ACM and IEEE, and funded by the USA's National Science Foundation. The increased globalization and inherent nature of ICT transcend physical and cultural borders, making it increasingly difficult to enforce accepted laws, regulations, and codes of conduct. It is thus the responsibility of Computer Science and Information Systems instructors to teach and instill professional values and ethical analysis skills in each and every student. Therefore, we investigate some issues pertaining to the teaching of computer ethics. We conclude this paper by presenting a possible framework to be used in the teaching of computer ethics, and apply this framework to our own institution.

Keywords: Computer ethics, curriculum studies, computer science education

Introduction

The terms "ethics", and in particular "computer ethics" (CE), may seem vague and not at all important to many computer scientists, especially those who hold the view that technologists should deal with technology, and philosophers with philosophy and ethics (Couger, 1989). It is therefore perfectly reasonable to expect that the teaching of CE may also be considered by some as of little importance. Yet many computing instructors hold a different view, namely that the teaching of CE to computing students is as important as the teaching of technological topics. In this paper we explore this topic in some detail, in particular what CE is, and why and how it may be taught.

We discuss a generic model based on the ImpactCS proposal (Martin & Yale-Weltz, 1999) that may be used when considering the teaching of CE, and then apply it to our own (distance education) institution. In this paper we restrict our attention to the ACM/IEEE approach to the teaching of CE.

Material published as part of these proceedings, either on-line or in print, is copyrighted by Informing Science. Permission to make digital or paper copy of part or all of these works for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage AND that copies 1) bear this notice in full and 2) give the full citation on the first page. It is permissible to abstract these works so long as credit is given. To copy in all other cases or to republish or to post on a server or to redistribute to lists requires specific permission from the publisher at Publisher@InformingScience.org

Paper Accepted as a Regular Paper

Background

What Is Computer Ethics?

Many authors (Appel, 1998; Couger, 1989; Gotterbarn & Riser, 1997; Pierce & Henry, 1996; Riser & Gotterbarn, 1996; Tavani, 1996) who discuss the topic of ethics simply assume that their audience is conversant with the notion of ethics. We find it useful to state clearly what we consider this broad term to imply, and its particular relevance to CE. The literature offers numerous definitions and notions of the terms ethics and computer ethics (Johnson & Niessenbaum, 1995), but for the purposes of this discussion we concur with Britz (1996) who attempts to define ethics as follows:

> *The ethical actions of a person can be described in general terms as those **actions** which fall within the range of those **activities** that would be regarded as "good". It relates thus to the question of what is good and bad in terms of **human actions**,*

and Baase (1997) who simply says:

> *Ethics is the study of what it means to "do the right thing".*

These definitions may seem somewhat non-precise. Surely that which one individual may understand under the term "good" may differ from that of another individual. The Odhams Dictionary of the English Language defines the term "good" as *a merit, a moral quality, a virtue, an advantage, benefit, profit*. However, what appeals to us here is the strong emphasis on actions, and the practical nature of ethics. This conforms to Aristotle's classification of different kinds of science, viz theoretical, productive and practical science (Moor, 1998), with ethics, politics, and economics constituting the practical sciences. This is but one of the reasons why we consider the teaching of CE to form an integral part of the training of any future ICT professional. Again, Baase (1997) supports this by stating that computer ethics involves "ethical issues faced by a computer professional as part of the job". We therefore describe the term *computer ethics* (CE), as *the study of those behavioural **actions** of ICT professionals that will benefit all of society.*

Why Teach Computer Ethics?

A certain historical perspective is gained by recalling three major contributions in the teaching of computer science, viz. the development of Computing Curricula 1991 (CC91) by a joint task force of the ACM and the IEEE Computer Society as a framework for the previous iteration of the computer science curriculum, the subsequent ImpactCS Project, and the Computing Curricula 2001 (CC2001) for the current iteration of the computer science curriculum.

The *nine core subject areas* in CC91 were algorithms and data structures, architecture, artificial intelligence and robotics, database and information retrieval, human-computer communication, numerical and symbolic computations, operating systems, programming languages, and software methodology and engineering. The recognition of the *social, ethical and professional context of computer science was included as one of the foundational principles*, but CC91 fell short of providing sufficient detail and guidelines for implementation of CE within the curriculum.

In 1994 the subsequent ImpactCS Project was funded by the US National Research Foundation to produce a more rigorous definition of the *content area* of ethics and social impact within computer science, and a *methodology* for integrating these topics across the computer science curriculum.

In 1998 another major review of curriculum guidelines for undergraduate programs in computing, CC2001, was undertaken. *Fourteen knowledge areas* were identified including one entitled *Social and Professional Issues*. CC2001, Draft (February 1, 2001) contains extensive recommendations for all the knowledge areas except for Social and Professional Issues (Computing Curricula 2001). This task still

remains to be completed. Since we are of the opinion that the introduction of CE into our computing programs is a matter of some urgency, we use the ImpactCS Project framework together with the available CC2001 recommendations.

The computing instructor needs to be aware of the numerous ethical and social issues caused by computer technology. According to Martin and Yale-Weltz (1999), these issues have three unique characteristics:

1. new concerns are rapidly emerging,
2. computer ethics presents a continuous stream of new situations, and
3. computerised information systems are usually complex.

Ideally students should be equipped with theories of philosophy and ethics pertinent to these characteristics, as well as the skills to analyse, evaluate, and react appropriately to ethical dilemmas that may arise during their careers as ICT professionals. Forester and Morrison (1994) identified the following main categories (or groupings of topics) in which these ethical and social concerns usually arise:

- computer crime and computer security,
- software theft and intellectual property rights,
- computer hacking and the creation of viruses,
- computer and information system failure,
- invasion of privacy,
- social implications of artificial intelligence and expert systems, and
- workplace computerisation.

It is often within the context of these categories that the behavioural actions of the ICT professional impact on society. We believe this to be one of the main reasons why computing instructors should sensitise their students, the ICT professionals of the future, to actions that underpin concepts such as "good", "moral", "ethical", and "beneficial" for society at large. This belief is in agreement with Searls (1988), who states that the teaching of CE should aim towards several important goals, viz.:

- increased *sensitivity* to ethical concerns and situations, as well as
- *reasoning* about alternative courses of action and the integrity to make moral decisions.

Who Should Teach Computer Ethics?

Computer ethics is by definition a subject that spans multiple disciplines, certainly at least *computer science* and *philosophy*. When considering the teaching of CE an obvious question arises: Should it be taught by philosophers, or should computer scientists take this task upon themselves?

The vast and growing literature on CE distinguishes between at least two main schools of thought. An argument in favour of *philosophers* is based on the view that the *goals* of such a course *should determine who* ought to teach it. According to the philosopher Johnson (1994) these goals are:

1) to make students (future computing professionals) aware of the ethical issues surrounding computers,
2) to sensitise them to ethical issues in the use of computers, and in the practice of the computing professions,

3) to convey to students a deep understanding of the ways in which computers change and impact on society, and
4) to provide conceptual tools and develop analytic skills for ethical decision making within the computing world.

Johnson (1994) maintains that, while having to do with computers, these goals and issues are at base ethical, social, and professional, and that:

> *nothing in the training of computer scientists and engineers prepares them for these types of activities. ... Philosophers are specifically trained to analyse issues, to uncover the assumptions or implications of arguments and claims, and to develop arguments. Philosophers trained in ethical theory have a repertoire of concepts and frameworks that are useful for examining situations, identifying the moral core, and evaluating and justifying courses of action* (Johnson, 1994).

She therefore envisages the philosopher as the teacher, assisted by computer scientists when and where required.

While the goals of Johnson are readily recognised, her conclusions do not enjoy equal acceptance. Indeed, a general view is that *computer scientists* are capable of learning the ethical theories and strategies. Given some training in ethical issues, provided by books, workshops, seminars, etc., and assisted by philosophers when necessary, they would be able to effectively include computer ethics in their curricula (Gotterbarn as quoted in (Johnson, 1994)). Moreover, the fundamental part that social and ethical issues should play in modern computer science would be best emphasised to students by the involvement of senior computer science faculty in the planning, implementation and teaching thereof (Martin as quoted in (Johnson, 1994)). We anticipate that once computer ethics has been established as mainstream computer science, it will be taught and researched like all other fields of computing and that the question under discussion will simply disappear.

Having established the role of computer scientists in the teaching of CE, we now focus on the CE contents to be taught, based on proposals in (Huff & Martin, 1995; Martin, Huff, Gotterbarn & Miller, 1996; Martin & Yale-Weltz, 1999).

What Computer Ethics Topics Should We Teach?

The ImpactCS Project proposes five knowledge units (Martin, Huff, Gotterbarn & Miller, 1996; Martin & Yale-Weltz, 1999) as the basis for CE courseware. For the sake of completeness and readability we list these knowledge units and its associated topics:

Unit 1 - *Responsibility of the Computer Professional*

- 1.a) history, development, and impact of computer technology,
- 1.b) why be ethical,
- 1.c) major ethical models,
- 1.d) definition of computing as profession, and
- 1.e) codes of ethics and professional responsibility for computer professionals (in (Couger, 1989) and (Rosenberg, 1998) the reader is presented with useful information as to the ACM and IEEE codes of conduct).

Unit 2 - *Basic elements of ethical analysis*

- 2.a) ethical claims can and should be discussed rationally,
- 2.b) ethical choices cannot be avoided, and
- 2.c) easy ethical approaches and solutions are questionable, i.e. it is hard.

Unit 3 - *Basic skills of ethical analysis*

3.a) arguing from example, analogy, and counter-example,

3.b) identification of stakeholders in concrete situations,

3.c) identification of ethical issues in concrete situations,

3.d) application of ethical codes to concrete situations, and

3.e) identification and evaluation of possible courses of action.

Unit 4 - *Basic elements of social analysis*

4.a) social context influences the development and use of technology,

4.b) power relations are central in all social interactions,

4.c) technology embodies the values of the developers,

4.d) populations are diverse, and

4.e) empirical data are crucial to the design and development processes.

Unit 5 - *Basic skills of social analysis*

5.a) identification and interpretation of the social contexts of a particular implementation,

5.b) identification of assumptions and values embedded in a particular system, and

5.c) evaluation by means of empirical data of a particular implementation of a technology.

How Should These Topics Be Taught?

Applying this framework to one's own institution, faculty, and department, as well as the formulation of an appropriate educational model, now needs to be considered. The ImpactCS Project (Martin & Yale-Weltz, 1999) identified two main strategies for the teaching of CE. These are:

- the integration of the five knowledge units in the existing material on each level, and
- a stand-alone course dedicated to CE material.

However, it favours the teaching of CE in an integrated fashion (Huff & Martin, 1995; Martin, Huff, Gotterbarn & Miller, 1996; Martin & Yale-Weltz, 1999; Orwant, 1991; Riser & Gotterbarn, 1996). According to (Martin & Yale-Weltz, 1999), the key to the integrated approach lies in the overall coordination of the CE curriculum. One example of this approach is presented by Yale-Weltz (1998) and describes a faculty initiative at the Seattle Pacific University for the teaching of CE at various levels. Cohen and Cornwell (1989) investigated ethical instruction in an integrated manner at Bradley University using a question-asking pedagogy. Although ImpactCS prefers the integrated approach, it is noteworthy that many literature references (Benbunan-Fich, 1998; Couger, 1989; Gotterbarn & Riser, 1997; Roberts, 1998; Schulze & Grodzinsky, 1996; Wahl, 1999) deal with the practical implementation from the point of view of a stand-alone course. A possible reason for this phenomenon could be that although computing instructors are in agreement with the principle of teaching CE in an integrated fashion, the practical implementation thereof is problematic.

Various examples of stand-alone courses are presented in the literature. Roberts (1998) presents a curriculum as well as assignments to be completed by students, ranging from written, oral, to on-line participation for his course. Wahl (1999) presents yet another stand-alone course curriculum with reference to Project ImpactCS, ACM, and IEEE professional codes of conduct. Other examples may be found in (Benbunan-Fich, 1998; Couger, 1989; Gotterbarn & Riser, 1997; Schulze & Grodzinsky, 1996).

While we in principle support the integrated approach suggested by ImpactCS, it is clear from the literature (Benbunan-Fich, 1998; Couger, 1989; Gotterbarn & Riser, 1997; Roberts, 1998; Schulze &

Grodzinsky, 1996; Townsend, 1999; Wahl, 1999) that in most cases, departmental pragmatics and logistics necessitate a combination of the integrated and stand-alone approach. Since our goal is to make *all* our computing students aware of the issues in CE, and to equip them with the capabilities and skills of evaluation, decision-making, and appropriate responsible action, we focus our attention on core (compulsory) modules as the vehicle for teaching CE. Our combined approach will facilitate an early introduction (on first level), continued discussion at following levels, and the integration of CE topics in the core modules. Together with our subsequent stand-alone third level module we expect to achieve maximum coverage with minimum overlap (Martin & Yale-Weltz, 1999). For this reason we propose a generic integration of the five ImpactCS project knowledge units and the associated topics, into the computing curriculum.

An Integration Plan

Our own institution, the University of South Africa (Unisa), is a distance education one. It is one of 11 mega distance education universities (more than 100 000 students) in the world. Its Department of Computer Science and Information Systems, with a teaching staff of over 50, has more than 11 000 students, enrolled for 29 000 module papers - arguably the largest computing department on the African continent! Indeed, teaching CE to thousands of students by means of distance education is certainly a worthwhile challenge.

A Team Approach in Teaching Computer Ethics

The BSc (three year) degree programme offered by Unisa has a modular structure, with fourteen modules at first level, eight at second level, and eight at third level. The teaching is usually done in *small teams*, with particular emphasis on planning, coordination and student support. In our computing department, each module is offered by a team consisting of two to four lecturers, who are collectively responsible for the planning, implementation and teaching of their module(s). Each individual is typically involved in the teaching of two undergraduate modules, with student numbers per undergraduate module ranging from 100 to 1600 or more. This team approach turns out to be ideally suited to the teaching of CE, using our combined approach.

The CE team is responsible for:

- Ensuring that the identified topics (shown in the following section) are introduced into the existing module structure.
- Designing and teaching a third level module that addresses all of the topics not covered by the integration process.

The advantages of this dedicated team based approach to the teaching of CE include the following:

- Lecturing staff involved with the instruction of the core modules will not be burdened with developing and teaching the CE topics as the CE team will be responsible for this.
- It guarantees the structured and co-ordinated teaching of ethical issues and concerns embodied by the topics, minimizing overlap yet maximizing coverage of ethical issues.

The Status Quo at Unisa

When embarking on such a project or investigation, it makes sense to take note of the status quo. So, let us briefly examine what is offered in terms of the teaching of CE in our courseware at Unisa at present.

While many of the textbooks and study material used by our department may in passing refer to aspects of CE, we only list those modules where it assumes a somewhat more central role. For each module we also list the relevant ImpactCS topics addressed, in curly brackets { }:

End User Computing (ICDL Study Guide, 2002)

Topics: Security, copyright and the law, treating issues such as understanding software copyright and security questions, awareness of privacy issues, data protection, viruses and anti-virus measures, what happens in the case of a power failure. {1.a, 1.b, 4.a}

Computer Systems - Fundamental Concepts (Shelly, Cashman , Vermaat & Walker, 2002)

Topic: Netiquette - focussing on the unwritten rules for acceptable conduct when using the internet. {None}

Human Computer Interaction (Kotzé et al, 2002)

Topics: Netiquette, security of digital data, safety and the repercussions of computer hardware and software failure, the provocative question concerning human failure (which may include unethical conduct) is also posed. {1.a, 1.b, 3.b, 4.a}

We observe:

- The modules in question are all first level modules.
- Even at this basic level the overlap is significant.
- *End User Computing* and *Computer systems - fundamental concepts* are core modules in the BCom Informatics and the BSc degree programs respectively. *Human Computer Interaction* is an elective module.
- The positive side however is that students are sensitised at an early stage in their studies, and the spreading of the CE material across a number of modules emphasises the importance thereof.
- The negative side is that CE is neglected at the second and later levels. Many of the key CE issues and topics are not addressed.
- The ImpactCS topics 1.a, 1.b, 3.b, and 4.a are addressed in these modules. However not in an integrated manner, with the bulk of these topics dealt with in *Human Computer Interaction*, an elective.

Selecting computer ethics topics for modules

In order to facilitate the integration of CE topics in the computer science programs of the department (a similar iteration can be performed for the IS programs), we follow the steps listed below:

Step 1 - identify suitable core modules

First of all we identify the following level 1 core modules:

Computer systems - fundamental concepts,

Introduction to programming,

We also identify the following level 2 core modules:

Programming: data structures, and

Programming: practical.

Step 2 - Identify the CE and ImpactCS topics that can be addressed by the existing study material or textbook information

Three programming core modules as well as one introductory module have been identified as modules which allow the inclusion of certain aspects of CE. We briefly mention the CE topics which could be incorporated in these modules - the associated ImpactCS topics are indicated in round brackets ( ).

Computer systems - fundamental concepts

- History and evolution of computers and their impact on society (1.a, 4.a, 4.b).
- Virus protection, security and safety issues (1.a, 3.b, 3.c, 3.e).
- Internet-related security and privacy issues (3.c, 3.e, 4.a).

Introduction to programming

- The history and evolution of computers and their impact on society (1.a, 4.a, 4.b).
- Copyright and virus protection (3.b, 3.c, 3.e).

Programming: data structures

- Ethical dilemmas should be introduced to students, with an emphasis on situations, which present difficult choices, sometimes even no "good" option (2.a, 2.b, 2.c).
- Arguing from example, analogy, and counterexample (3.a).

Programming: practical

- Ethical models should be studied informally, and rational decision-making should be emphasized (1.c).
- ACM and IEEE codes of conduct should be introduced (3.d).
- Students must be sensitised to ethical issues related to program design and should be provided with at least one programming assignment in which they should identify some of the ethical and social issues pertaining to the specific program (3.c). Applicable examples may be found in (Gotterbarn & Riser, 1997; Riser & Gotterbarn, 1996).

Step 3 - Minimise overlap

From step 2 it is clear that several topics are repeated in more than one core module. With the limited time available to devote to the teaching of CE in an already full computing curriculum, it therefore makes sense to follow the suggestion of Martin and Yale-Weltz (1999), and attempt to integrate the CE topics into the computing curriculum to ensure minimal overlap of topics in the identified core modules (for example topic 1.a is addressed by the modules *Computer systems - fundamental concepts* and *Introduction to programming*). In order to identify which topic is best integrated into which module, (Yale-Weltz, 1998) provides useful information. Below we show which topic(s) are integrated into which core mo-dule(s) as part of our CE integration plan, in curly brackets { }:

Computer systems - fundamental concepts

- History and evolution of computers and their impact on society (1.a, 4.a, 4.b).
- Internet-related security and privacy issues (4.a).

The topics to be integrated into *Computer systems - fundamental concepts* is thus: {1.a, 4.a, 4.b}.

Introduction to programming

- Copyright and virus protection (3.b, 3.e).

The topics to be integrated into *Introduction to programming* is thus: {3.b, 3.e}.

Programming: data structures

- Ethical dilemmas should be introduced to students, with an emphasis on situations which present difficult choices, sometimes even no "good" option (2.a, 2.b, 2.c).
- Arguing from example, analogy, and counterexample (3.a).

The topics to be integrated into *Programming: data structures* is thus: {2.a, 2.b, 2.c, 3.a}.

Programming: practical

- Ethical models should be studied informally, and rational decision-making should be emphasized (1.c).
- ACM and IEEE codes of conduct should be introduced (3.d).
- Students must be sensitised to ethical issues related to program design and should be provided with at least one programming assignment in which they should identify some of the ethical and social issues pertaining to the specific program (3.c).

The topics to be integrated into *Programming: practical* is thus: {1.c, 3.c, 3.d}.

We therefore propose that the following ImpactCS topics be integrated into our present computing curriculum: {1.a, 1.c, 2.a, 2.b, 2.c, 3.a, 3.b, 3.c, 3.d, 3.e, 4.a, 4.b}.

Step 4 – Compile the remaining topics into a capstone module

The remaining topics will constitute the basis of the stand-alone capstone course to be developed and taught in the final year of study. They are: {1.b, 1.d, 1.e, 4.c, 4.d, 4.e, 5.a, 5.b, 5.c}.

While this step-wise approach attempts to guarantee maximum coverage with minimum overlap, it remains the prerogative of the CE team to repeat certain topics for the purpose of deepening knowledge and insight.

The remaining topics listed above to be included in the capstone module, can be roughly divided into three groups. The three groups are as follow:

1. Generic CE-topics

- Formal discussion of major ethical models (1.c).
- Formal approach towards ethical decision-making (2.a, 2.c, 3.b).
- Professionalism (1.b, 1.d, 1.e).
- Codes of ethics, ACM and IEEE codes of conduct (1.e, 3.d).

2. Concepts concerning programming languages

- The impact of the designers' values and assumptions (4.a, 4.c, 5.b).
- Exception handling (5.c).
- Security and safety issues (1.a, 3.d).
- Extension and ownership (1.a, 3.b).

3. Concepts concerning system design

- The developer's responsibility (1.a, 4.e).

- The implications of the fact that systems are designed for real people of different social and cultural backgrounds (4.a, 4.d, 4.e, 5.a).
- File systems, security protection from hackers, viruses, worms and Trojan horses (1.a, 4.b).

Summarising, we discussed steps for mapping the ImpactCS topics to module in a meaningful manner. We also applied these steps to our own institution and showed how we integrated the teaching of CE into our particular existing module structure. In the Appendix we provide an example of the computer ethics specific tuition material for use in the core module *Computer systems - fundamental concepts.*

Conclusion and Future Work

In this paper we proposed a framework for introducing CE into our computing curriculum. A key feature of our teaching methodology is the *dedicated team approach.* This approach ensures complete coverage of relevant computer ethics topics in a unified way. This is made possible by a team, dedicated to the integration of computer ethics topics throughout the curriculum. We trust that this will sensitise future IT professionals to take note of and appreciate the importance of existing ethics codes of conduct and of ethical behaviour.

As point of departure we used the ACM/IEEE curriculum proposals. In our ongoing efforts to research and integrate the teaching of CE into our curriculum, we plan to take note and introduce other suitable curriculum proposals and teaching approaches as used, for example, in the UK, Europe and Australia.

We are presently conducting empirical research regarding the effectiveness of our approach. This include, among others, the provision of CE tuition material to students enrolled for the module *Computer systems - fundamental concepts*, the distribution of a questionnaire designed to measure ethical awareness of students and the statistical analysis of the results.

In conclusion we support the ACM/IEEE Joint Curriculum Taskforce's position regarding the teaching of CE (Martin & Yale-Weltz, 1999):

> *Undergraduates need to understand the basic cultural, social, legal, and ethical issues inherent in the discipline of computing. They should understand where the discipline has been, where it is, and where it is heading. They should understand their individual roles in this process, as well as appreciate the philosophical questions, technical problems, and aesthetic values that play an important part in the development of the discipline. ... Students also need to develop the ability to ask serious questions about social impact and to evaluate proposed answers to those questions.*

References

Appel, F. (1998). Including the social and ethical implications of computing in the computer science curriculum, *Computers and Society*, Vol. 28, no. 2, pp.56-57.

Baase, S. (1997). *A gift of fire: Social, legal, and ethical issues in computing*, Prentice Hall, Upper Saddle River, N. J.

Benbunan-Fich, R. (1998). Guidelines for using case scenarios to teach computer ethics, *Computers and Society*, vol. 28, no. 3, pp.20-24.

Britz, J. J. (1996). Technology as a threat to privacy: ethical challenges and guidelines for the information professionals, *Microcomputers for Information Management: Global Internetworking for Libraries*, 13 (3-4), pp.175-193.

Cohen, E. and Cornwell, L. (1989). A question of ethics: developing information system ethics, *Journal of Business Ethics*, 8, 431-437.

Computing Curricula, (2001. *Computing Curricula 2001 - DRAFT version, (February 1, 2001)*; http://www.acm.org/sigcse/cc2001/,.

Cortada, J. W. (2002) Researching the History of Software from the 1960's; *IEEE Annals of the History of Computing*, Jan. - March 2002, pp. 72-79 .

Couger, J. D. (1989). Preparing IS students to deal with ethical issues, *Management Information Systems Quarterly*, Vol. 13, no. 2, pp.211-217.

Forester, T. and Morrison, P. (1994). *Computer ethics: cautionary tales and ethical dilemmas in computing*, Second edition, MIT Press, Cambridge MA.

Gotterbarn, D. and Riser, R. (1997). Ethics activities in computer science courses: goals and issues, *Computers and Society*, vol. 27, no. 1, pp.10-15.

Huff, C. and Martin, C. D. (1995). Computing consequences: a framework for teaching ethical computing, *Communications of the ACM*, Vol. 38, no. 12, pp.75-84.

Hutchinson, S. E. and Sawyer S. C. (200) Computers, Communications & Information, A User's Introduction, Seventh Edition, McGraw-Hill, Boston, MA.

ICDL Study Guide, (2002). *ICDL Module 1: Basic concepts of IT*, University of South Africa, Pretoria. Published under licence from ElementK, formerly Ziff-Davis Education, Rochester, N. Y.

Johnson, D. G. (1994). Who should teach computer ethics and computers and society?, *Computers and Society*, Vol. 24, no. 2, pp.6-13.

Johnson, D. G. and Niessenbaum, H. (1995). *Computers, ethics, and social values*, Prentice-Hall, Inc., Upper Sadle River, NJ.

Johnson, D. G. and Snapper, J. W. (1985); *Ethical Issues in the Use of Computers*, Wadsworth, Belmont, CA.

Kizza, J. M. (1985). Civilizing the Internet - Global Concerns and Efforts Towards Regulation, McFarland and Company, Jefferson, NC.

Kotzé et al, (2002). *Study guide for INF120*, University of South Africa, Pretoria.

Martin, C. D., Huff, C., Gotterbarn, D. and Miller, K. (1996). Implementing a tenth strand in the CS curriculum, Communications of the ACM, Vol. 39, no. 12, pp.75-84 .

Martin, C. D. and Yale-Weltz, E. (1999). From awareness to action: Integrating ethics and social responsibility into the computer science curriculum, *Computers and Society*, Vol. 29, no. 2, pp.6-13.

Moor, J. H. (1998). If Aristotle were a computing professional, *Computers and Society*, Vol. 28, no. 3, pp.13-16.

Orwant, J. C. (1991). Computer ethics - part of computer science!, *Computers and Society*, Vol. 21, no. 2-4, pp.40-45.

Pierce, M. A. and Henry, J. W. (1996). Computer ethics and social issues case analysis: an environmental algorithm, *SIGCSE Bulletin*, vol. 28, no. 4, pp.50-54.

Pistorius, M. C. (1995) COS113 study guide, Pistorius et al, UNISA.

Riser, R. and Gotterbarn, D. (1996). Ethics activities in computer science courses, *Computers and Society*, Vol. 26, no. 3, pp. 13-17.

Roberts, E. (1998). Strategies for using technology in the teaching of ethics, *SIGCSE Bulletin*, vol. 30, no. 3, pp.209-212.

Rosenberg, R. S. (1998). Beyond the code of ethics: The responsibility of professional societies, *Computers and Society*, Vol. 28, no. 2 , pp.18-25.

Schulze, K. G. and Grodzinsky, F. S. (1996). Teaching ethical issues in computer science: what worked and what didn't, Proceedings of the Twenty-Seventh SIGCSE Technical Symposium on Computer Science Education, *SIGCSE Bulletin*, vol. 28, no. 1, pp.98-101.

Searls, D. E. (1998). Teaching computer ethics, *SIGCSE Bulletin*, Vol. 20, no. 3, pp. 45-48.

Shelly, G. B., Cashman, T. J., Vermaat, M. E. and Walker, T. J. (2002). *Discovering computers 2003: concepts for a digital world, Web and XP enhanced*, Thomas Learning, Cambridge MA.

Stair, R. M. and Reynolds, G. W. (1999); *Principles of Information Systems, fourth edition*, Thomson Course Technology, Cambridge, MA.

Tavani, H. T. (1996). Selecting a computer ethics coursebook: a comparative study of five recent works, *Computers and Society*, vol. 26, no. 4, pp.15-21.

Townsend, G. C. (1999). The TenthStrand == 3*Ethical Debate+Solution, *SIGCSE Bulletin*, vol. 31, no. 1, pp.17-21.

Wahl, N. J. (1999). YAATCE - Yet another approach to teaching computer ethics, Proceedings of the Thirtieth SIGCSE Technical Symposium on Computer Science Education, *SIGCSE Bulletin*, vol. 31, no. 1, pp.22-26.

Yale-Weltz, E. (1998). A staged progression for integrating ethics and social impact across the computer science curriculum, *Computers and Society*, Vol. 28, no. 1, pp.30-34.

Appendix: The Tuition Material for Computer Systems - Fundamental Concepts

In this appendix we explore several aspects pertaining to the creation of the CE tuition material to be integrated into the core module *Computer Systems - fundamental concepts*.

ImpactCS topics and related information

In the article we explored the five knowledge units and its associated topics proposed by the ImpactCS project as a basis for the integration of CE topics into the core modules (Martin & YaleWeltz, 1999). This was expanded on in Step 3 where we identified the topics to be integrated into the *Computer Systems - fundamental concepts* curriculum as: {1.a, 4.a, 4.b}.

We note that during the current iteration of the computing curriculum proposed by a joint task force of the ACM and IEEE, topics regarding CE to be included in the computing curriculum (Computing Curricula, 2001) are also addressed. Of the *fourteen knowledge areas* identified, knowledge area 12 deals with the so-called *Social and Professional Issues* (referred to as SP units throughout the rest of this paper). We only list those CC2001 topics that are relevant to our discussion of the ethics course material for *Computer Systems - fundamental concepts* to be integrated, and that will serve to supplement the ImpactCS topics given above (refer to (Computing Curricula, 2001) for a complete listing of all topics):

SP.1 (**History of computing**) elaborates on the contents and learning objectives of ImpactCS topic 1.a and suggests that the following issues should be addressed in the curriculum:

SP.1.1 prehistory of the world before 1946,

SP.1.2 history of computer hardware, software, and networking, as well as

SP.1.3 pioneers of computing.

In the CC2001 proposal (Computing Curricula, 2001) *learning objectives or outcomes* associated with SP.1 (referred to as SP.1.O's) are incorporated into said proposal and is listed below:

SP.1.O.1 list the contributions of several pioneers in the computing field,

SP.1.O.2 compare daily life before and after the advent of personal computers and the Internet, as well as

SP.1.O.3 identify significant continuing trends in the history of the computing field.

SP.2 (**Social context of computing**) elaborates on the contents and learning objectives of ImpactCS topic 4.a and suggests that the following issues should be addressed in the curriculum:

SP.2.1 introduction to the social implications of computing,

SP.2.2 social implications of networked communication, as well as

SP.2.3 growth of, control of, and access to the Internet.

Again CC2001 proposes some learning objectives (outcomes) to supplement the course material and assessment strategies:

SP.2.O.4 describe positive and negative ways in which computing alters the modes of interaction between people.

SP.2 (**Social context of computing**) elaborates on the contents and learning objectives of ImpactCS topic 4.b and suggests that the following issues should be addressed in the curriculum:

SP.2.3 growth of, control of, and access to the Internet, as well as

SP.2.5 international issues.

CC2001 also proposes some learning objectives (outcomes) to supplement the course material and assessment strategies:

SP.2.O.5 explain why computing / network access is restricted in some countries.

What remains is to apply the ImpactCS framework (supplemented by the CC2001 topics and outcomes) to our own Department, and in this Appendix in particular, to the *Computer Systems - fundamental concepts* course material.

Computer Systems - fundamental concepts: the integration phase

The purpose of *Computer Systems - fundamental concepts* is to introduce students to the computer as a system. This covers hardware concepts such as internal representation of numbers, characters, and basic computer architecture, as well as software concepts such as systems software and application software. A brief introduction to databases and system analysis and design is also included. The prescribed book for *Computer Systems - fundamental concepts* is *Discovering Computers 2003, Concepts for a Digital World, Web and XP enhanced* (Shelly, Cashman , Vermaat & Walker, 2002), referred to as Shelly et al throughout the rest of this appendix. Shelly et al addresses various ethical and social issues, but does not cover all the relevant ethical and social issues to be included in this module. Thus our study guide (necessitated by our distance mode of teaching) supplements the textbook in order to address all the relevant issues to be included in this module.

Topic 1.a - History, development, and impact of computer technology:

Shelly et al:

• SP.1.1 - A timeline indicating the milestones in computing history (Shelly et al, 2002, chapter 1, pp.48-60).

• SP.1.1 - The history of the Internet (Shelly et al, 2002, chapter 2, pp.3-4).

Study guide:

• SP.1.1 and SP.1.O.2 - Prehistory of the world before 1946, e.g. comparing daily life before and after the advent of personal computers and the internet, (Pistorius, 1995, pp.6-16).

• SP.1.1 and SP.1.O.3 - The history of the hardware and software of general purpose computers, divided into four generations, are covered by (Hutchinson & Sawyer, 2000, pp.117-119) and (Cortada, 2002, pp.73-74).

• SP.1.2 - History of the Internet, including the APRANET, MILNET, Internet Protocol and Internet 2 (Stair & Reynolds, 1999, p.291).

• SP.1.2 and SP.1.O.3 - The Internet globalisation process that facilitated the spread of cybercrimes. Attention is paid to the following cybercrimes related to the Internet:

  - Espionage viz Internet information gathering (Kizza, 1989, p.50).
  - Internet fraud (Kizza, 1989, p.48).

• SP.8.1 - The history of computer crime and examples of computer crime (Forester & Morrison, 1994, pp.1-27).

• SP.1.3 and SP.O.1 - The contributions of the following four computing pioneers will be discussed:

  - Alan Turing
  - John von Neumann
  - Edsger Dijkstra
  - Alan Kay

Topic 4.a - Social context influences the development of technology

Shelly et al:

• SP.2.1, SP.2.2, SP.2.3 and SP.2.O.4 - Certain social issues are covered in (Shelly et al, 2002, chapter 11), *Computers and Society: Home Work and Ethical issues*, as well as (Shelly et al, 2002, chapter 12), *Computers and Society: Security and Privacy*). The issues relevant to our curriculum are:

  - The use of computers at home.
  - The evolution of society re education, entertainment, finance, government, health care, science, publishing, and travel brought about by the age of computing.
  - The digital divide.
  - E-commerce and its affects regarding the conduction of business.
  - The manners by which virtual reality, intelligent agents, and robots are being used in daily life.
  - The prevention of injuries and health related disorders due to computer use.
  - Symptoms of computer addiction.
  - Green computing.
  - Ethical issues surrounding computer use, amongst others information accuracy.
  - The ergonomic design of the workplace.

Study guide:

• SP.2.1 and SP.2.2 - Social implications of computing and networked communities:

  - New range of social problems or issues (Forester & Morrison, 1994, p.4).
  - Downside issues relating to the use of the Internet and Web (Hutchinson & Sawyer, 2000, chapter 8, pp.25-26).

Topic 4.b - Power relations are central in society

Study guide:

• SP.2.O.4 - Computers and social power (Johnson & Snapper, 1985, pp.271-280):

  - The relevance of power.
  - Power, influence, authority and organisations.
  - Routine use of AIS (Automated Information Systems) in organisations.
  - New reporting systems.
  - Shifts of power and influence across organisational boundaries.

• SP.2.3, SP.2.5 and SP.2.O.5 - International issues (Kizza, 1989, pp.97-145):

  - Regulating the Internet.
  - Network access restrictions in some countries.

Biographies

Andries Barnard, associate professor in the Department of Computer Science and Information Systems, University of South Africa, holds a PhD (Computer Science). He teaches undergraduate courses in automata theory and formal languages and project management, as well as postgraduate courses in project management and research methodology. His research interests include computer ethics and graph grammar languages.

Corné de Ridder, a postgraduate student in the Department of Computer Science and Information Systems, University of South Africa, is currently enrolled for an MSc (Computer Science). Her research interests include computer ethics and related philosophical theories.

Laurette Pretorius, professor in the Department of Computer Science and Information Systems, University of South Africa, holds a PhD (Applied Mathematics). She teaches undergraduate courses in numerical methods, automata theory and formal languages and computer graphics, as well as postgraduate courses in computability theory and research methodology. Her research interests include computer ethics and natural language processing, in particular computational morphology of the African languages.

Eli Cohen is, among other others, full professor at the Leon Kozminski Academy of Entrepreneurship and Management (Warsaw), principal, Informing Science Institute, principal, Management Technologies, editor of the Informing Science Journal, and editor of the Journal of IT Education. He holds a PhD (Measurement and Methodology, Statistics) from the Indiana University. His research interests are interdisciplinary and spans a wide variety of fields, including computer ethics.