
On the Difficult Task of Teaching Computer Ethics to Engineers

Patrignani, Norberto
Catholic University of Milano.

Kavathatzopoulos, Iordanis
Uppsala University.

Corresponding Author: Norberto Patrignani, norberto.patrignani@polito.it

Abstract

This paper addresses the challenges of teaching computer ethics to engineers. The computer professionals are identified as the starting point of the stakeholder's network of ICT and as a consequence it is underlined the importance of computer ethics courses for engineers. To this purpose, a simple four-steps methodology is proposed for teaching computer ethics. The importance of applying the paradigm of complex systems is then described and the three dimensions of Slow Tech (good, clean, and fair ICT) are proposed as a compass for designing complex socio-technical systems. Finally some preliminary results coming from the feedback of about some hundreds students in several years are illustrated.

Keywords: stakeholders network, proactive computer ethics, complex socio-technical systems, Slow Tech.

Introduction

In order to address the many aspects of the social and ethical impacts of computing, one of the most useful tools is the so called "stakeholders network". By designing a typical stakeholders network related to Information and Communication Technologies (ICT), one will find a complex net of nodes and relationships (see fig.1):

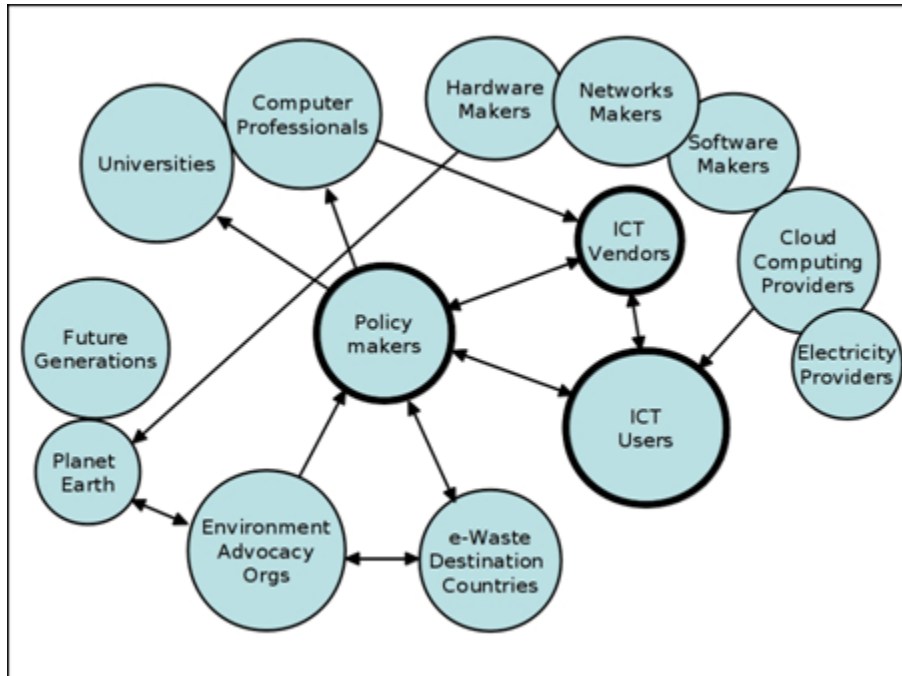


Figure 1. A typical ICT stakeholders' network

It is very clear that at the core of this network there are the three main stakeholders: ICT vendors, ICT Users, and Policy makers. But the main agents that are shaping the society are the large ICT vendors: the companies that are providing ICT innovations to society. At the "core" of these companies one will find the main actors: the persons that design ICT products, hardware, software, networks, and services and related Human Computer Interactions (HCI). They use all the technologies (from cloud computing to big data, from the Internet-of-things to machine-learning, etc.) for developing new products and services that become innovations for billions of users and for a gigantic market that is now around 3.5 Trillion Dollars (Gartner, 2017). These persons are the ICT designers: computer professionals. Due to their role, they have a growing impact on society and a growing responsibility in their hands towards to users and the environment. For this reason this paper concentrates the attention on computer professionals the 'core' stakeholders of ICT. The paper starts from the definition of computer ethics, then investigates the educational pipeline that creates these persons and in particular on the introduction of the subject "Computer Ethics" in universities. The question then becomes: what could be a good methodology for teaching computer ethics to (computer) engineers? What tools could be introduced in these computer ethics classes for engineers? It is proposed a simple methodology based on case-based analysis, complex systems, and Slow Tech (Patrignani & Whitehouse, 2014). Then some preliminary results from the experiences of computer ethics courses with PhD students in engineering faculties are described. In the conclusions of the paper some further directions of future studies are proposed.

Concentrating on computer professionals

The field of Computer Ethics has evolved since 1950 when the founder, Norbert Wiener, recommended to take into account the social and ethical impacts of computing. From the very beginning of his work, at the dawn of computer age, Wiener invited the scientific community to recognize its responsibility in front of society (Wiener, 1954). Nowadays, there are many definitions of Computer Ethics. The definition of Walter Maner: as the field of study that examines "*ethical problems aggravated, transformed or created by computer technology*" (Maner, 1980). The definition of James Moor: as the discipline that should help us in clarifying and filling the "*policy vacuums*" created by new information and communication technologies (ICT), an applied ethics deserved by those special machines due to their "*logical malleability*" (Moor, 1985). The definition of Deborah Johnson: as the applied ethics needed for facing the "new twists" given by computers to well-known ethical questions: "*computers pose new version of standard moral problems and dilemmas*" (Johnson, 1985). The Terry Bynum's definition: "*identification and analysis of the impacts of information technology on such social and human values like health, etc. ...*", an applied ethics with a broad view (Bynum, 2009). Just to mention the most famous ones. But the one that is more relevant in the context of this paper is the Donald Gotterbarn's definition: "*... the values that guide the day-to-day activities of computing professionals*" (Rogerson, 2009). According to Gotterbarn, it is important to narrow the focus of computer ethics around the choices and actions that computer professionals made on their daily activities and projects. Also, Simon Rogerson suggests that, in order to "*make a difference*", computer ethics reasoning need to be incorporated into the project management standards guidelines. In particular he suggests to identify the "*ethical hotspots of project management ... points where activities and decisions are likely to include a relatively high ethical dimension*" (Rogerson, 2009). This can be done by mapping the several steps of structured project management (visualize what the goal is, etc.) to the principles of ethics (honor, etc.). More recently Sara Spiekermann focuses her analysis on the practical processes and tools needed by computer professionals for incorporating values into the computer systems design. What she calls "*Ethical Value-Based IT System Design*", is the incorporation of a value-based approach in the typical processes of products (and services) development: discovery, scoping, build-business-case, development, testing and validation, launch followed by post-launch review (Spiekermann, 2016).

This paper elaborates on the ground of the Gotterbarn's approach: in order to make a difference, computer ethics should be part of the computer professionals background. But where do computer professionals come from?

Computer ethics for engineers

Most of computer professionals come from computer science or computer engineering university studies. An important milestone in the history of computer ethics was indeed the introduction of the "Social and Professional Issues" content in the undergraduate computing curriculum in the US in 1991. Since then, the Computer Science Body of

Knowledge formally includes seven core units (History of computing, Social context of computing, Methods and tools of analysis, Professional and ethical responsibilities, Risks and liabilities of computer-based systems, Intellectual property, Privacy and civil liberties) and three elective units (Computer crime, Economic issues in computing, Philosophical frameworks) (Tucker, 1991). On the same year another related milestone in the history of computer ethics was the event National Conference on Computing and Values (NCCV) that was held on the campus of Southern Connecticut State University in August 1991. Among the several tracks of that conference there was one dedicated to computer ethics education: "Teaching Computing and Human Values" (Bynum, Maner, & Fodor, 1991). The materials of this publication represent a basic starting point for those involved into the computer ethics education as it contains contributions by main scholars such as Terry Bynum, Keith Miller, Donald Gotterbarn, Dianne Martin, Batya Friedman, Hilary Holz, Doris Lidtke. One year before, another historical material for computer ethics education is published by Computer Professionals for Social Responsibility (CPSR): a collection of course syllabi on "Computing and Social Responsibility" (Friedman & Winograd, 1990). Among the several interesting clusters there is and interesting one, "Ethics for computer professionals", with contributions by Terry Bynum and Batya Friedman.

A proposed methodology for teaching computer ethics to engineers

Among the several possible approaches for introducing computer ethics for engineers the one that, in our experience, is the most fruitful is the "*case-based*" one. This methodology expose students to a real case, then divides the class into three main groups of stakeholders, ICT Vendors (or, more precisely, the "*developers*", the "*techies*", ICT Users, and Policy makers, and starts a brainstorming session.

The first goal of this discussion is to write down a detailed description of the scenario under analysis. For and engineer, this apparently simple task, is already a very useful exercise: with simple words and without using technical details they should be able to provide a description of the case (that of course includes complex computer systems) understandable to a lay person. In this step students develop the skill of "social analysis": the capability of describing a scenario that could include many level of social analysis, from individuals to communities, from organizations to cultures, institutions, at national and global level!

The second task for the students is to design the complete stakeholders network involved with the case under analysis. This simple task for many engineers represents the opportunity to discover the many agents impacted by the complex systems that they design, in a way, it is a reflecting action potentially able to trigger an initial ethical awareness and skills (Kavathatzopoulos, 2012), to handle the problem at hand (Kavathatzopoulos & Laaksoharju, 2010).

The third task, after a sufficient reflecting time to analyze the stakeholders network, is to identify possible conflicts and ethical dilemmas. Here the background provided by the teacher is very important, since for many of them it is the first time that they establish a soft link between the technologies they work on and ethics.

The fourth and final task for students is to make some proposals, maybe just elaborating possible different scenarios (that could mitigate, if not solve, the ethical dilemmas).

This four steps methodology for addressing computer ethics dilemmas has been introduced at graduate school of Politecnico of Torino since 2008 with excellent results: it is the most requested course by PhD students.

The background for this course is strongly related to the approaches proposed by Gotterbarn and Bynum. In his "*Captstone course in computer ethics*" Gotterbarn proposes a mixed approach requiring both technical and philosophical competences. The goals of the "captstone" course include: "*student socialization into professionals norms, student recognition of role responsibilities, student awareness of the nature of the profession, student ability to anticipate ethical issues, student ability to reason from professional standards to practical applications, student ability to solve ethical problems morally or technically*" (Gotterbarn, 1991) In his method for case analysis Bynum propose the steps: "*take the ethical point of view, develop a detailed description of the case to be analyzed, try to 'see' the ethical issues and any 'traditional' solutions that fit the case, call upon your own ethical knowledge and skills, get the advice of others, get advantage of one or more systematic analysis techniques, draw relevant ethical conclusions about the case, draw relevant lessons about the future*" (Bynum, 2009).

Teaching computer ethics to engineers is a difficult but fascinating task: exploring and questioning a real scenario together with students with a bottom-up approach is the opening of a new communication channel between teacher and students. After this step the entire "*learning community*" starts making an ethical reflection exercise.

The above approaches have something in common: they all start from the analysis of a real case, then they propose a discussion and a "description" of the case. This is the first challenge for a student coming from several years of technical education: to use a simple language in a text! Articulating a textual description (no formal languages, no formulas, no models!) it is in itself an "eye-opening" experience: students start realizing that the technical components are part of a much more complex (and interesting!) scenario. In many cases they discover that there is something that could be right or wrong with ICT: innovations are not considered just as technical systems but as *socio-technical systems* (Bynum, 2009).

It is also useful to organize the group in several teams for "role playing": the Developers (the "techies"), the Users, and the Policy Makers. This "trick" enforces them to deepening their knowledge about the content related to the case but from a precise perspective.

Then the teachers stimulate with probing questions the debate and try to keep the discussion on three main levels of discussion: brainstorming (just free discussion with

somebody keeping notes or putting stickers on a wall), a "soft" normative approach (can be useful a Code of Ethics? Some "soft" law?), laws (can policy makers define new legal frameworks useful in the case under analysis? Some "hard" laws?).

Also, often during the discussion it is quite usual to see how spontaneously emerge four different dimensions of the discussion: "Architecture" (engineers that proposes a "techno-fix" approach, technology creates and solve(!) ethical dilemmas), "Education" (the importance to educate people towards a wise use of technology), "Market" (no problem, the market will spontaneously find a solution), and "Law" (we need an "authority" able to enforce a norm). It is impressive to note that it is very close to the framework explored by Lawrence Lessig in 1999, in his ground-breaking book on cyberspace where he introduces a simple model for the regulation of complex scenarios that need governance (Lessig, 2002).

The methodology can be summarized by the four steps:

1. SCENARIO (describe and analyze a real case)
2. STAKEHOLDERS' NETWORK (identify stakeholders and their relationships)
3. ETHICAL ISSUES (identify social and ethical issues)
4. ALTERNATIVE SCENARIO(s) (envisage and evaluate alternative courses of action).

By applying this methodology engineers are also acquiring skills that enable them to be compliant with the new requirements coming from the International Federation for Information Processing (IFIP): "... *definition of computer professional: ... Given the reach of ICT in our lives, it is important for an ICT Professional to be ... Technically Strong (in order to use the Right Technology for the Relevant Problem), Ethically Grounded (to ensure that Technology is put to the Right Use), Socially Conscious (so that the technical solution takes into consideration elements of Sustainability), Business Savvy (to ensure commercial viability which is required for Social Prosperity and Funding of new Developments)* (International Professional Practice Partnership, 2015). Finally, after more than sixty years, the recommendations of Wiener, the founder of computer ethics are becoming a reality for the entire ICT international community.

Complex Systems and computer engineers

By reflecting on the complexity of the ICT stakeholders network, students realize that the traditional approach of computer-based systems' design was based on too many simplifications, the most important one: the human beings were considered out of the picture. By including human beings into the scenario, the system becomes a "*complex socio-technical system*". This concept is very important for computer professionals and for their computer curricula in university. Computer science curricula tended to avoid this approach since cybernetics was dropped from tertiary education. The deepening specialization of scientific research or the growing market pressure for growing numbers of programmers maybe some of the reasons. Since then computer scientists and engineering students are concentrated on programming. Only recently the need for a

more trans-disciplinary approach is recognized. Complex systems enter into the scenario: they contain a large number of interacting variables, exchanging messages and information - their main characteristic is the "emerging property". These systems cannot be analyzed by a simple formula or with statistics. They need new methods like simulations and observations, and the project team has to be interdisciplinary. Designing computer-based complex socio-technical systems implies a *systemic design* approach. An approach that takes into account the identities of the components, their interactions and the system evolution along time. Computer professionals and scholars are learning from these new frontiers of science that it is fundamental to include human beings into the systems they are designing: it is the well-known proposal of "participatory design".

In this view, HCI is important for understanding the complex relationship between people and technology and how it can help computer professionals, engineers, interaction designers, information architects, etc. in their daily work. Engineering becomes "cognitive engineering". In the work of Penelope M. Sanderson, cognitive engineering is defined as the analysis, modeling, design, and evaluation of complex socio-technical systems. The focus is shifted from human interaction with ICT to the human-interaction with the complex physical systems. This approach takes into account the whole human-environment system and engineers discover the emerging properties from the interactions of system components. It is an HCI based on a "system perspective" ("*the whole is more than the sum of the parts*"), the properties of the whole system cannot be inferred by the properties of its parts, they can be inferred by studying the relations and interactions between the parts.

Also, it is important to underline the concept of "open system", that is one that is subject to disturbances that require intervention (control) in order to continue functioning according to some criterion (cybernetics). The strong recommendation is to build "ecological information systems" that human beings can operate easily as in the natural world (Sanderson, 2003)

Slow Tech as a compass for computer engineers

The most challenging part of the methodology is the fourth and final task: to make some proposals. As PhD students they need to develop an autonomous professional capability in formulating positive proposals. This is the main reason for introducing in this part of the computer ethics course for engineers a so-called *proactive computer ethics* based on the concept of Slow Tech: "*a good, clean, and fair ICT*". According to Slow Tech approach, the complex socio-technical systems they are going to design and develop should be designed by bearing in mind the three basic elements of Slow Tech.

This can be done by taking into account, first, human limits (*good ICT*). ICT can be good for human beings when the systems are designed using a human-centered approach, by accepting the complex interactions of the human being and ICT. The user experiences can be enjoyable only if the system and the HCI are designed taking into account human

limits. The skills and competences needed are: HCI, Design-for-All, e-Inclusion and methods such as Participatory Design.

Second, ICT should be *clean*: high tech generates toxic hazards throughout its entire lifecycle (including design, production, consumption and disposal). Clean ICT considers the environmental impact (the materials involved, chip manufacturing, power consumption of data centers and devices, ICT applications, e-waste management and recycling). Engineers start investigating the long-term sustainability of ICT, since its exponential growth risks to be unsustainable. The growing importance of new paradigms based on cycles, like the "circular economy" maybe of fundamental importance.

Third, ICT should be *fair*: it should be taken into account the working conditions of workers throughout the entire, globally extended ICT supply chain. The low cost features of ICT are possible at the price of the increasing costs paid by workers. The entire ICT Value-Chain is indeed quite long and complex and, in many parts, definitely not transparent.

For navigating the new territory of complex socio-technical systems, engineers need a *compass*, and Slow Tech can be useful to that purpose.

Preliminary results

The computer ethics course has been experimented since 2008 and its importance is demonstrated by the success among students: the number of PhD students that register for computer ethics is growing from about fifty in 2008 to more than one hundred in 2016 (the course is not mandatory).

The methodology is now well experimented and the main interesting point is the feedback from students. Since they have to prepare a report on the case analyzed based on the four steps, these are the main comments for each one:

Feedback from students

1. SCENARIO: "*difficult only at first attempt*"
2. STAKEHOLDERS' NETWORK "*very useful*"
3. ETHICAL ISSUES "*quite complex but affordable*"
4. ALTERNATIVE SCENARIO(s) "*the most challenging one*".

Based on these results, the last year the course has been reinforced in the last step of the methodology for providing the Slow Tech "*compass*" to students in order to facilitate their identification of possible alternative scenarios.

These are the main results of this experiment, where for each dimension of Slow Tech are listed the most important characteristics identified by students:

GOOD ICT

- Data protection / location
- Design-for-All
- HCI design
- Participatory design

- Precautionary principle
- Privacy-by-design
- Responsible research and innovation (RRI)
- Social desirability
- *Otium-negotium* balance (an ICT that helps in managing working time - free time)
- Web accessibility.

CLEAN ICT

- Supply-chain certification
- Environmental sustainability
- Power consumption minimization
- Modular design
- Power sources from renewable energies
- Transparency about power sources
- Recyclable-by-design
- Repairability
- Source material minimization.

FAIR ICT

- Fair working conditions
- Fair working conditions extended to all suppliers
- Compliance to Corporate Social Responsibility standards
- Location of suppliers
- Transparency of supply-chain
- Open data
- Open hardware
- Open software.

Conclusion

Teaching computer ethics to engineers is a very interesting challenge that deserves more work and research. In particular the need of "*making a difference*", that is, to prepare future generations of engineers to the design and development of *complex socio-technical systems* justifies further studies in fine tuning the four-steps methodology presented but also for preparing a more precise "compass" for navigating these new paradigms in computing such as cloud computing, big data, Internet of things, autonomous robots, machine learning, quantum computing.

References

Bynum, T. W. (2009). Ethical Decision Making and Case Analysis in Computer Ethics. In T. W. Bynum & S. Rogerson (Eds.), *Computer ethics and professional responsibility*. New Jersey, United States: Blackwell.

Bynum, T. W., Maner, W., & Fodor, J. L. (1991). Teaching Computer Ethics. Presented at the National Conference on Computing and Values (NCCV), Southern Connecticut State University.

Friedman, B., & Winograd, T. (1990). *Computing and social responsibility: a collection of course syllabi*. Palo Alto, CA, USA: Computer Professionals for Social Responsibility.

Gartner. (2017). Gartner Worldwide IT Spending Forecast Q4 2016. Retrieved January 20, 2017, from <http://www.gartner.com/technology/research/it-spending-forecast/>

Gotterbarn, D. (1991). Computer ethics: Responsibility regained. *National Forum: The Phi Beta Kappa Journal*, 71(3), 26–31.

International Professional Practice Partnership

Johnson, D. G. (1985). *Computer ethics*. Upper Saddle River, NJ: Prentice Hall.

Johnson, D. G., & Miller, K. (2009). *Computer ethics: analyzing information technology* (4th ed). Upper Saddle River, N.J: Prentice Hall.

Kavathatzopoulos, I. (2012). Assessing and Acquiring Ethical Leadership Competence. In G. P. Prastacos, F. Wang, & K. E. Soderquist (Eds.), *Leadership through the Classics* (pp. 389–400). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from http://link.springer.com/10.1007/978-3-642-32445-1_26

Kavathatzopoulos, I., & Laaksoharju, M. (2010). Computer aided ethical IT systems design. In M. Arias-Oliva, T. W. Bynum, & T. Torres-Coronas (Eds.), *ETHICOMP 2010: The “backwards, forwards, and sideways” changes of ICT* (pp. 332–340). Tarragona, Spain: Universitat Rovira i Virgili.

Lessig, L. (2002). *Code: and other laws of cyberspace*. New York: The Perseus Books Group.

Maner, W. (1980). *Starter Kit in Computer Ethics*. Switzerland: Helvetia Press.

Moor, J. H. (1985). What is Computer Ethics? *Metaphilosophy*, 16(4), 266–275.

Patrignani, N., & Whitehouse, D. (2014). Slow Tech: A Quest for Good, Clean and Fair ICT. *Journal of Information, Communication and Ethics in Society*, 12(2).

Rogerson, S. (n.d.). The Ethics of Software Development Project Management. In T. W. Bynum & S. Rogerson (Eds.), *Computer ethics and professional responsibility*. New Jersey, United States: Blackwell.

Spiekermann, S. (2016). *Ethical IT innovation: a value-based system design approach*. Florida: Auerbach Publications.

Tucker, A. B. (Ed.). (1991). Computing Curricula 1991. *Commun. ACM*, 34(6), 68–84. <https://doi.org/10.1145/103701.103710>

Wiener, N. (1954). *The human use of human beings: cybernetics and society* (2nd ed.). New York: Houghton Mifflin and Doubleday.