Fundamental Ethics in Information Systems

Christopher N. Chapman
Microsoft Corporation
chris.chapman@microsoft.com

Paper presented at the Hawaii International Conference on System Sciences
January 2006, Poipu, Kauai, HI

Abstract

Information systems often present virtual spaces that are sufficient to enable important human interaction. By enabling such interaction, systems designers are inherently creating certain ethical structures. When one creates an information system, one also creates the ethics for a new world of interaction and such ethics needs specific attention. I outline the basic elements of ethical structures: a framework for interpersonal interaction, personal identity, and structural conditions for customs and rules. I then examine philosophical methods for examining such structures and give a framework for thinking about them in IT systems. Finally, I propose a method to apply ethical design within traditional system development lifecycle models. Applying an ethical framework to IT systems provides more complete conceptual models of systems. Instead of arguing for specific prescriptive rules, I wish to help systems designers understand how they create ethical structures and can do so more with more deliberation.

1. Introduction

Advances in information technology (IT) have presented society with a wide range of ethical issues. Technologists, academicians, philosophers, politicians, and even the general public have paid significant attention to problems with information privacy, ownership and intellectual property, network access and security, fraud and other criminal activity, obscenity, access to sensitive technical information such as plans to make explosive devices, and various issues in professional ethics such as responsibility for flawed technical systems. Those areas of inquiry are valuable and increasingly provide useful guidelines for specific IT activities, such as the HIPAA (Healthcare Information Privacy and Accountability Act) IT standards.

We may draw a distinction between applied ethics and fundamental ethics. Applied ethics, which may also be called "morals," investigates specific rules for behavior, such as issues and standards for preserving privacy. This is the kind of analysis that we see in most discussions of IT ethics.

Rather than investigating specific rules or practices, fundamental ethics is concerned with the principles and premises that underlie such systems. For instance, fundamental ethics might investigate what kinds of entities are concerned in ethical systems, and how it is that moral rules do or do not apply to them. Despite the breadth of IT-related activities in applied ethics, surprisingly little attention has been paid to issues of fundamental ethics for IT systems.

In this paper, I propose two things: first, that systems designers often engage in fundamental ethics, perhaps unknowingly; and second, that a constructivist approach to ethics is a valuable framework for understanding such activity. On that basis, I then discuss how to apply such ethical thinking in IT systems design. I do not present moral arguments, such as whether privacy is a moral good, or some proposed way in which privacy should be implemented in systems. Instead, I present an ethical framework in which IT system designers may consider how to address such issues as an integral part of systems development.

2. Information systems design as fundamental ethics

For purposes here, I assume the following: interpersonal interaction mediated by at least some information systems provides the essential bases for behavior that may be characterized as moral, that is, as right or wrong. This is a complex issue, the complete justification of which is outside the scope of this paper. However, it has not been persuasively criticized and the alternative assumption (that all interpersonal interaction in all information systems is

necessarily amoral) requires commitments to positions (such as anti-realism or amoralism) that are not self-evident.

In support of my assumption, Powers [5] argues from a realist stance that acts in cyberspace have real effects and thus are subject to moral evaluation. I have previously argued from a phenomenological stance that some software systems provide sufficient conditions for moral behavior and thus establish ethical systems [1].

There are several features of at least some IT systems that jointly establish them as ethical systems (cf. [5] and [1]):

- People participate in the systems
- Those people interact with one another
- Such interaction is mediated by rules or structures provided by the systems
- The interaction is of a quality sufficient to allow real personal benefit or harm
- People using the system can establish expectations around such behavior
- Those behavioral expectations may be represented and understood as behavioral principles shared among people, i.e., they can be morals

IT systems designers create the spaces in which such interaction takes place. They are creating the conditions that define the scope and nature of possible moral interaction. Thus, in defining such systems, IT designers are not only engaging in implementation of various moral principles (e.g., the privacy of their users), they are also engaging in fundamental ethics (establishing conditions for the moral behavior of the users within the systems).

Examples of systems where such interaction is possible include some email and messaging systems, online discussions, ecommerce sites and auction communities, multiplayer games, various kinds of networks, and even business applications such as sophisticated operations and accounting systems. (See [1] for a longer discussion of these issues.)

3. Constructivist ethics

Many philosophical frameworks exist for understanding (or sometimes denying) ethical behavior. Unfortunately, most of those frameworks are of little direct value to systems designers whose role is not to understand or justify but to create such systems. Among the useful frameworks, I propose that a variety of moral constructivism based on Kant and Rawls is most directly amenable to IT systems design.

Kant [4] argued for a conception of ethics based on three elements: a distinction between transcendental (abstract) principles for behavioral rules and specific implementations of such rules; the person as the constructor of rules that comply with the principles; and the consistency of considering such principles to be universal across persons. Kant arrived at the concept of "duty" as the highest value, arguing that we should formulate moral rules such that we do what is right just because it is right, because any other principle would lead to inconsistency if applied universally. The key point for us, however, is that this outlines a framework in which specific behavioral standards are explicitly constructed by rational actors in accordance with general and universally applicable principles.

John Rawls [7] adapted this constructivist framework to address procedures to resolve questions of fairness and justice in societies based on social contract. In particular, Rawls [6][7] advocated that we consider moral rules from the perspective of the "original position." In the original position, people are imagined to be placed in charge of constructing the rules for a society, while being behind a "veil of ignorance." The veil of ignorance means that they do not know what place they will occupy within the society.

Rawls asserts that in such a situation, people in modern Western society would arrive at a social contract in which freedom and other basic rights are not arbitrarily restricted (e.g., by race or gender) and gross imbalances in distribution of social goods are avoided. Because people would not know the place they would occupy outside the veil of ignorance, they would wish to have a system that would be maximally fair to all (while allowing some degree of inequality due to personal choices and so forth). The important insight from Rawls is that there can be specific procedures (e.g., the "veil of ignorance") abstracted from particular situations that people may follow to construct ethical systems.

Neither Kant's nor Rawls's framework is sufficient for fundamental ethics in IT systems. In particular, there are two shortcomings. First, they are dependent upon a specific conception of person. Kant adopted a particular kind of rational agent, and Rawls further narrowed this to people who are mutually present in some kind of established society similar to current Western cultures. However, these conditions are not established in some kinds of IT systems. What it means to be an actor in a system, how presence occurs, and the scope of shared online or offline cultural background are choices made by the designer. For instance, an online worldwide multiplayer game would have very different conceptions of the "person"

and "society" than would an internal email system for a company in a fixed place.

Second, both Kant's and especially Rawls's frameworks assume that the persons interacting are ultimately responsible for the ethical nature of their society. People construct a certain kind of society and are responsible for maintaining it through their actions. In IT systems, on the other hand, this kind of responsibility accrues primarily, if not exclusively, to the system designer. It is possible to design a system in which people can construct their own rules and engage in maintaining them (for example, discussion rooms and some kinds of multiplayer games), but it is also possible to design systems in which such ability is curtailed or explicitly denied. Responsibility accrues to the system designer, and may or may not also involve the ultimate actors in the system.

Kant's and Rawls's frameworks show us how to think about ethical systems abstracted from specific situations, and outline some of the elements that must be considered. Once we add considerations of who the "persons" are, and who is responsible for constructing the grounds of ethical principles, we will have a complete framework for IT systems.

4. Framework for fundamental ethics in IT systems

We have seen that fundamental ethics has to answer questions about who people are, how they interact, and how the principles of interaction are formed. In Table 1, I combine all of the areas identified above, which must be addressed in thinking about the ethics of an IT system.

The interesting thing about the areas in Table 1 is that some of them fall well within traditional systems design, while others are rarely, if ever, considered. In particular, questions 1, 2, 3, 4, and 9 are basic questions for any system design: who will use it and how they will interact. Questions 5 and 6 are sometimes addressed, especially in critical reviews of technical systems. For instance, social critics have investigated how social interaction has changed as a result of technology adoption, and whether there are social inequities such as a "digital divide" between people who do and do not have access to technology.

The remaining questions (numbers 7, 8, 10, 11, 12, and 13) in Table 1 are precisely those that are central to fundamental ethics. They are almost never considered by systems designers. They should be; those issues and decisions form the groundwork for the other questions that are commonly considered. Without a stance on how ethical principles should be decided, how can one decide or implement specific instances of ethical principles? For instance, if one has no stance on how to know whose duty it is to ensure fair access to technology, how could one appropriately be concerned with inequities in access?

The answer, of course, is that we all have preconceptions and assumptions about such things. For instance, a social activist might assume that such things are constructed in a particular way as a reflection of a specific corporate power dynamic, while a systems designer might believe that the system should be "neutral" and allow any such framework to "emerge" from user behavior.

Table 1. Questions for fundamental ethics

Participants and interaction

1. What does it mean to be a "participant"?
2. Who participates in the system, i.e., what is the "society" involved?
3. How do those people interact with one another?
4. What are the rules or structures provided by the system?
5. Does the interaction allow personal benefit or harm?

Ethical principles

6. What are the basic concerns of people in the system (the "goods" with which fairness is concerned)?
7. Could people using the system establish expectations around behavior with respect to those concerns?
8. Can those behavioral expectations be represented and understood as behavioral principles shared among people (as morals)?
9. What are the mechanisms, if any, for maintaining or enforcing the principles within the system?

Ethical construction

10. Is there a general form of those ethical principles (like "fairness")?
11. If so, how can the general ethical principles be implemented?
12. Who is responsible for establishing the principles?
13. Is there a general procedure for establishing the principles?

Such assumptions are inadequate. Many IT systems are sufficiently complex that the results of interpersonal interaction cannot be assumed, and IT systems are sufficiently different from ordinary life that our working assumptions from everyday interaction do not apply. To take one difference as a point: in everyday interaction, we assume that people

are in principle present to us. If someone hits, insults, or defrauds me, I have various means to identify him or her and follow up on the behavior. In IT systems, the ability to identify another, however, may be severely curtailed or even impossible; it can even be impossible to know whether any person is present or acting at all.

To overcome these assumptions, it is helpful not only to be aware of them, but also to have a method that addresses the neglected ethical questions. Such a method must address two aspects: it must be constructive as described above; and it must be transcendental, that is, separate from particular instances and amenable to generalization. For our purposes, it also must be applicable to IT systems design, preferably in a straightforward manner.

5. Outline of a method to address IT systems ethics

To address IT systems ethics with an explicitly transcendental and constructivist philosophical method, as we find in Kant or Rawls, would pose many challenges. Foremost among the challenges is that such a method depends upon a certain degree of training (such as graduate education in philosophy) that is not only laborious but also is likely to seem foreign and of questionable value to IT systems designers. Second is that such methods themselves were not developed with IT issues in mind; as we saw above, they make assumptions about people and societies that are not applicable to IT systems in general.

A more fruitful starting point is in IT design methods themselves. In particular, there are three IT engineering traditions that offer useful cognitive tools: traditional algorithmics, user-centered design, and security threat modeling. Algorithmics provides the transcendental skeleton for such thought. In algorithmics, designers learn to consider all the implications of their work, proof of its function (that is, its generalized correctness), and how to consider boundary cases that test the work. In such work, IT designers have already learned to think very much like philosophers do about abstract conditions, albeit with different application.

User-centered design can provide the constructivist flesh. Applying user-centered design principles provides a way to address people's behavioral expectations, and suggests ways to approach the explicit construction of ethical principles by involving users in such a process. User-centered design also supplies answers to some of the commonly considered questions in Table 1, such as a description of the people who will use the system.

Security threat modeling is a recent development in software engineering that attempts to form an overview of all possible areas that could be attacked in a system, the kinds of attacks that might be posed against those areas, and possible responses or design changes to thwart them (cf. [3]). Because of this focus on the overall system, security modeling complements algorithmic approaches in thinking about the boundary conditions and overall interaction of a system. The interesting thing for our purposes is that security modeling can be rather easily transposed to the ethical realm. Instead of a system's "attack surface" against which threats are directed, we could think of an "ethical surface" where ethical behaviors occur; instead of attacks, we can think of ethical transgressions; and so forth.

6. Applying the ethical method in IT systems development

This collection of tools might function together as follows. (At this point, I am proposing an untested method; to my knowledge, such an approach has not been attempted in its entirety by any researchers or designers.) First, one would outline the system and basic interactions, following the appropriate principles of user-centered design. This establishes a first take on who will be interacting in the world and what they will do. Then one extends the design, and perhaps begins to build the system with traditional engineering techniques (coding, interface design, database design, etc.). At this point, one begins to think about algorithmic issues, but instead of merely looking at the functional correctness of the system, one should extend to consider the consistency of the interaction between people. This would involve thinking about the boundary conditions for interaction, and what kinds of things can go wrong. As this occurs, it is possible to apply principles borrowed from security modeling to start to outline the ethical surface of the system. Where do people interact in ways that have ethical assumptions? What are the potential areas for problematic interaction? What can we do to mitigate those?

At this point, there is a relatively complete draft of the system and its intended interactions (both functional and ethical). This becomes a good point for user-centered feedback on all elements. Traditionally, users would give feedback on the features (e.g., in market requirements research) and interaction of a system (e.g., in usability testing). I propose to add a third component: feedback on the

ethical structure. This could occur in two ways. First, it could be given directly, with users commenting on the identified assumptions, problem areas, and mitigation. In that sense, it would be similar to feedback on any other kind of interaction design.

Second, one might add a specifically constructive exercise, similar to Rawls's procedure for deciding a social contract. Present users with the goals of the system and outline the structure in which users may interact. Then instruct them to adopt an attitude similar to Rawls's original position: to think about the system on behalf of all users, apart from how they might personally interact with it. Then ask them to perform two tasks, similar to traditional user design reviews or participatory design (cf. Constantine & Lockwood [2], pp. 391-415, 483-484). The first task is to decide on the general principles that should underlie interaction of the system; ask them to create the rules for such interaction. Second, ask them to apply those rules and decide on types of mitigation that make sense within the system.

The goal of these user exercises in commentary and ethical participatory design is not to design the specific ethical structure of the system. Rather, such exercises serve to test the prior ethical modeling performed by the designers. Do users arrive at the same conclusions as the designers? Do they seem to understand the ethical framework in the way it was intended? Do they identify areas that the designers did not consider?

Finally, as in development of any sophisticated system, one iterates on these steps. Ethical modeling and user feedback contribute to design changes, and then those design changes are subject to further analysis, user commentary, and revision.

7. Conclusion

I have argued that IT systems are rarely subjected to consideration of their most basic ethical foundations. This is likely due to inattention on the part of philosophers, and varying degrees of trepidation, laissez-faire attitudes, and oversight on the part of systems designers. The philosophical literature that bears on fundamental ethics may be difficult or foreign to IT designers, but it appears quite possible that methods that are common and well-understood within the IT community can be modified and extended to address ethical issues. If so, there is great opportunity to inform the IT community. Increased attention to the fundamentally ethical components of systems design may help us build IT systems that are more completely designed, more predictable to users, and safer and more enjoyable for society.

8. References

[1] Chapman, C.N., "Designing software ethics", Paper presented at Society for Philosophy in the Contemporary World 9th Annual Conference, Santa Fe, NM, July 2002.

[2] Constantine, L.L. and Lockwood, L.A.D., *Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design*, ACM Press, New York, NY, 1999.

[3] Howard, M., and LeBlanc, D.C., *Writing Secure Code*, 2nd ed., Microsoft Press, Redmond, WA, 2002.

[4] Kant, I., *Groundwork for the Metaphysics of Morals* (Trans. and ed., A. W. Wood), Yale University Press, New Haven, CT, 2002 (1785) (Alternate translation available online at: http://etext.library.adelaide.edu.au/k/kant/immanuel/k16prm/)

[5] Powers, T.M., "Real wrongs in virtual communities", *Ethics and Information Technology*, 2003, vol. 5, 191-198.

[6] Rawls, J., *A Theory of Justice*, Harvard University Press, Cambridge, MA, 1971.

[7] Rawls, J., "Kantian constructivism in moral theory", *Journal of Philosophy*, 1980, vol. 77, 515-72.