THE CAMBRIDGE HANDBOOK OF INFORMATION AND COMPUTER ETHICS

Edited by Luciano Floridi

CAMBRIDGE

3 Values in technology and disclosive computer ethics

Philip Brey

3.1 Introduction

Is it possible to do an ethical study of computer systems themselves independently of their use by human beings? The theories and approaches in this chapter answer this question affirmatively and hold that such studies should have an important role in computer and information ethics. In doing so, they undermine conventional wisdom that computer ethics, and ethics generally, is concerned solely with human conduct, and they open up new directions for computer ethics, as well as for the design of computer systems.

As our starting point for this chapter, let us consider some typical examples of ethical questions that are raised in relation to computers and information technology, such as can be found throughout this book:

- Is it wrong for a system operator to disclose the content of employee email messages to employers or other third parties?
- Should individuals have the freedom to post discriminatory, degrading and defamatory messages on the Internet?
- Is it wrong for companies to use data-mining techniques to generate consumer profiles based on purchasing behaviour, and should they be allowed to do so?
- Should governments design policies to overcome the digital divide between skilled and unskilled computer users?

As these examples show, ethical questions regarding information and communication technology typically focus on the morality of particular ways of *using* the technology or the morally right way to *regulate* such uses.

Taken for granted in such questions, however, are the computer systems and software that are used. Could there, however, not also be valid ethical questions that concern the technology itself? Could there be an ethics of computer systems separate from the ethics of *using* computer systems? The *embedded values* approach in computer ethics, formulated initially by Helen Nissenbaum (1998; Flanagan, Howe and Nissenbaum 2008) and since adopted by many authors in the field, answers these questions affirmatively, and aims to develop a theory and methodology for moral reflection on computer systems themselves, independently of particular ways of using them.

The embedded values approach holds that computer systems and software are not morally neutral and that it is possible to identify tendencies in them to promote or demote particular moral values and norms. It holds, for example, that computer programs can be supportive of privacy, freedom of information, or property rights or, instead, to go against the realization of these values. Such tendencies in computer systems are called ‘embedded’, ‘embodied’ or ‘built-in’ moral values or norms. They are built-in in the sense that they can be identified and studied largely or wholly independently of actual uses of the system, although they manifest themselves in a variety of uses of the system. The embedded values approach aims to identify such tendencies and to morally evaluate them. By claiming that computer systems may incorporate and manifest values, the embedded values approach is not claiming that computer systems engage in moral actions, that they are morally praiseworthy or blameworthy, or that they bear moral responsibility (Johnson 2006). It is claiming, however, that the design and operation of computer systems has moral consequences and therefore should be subjected to ethical analysis.

If the embedded values approach is right, then the scope of computer ethics is broadened considerably. Computer ethics should not just study ethical issues in the use of computer technology, but also in the technology itself. And if computer systems and software are indeed value-laden, then many new ethical issues emerge for their design. Moreover, it suggests that design practices and methodologies, particularly those in information systems design and software engineering, can be changed to include the consideration of embedded values.

In the following section, Section 3.2, the case will be made for the embedded values approach, and some common objections against it will be discussed. Section 3.3 will then turn to an exposition of a particular approach in computer ethics that incorporates the embedded values approach, *disclosive computer ethics*, proposed by the author (Brey 2000). Disclosive computer ethics is an attempt to incorporate the notion of embedded values into a comprehensive approach to computer ethics. Section 3.4 considers *value-sensitive design* (VSD), an approach to design developed by computer scientist Batya Friedman and her associates, which incorporates notions of the embedded values approach (Friedman, Kahn and Borning 2006). The VSD approach is not an approach within ethics but within computer science, specifically within information systems design and software engineering. It aims to account for values in a comprehensive manner in the design process, and makes use of insights of the embedded values approach for this purpose. In a concluding section, the state of the art in these different approaches is evaluated and some suggestions are made for future research.

3.2 How technology embodies values

The existing literature on embedded values in computer technology is still young, and has perhaps focused more on case studies and applications for design than on theoretical underpinnings. The idea that technology embodies values has been inspired by work in the interdisciplinary field of science and technology studies, which investigates the development of science and technology and their interaction with society. Authors in this field agree that technology is not neutral but shaped by society. Some have argued, specifically, that technological artefacts (products or systems) issue constraints on the world surrounding them (Latour 1992) and that they can harbour political consequences (Wiener 1954). Authors in the embedded value approach have taken these ideas and applied them to ethics, arguing that technological artefacts are not morally neutral but value-laden. However, what it means for an artefact to have an embedded value remains somewhat vague.

In this section a more precise description of what it means for a technological artefact to have embedded values is articulated and defended. The position taken here is in line with existing accounts of embedded values, although their authors need not agree with all of the claims made in this section. The idea of embedded values is best understood as a claim that technological artefacts (and in particular computer systems and software) have built-in tendencies to promote or demote the realization of particular values. Defined in this way, a built-in value is a special sort of built-in consequence. In this section a defence of the thesis that technological artefacts are capable of having built-in consequences is first discussed. Then tendencies for the promotion of values are identified as special kinds of built-in consequences of technological artefacts. The section is concluded by a brief review of the literature on values in information technology, and a discussion of how values come to be embedded in technology.

3.2.1 Consequences built into technology

The embedded values approach promotes the idea that technology can have built-in tendencies to promote or demote particular values. This idea, however, runs counter to a frequently held belief about technology, the idea that technology itself is neutral with respect to consequences. Let us call this the *neutrality thesis*. The neutrality thesis holds that there are no consequences that are inherent to technological artefacts, but rather that artefacts can always be used in a variety of different ways, and that each of these uses comes with its own consequences. For example, a hammer can be used to hammer nails, but also to break objects, to kill someone, to flatten dough, to keep a pile of paper in place or to conduct electricity. These uses have radically different effects on the world, and it is difficult to point to any single effect that is constant in all of them.

The hammer example, and other examples like it (a similar example could be given for a laptop), suggest strongly that the neutrality thesis is true. If so, this would have important consequences for an ethics of technology. It would follow that ethics should not pay much attention to technological artefacts themselves, because they in themselves do not 'do' anything. Rather, ethics should focus on their usage alone.

This conclusion holds only if one assumes that the notion of embedded values requires that there are consequences that manifest themselves in each and every use of an artefact. But this strong claim need not be made. A weaker claim is that artefacts may have built-in consequences in that there are recurring consequences that manifest themselves in a wide range of uses of the artefact, though not in all uses. If such recurring consequences can be associated with technological artefacts, this may be sufficient to falsify the strong claim of the neutrality thesis that each use of a technological artefact comes with its own consequences. And a good case can be made that at least some artefacts can be associated with such recurring consequences.

An ordinary gas-engine automobile, for example, can evidently be used in many different ways: for commuter traffic, for leisure driving, to taxi passengers or cargo, for hit jobs, for auto racing, but also as a museum piece, as a temporary shelter for the rain or as a barricade. Whereas there is no single consequence that results from all of these uses, there are several consequences that result from a large number of these uses: in all but the last three uses, gasoline is used up, greenhouse gases and other pollutants are being released, noise is being generated, and at least one person (the driver) is being moved around at high speeds. These uses, moreover, have something in common: they are all *central* uses of automobiles, in that they are accepted uses that are frequent in society and that account for the continued production and usage of automobiles. The other three uses are *peripheral* in that they are less dominant uses that depend for their continued existence on these central uses, because their central uses account for the continued production and consumption of automobiles. Central uses of the automobile make use of its capacity for driving, and when it is used in this capacity, certain consequences are very likely to occur. Generalizing from this example, a case can be made that technological artefacts are capable of having built-in consequences in the sense that *particular consequences may manifest themselves in all of the central uses of the artefact.*

It may be objected that, even with this restriction, the idea of built-in consequences employs a too deterministic conception of technology. It suggests that, when technological artefacts are used, particular consequences are necessary or unavoidable. In reality, there are usually ways to avoid particular consequences. For example, a gas-fuelled automobile need not emit greenhouse gases into the atmosphere if a greenbox device is attached to it, which captures carbon dioxide and nitrous oxide and converts it into bio-oil. To avoid this objection, it may be claimed that the notion of built-in consequences does not refer to necessary, unavoidable consequences but rather to strong *tendencies* towards certain consequences. The claim is that these consequences are normally realized whenever the technology is used, unless it is used in a context that is highly unusual or if extraordinary steps are taken to avoid particular consequences. Built-in consequences are therefore never absolute but always relative to a set of typical uses and contexts of use, outside of which the consequences may not occur.

Do many artefacts have built-in consequences in the way defined above? The extent to which technological artefacts have built-in consequences can be correlated with two factors: the extent to which they are capable of exerting force or behaviour autonomously, and the extent to which they are embedded in a fixed context of use. As for the first parameter, some artefacts seem to depend strongly on users for their consequences, whereas others seem to be able to generate effects on their own. Mechanical and electrical devices, in particular, are capable of displaying all kinds of behaviours on their own, ranging from simple processes, like the consumption of fuel or the emission of steam, to complex actions, like those of robots and artificial agents. Elements of infrastructure, like buildings, bridges, canals and railway tracks, may not behave autonomously but, by their mere presence, they do impose significant constraints on their environment, including the actions and movements of people, and in this way engender their own consequences. Artefacts that are not mechanical, electrical or infrastructural, like simple hand-held tools and utensils, tend to have less consequences of their own and their consequences tend to be more dependent on the uses to which they are put.

As for the second parameter, it is easier to attribute built-in consequences to technological artefacts that are placed in a fixed context of use than to those that are used in many different contexts. Adapting an example by Winner (1980), an overpass that is 180 cm (6 ft) high has as a generic built-in consequence that it prevents traffic from going through that is more than 180 cm high. But when such an overpass is built over the main access road to an island from a city in which automobiles are generally less than 180 cm high and buses are taller, then it acquires a more specific built-in consequence, which is that buses are being prevented from going to the island whereas automobiles do have access. When, in addition, it is the case that buses are the primary means of transportation for black citizens, whereas most white citizens own automobiles, then the more specific consequence of the overpass is that it allows easy access to the island for one racial group, while denying it to another. When the context of use of an artefact is relatively fixed, the immediate, physical consequences associated with a technology can often be *translated* into social consequences because there are reliable correlations between the physical and the social (for example between prevention of access to buses and prevention of access to blacks) that are present (Latour 1992).

3.2.2 From consequences to values

Let us now turn from built-in consequences to embedded values. An embedded value is a special kind of built-in consequence. It has already been explained how technological artefacts can have built-in consequences. What needs to be explained now is how some of these built-in consequences can be associated with values. To be able to make this case, let us first consider what a value is.

Although the notion of a value remains somewhat ambiguous in philosophy, some agreements seem to have emerged (Frankena 1973). First, philosophers tend to agree that values depend on *valuation*. Valuation is the act of valuing something, or finding it valuable, and to find something valuable is to find it *good* in some way. People find all kinds of things valuable, both abstract and concrete, real and unreal, general and specific. Those things that people find valuable that are both ideal and general, like justice and generosity, are called *values*, with *disvalues* being those general qualities that are considered to be bad or evil, like injustice and avarice. Values, then, correspond to idealized qualities or conditions in the world that people find good. For example, the value of justice corresponds to some idealized, general condition of the world in which all persons are treated fairly and rewarded rightly.

To have a value is to want it to be *realized*. A value is realized if the ideal conditions defined by it are matched by conditions in the actual world. For example, the value of freedom is fully realized if everyone in the world is completely free. Often, though, a full realization of the ideal conditions expressed in a value is not possible. It may not be possible for everyone to be completely free, as there are always at least some constraints and limitations that keep people from a state of complete freedom. Therefore, values can generally be realized only to a degree.

The use of a technological artefact may result in the partial realization of a value. For instance, the use of software that has been designed not to make one's personal information accessible to others helps to realize the value of privacy. The use of an artefact may also hinder the realization of a value or promote the realization of a disvalue. For instance, the use of software that contains spyware or otherwise leaks personal data to third parties harms the realization of the value of privacy. Technological artefacts are hence capable of either *promoting* or *harming* the realization of values when they are used. When this occurs systematically, in all of its central uses, we may say that the artefact embodies a special kind of built-in consequence, which is a *built-in tendency to promote or harm the realization of a value*. Such a built-in tendency may be called, in short, an *embedded value* or *disvalue*. For example, spyware-laden software has a tendency to harm privacy in all of its typical uses, and may therefore be claimed to have harm to privacy as an embedded disvalue.

Embedded values approaches often focus on *moral values*. Moral values are ideals about how people ought to behave in relation to others and themselves and how society should be organized so as to promote the right course of action. Examples of moral values are justice, freedom, privacy and honesty. Next to moral values, there are different kinds of non-moral values, for example, aesthetic, economic, (non-moral) social and personal values, such as beauty, efficiency, social harmony and friendliness.

Values should be distinguished from norms, which can also be embedded in technology. *Norms* are rules that prescribe which kinds of actions or state of affairs are forbidden, obligatory or allowed. They are often based on values that provide a rationale for them. *Moral norms* prescribe which actions are forbidden, obligatory or allowed from the point of view of morality. Examples of moral norms are ‘do not steal’ and ‘personal information should not be provided to third parties unless the bearer has consented to such distribution’. Examples of non-moral norms are ‘pedestrians should walk on the right side of the street’ and ‘fish products should not contain more than 10 mg histamines per 100 grams’. Just as technological artefacts can promote the realization of values, they can also promote the enforcement of norms. *Embedded norms* are a special kind of built-in consequence. They are tendencies to effectuate norms by bringing it about that the environment behaves or is organized according to the norm. For example, web browsers can be set not to accept cookies from websites, thereby enforcing the norm that websites should not collect information about their user. By enforcing a norm, artefacts thereby also promote the corresponding value, if any (e.g., privacy in the example).

So far we have seen that technological artefacts may have embedded values understood as special kinds of built-in consequences. Because this conception relates values to causal capacities of artefacts to affect their environment, it may be called the *causalist* conception of embedded values. In the literature on embedded values, other conceptions have been presented as well. Notably, Flanagan, Howe and Nissenbaum (2008) and Johnson (1997) discuss what they call an *expressive* conception of embedded values. Artefacts may be said to be expressive of values in that they incorporate or contain symbolic meanings that refer to values. For example, a particular brand of computer may symbolize or represent status and success, or the representation of characters and events in a computer game may reveal racial prejudices or patriarchal values. Expressive embedded values in artefacts *represent* the values of designers or users of the artefact. This does not imply, however, that they also function to *realize* these values. It is conceivable that the values expressed in artefacts cause people to adopt these values and thereby contribute to their own realization. Whether this happens frequently remains an open question. In any case, whereas the expressive conception of embedded values merits further philosophical reflection, the remainder of this chapter will be focused on the causalist conception.

3.2.3 Values in information technology

The embedded values approach within computer ethics studies embedded values in computer systems and software and their emergence, and provides moral evaluations of them. The study of embedded values in Information and Communication Technology (ICT) has begun with a seminal paper by Batya Friedman and Helen Nissenbaum in which they consider *bias* in computer systems (Friedman and Nissenbaum 1996). A biased computer system or program is defined by them as one that systematically and unfairly discriminates against certain individuals or groups, who may be users or other stakeholders of the system. Examples include educational programs that have much more appeal to boys than to girls, loan approval software that gives negative recommendations for loans to individuals with ethnic surnames, and databases for matching organ donors with potential transplant recipients that systematically favour individuals retrieved and displayed immediately on the first screen over individuals displayed on later screens. Building on their work, I have distinguished *user biases* that discriminate against (groups of) users of an information system, and *information biases* that discriminate against stakeholders represented by the system (Brey 1998). I have discussed various kinds of user bias, such as user exclusion and the selective penalization of users, as well as different kinds of information bias, including bias in information content, data selection, categorization, search and matching algorithms and the display of information.

After their study of bias in computer systems, Friedman and Nissenbaum went on to consider consequences of software agents for the autonomy of users. Software agents are small programs that act on behalf of the user to perform tasks. Friedman and Nissenbaum (1987) argue that software agents can undermine user autonomy in various ways – for example by having only limited capabilities to perform wanted tasks or by not making relevant information available to the user – and argue that it is important that software agents are designed so as to enhance user autonomy. The issue of user autonomy is also taken up in Brey (1998, 1999c), in which I argue that computer systems can undermine autonomy by supporting monitoring by third parties, by imposing their own operational logic on the user, thus limiting creativity and choice, or by making users dependent on systems operators or others for maintenance or access to systems functions.

Deborah Johnson (1997) considers the claim that the Internet is an inherently democratic technology. Some have claimed that the Internet, because of its distributed and nonhierarchical nature, promotes democratic processes by empowering individuals and stimulating democratic dialogue and decision-making (see Chapter 10). Johnson subscribes to this democratic potential. She cautions, however, that these democratic tendencies may be limited if the Internet is subjected to filtering systems that only give a small group of individuals control over the flow of information on the Internet. She hence identifies both democratic and undemocratic tendencies in the technology that may become dominant depending on future use and development.

Other studies, within the embedded values approach, have focused on specific values, such as privacy, trust, community, moral accountability and informed consent, or on specific technologies. Introna and Nissenbaum (2000) consider biases in the algorithms of search engines, which, they argue, favour websites with a popular and broad subject matter over specialized sites, and the powerful over the less powerful. Introna (2007) argues that existing plagiarism detection software creates an artificial distinction between alleged plagiarists and non-plagiarists, which is unfair. Introna (2005) considers values embedded in facial recognition systems. Camp (1999) analyses the implications of Internet protocols for democracy. Flanagan, Howe and Nissenbaum (2005) study values in computer games, and Brey (1999b, 2008) studies them in computer games, computer simulations and virtual reality applications. Agre and Mailloux (1997) reveal the implications for privacy of Intelligent Vehicle-Highway Systems, Tavani (1999) analyses the implications of data-mining techniques for privacy and Fleischmann (2007) considers values embedded in digital libraries.

3.2.4 The emergence of values in information technology

What has not been discussed so far is how technological artefacts and systems acquire embedded values. This issue has been ably taken up by Friedman and Nissenbaum (1996). They analyse the different ways in which biases (injustices) can emerge in computer systems. Although their focus is on biases, their analysis can easily be generalized to values in general. Biases, they argue, can have three different types of origins. *Preexisting biases* arise from values and attitudes that exist prior to the design of a system. They can either be *individual*, resulting from the values of those who have a significant input into the design of the systems, or *societal*, resulting from organizations, institutions or the general culture that constitute the context in which the system is developed. Examples are racial biases of designers that become embedded in loan approval software, and overall gender biases in society that lead to the development of computer games that are more appealing to boys than to girls. Friedman and Nissenbaum note that preexisting biases can be embedded in systems intentionally, through conscious efforts of individuals or institutions, or unintentionally and unconsciously.

A second type is *technical bias*, which arises from technical constraints or considerations. The design of computer systems includes all kinds of technical limitations and assumptions that are perhaps not value-laden in themselves but that could result in value-laden designs, for example because limited screen sizes cannot display all results of a search process, thereby privileging those results that are displayed first, or because computer algorithms or models contain formalized, simplified representations of reality that introduce biases or limit the autonomy of users, or because software engineering techniques do not allow for adequate security, leading to systematic breaches of privacy. A third and final type is *emergent bias*, which arises when the social context in which the system is used is not the one intended by its designers. In the new context, the system may not adequately support the capabilities, values or interests of some user groups or the interests of other stakeholders. For example, an ATM that relies heavily on written instructions may be installed in a neighborhood with a predominantly illiterate population.

Friedman and Nissenbaum's classification can easily be extended to embedded values in general. Embedded values may hence be identified as preexisting, technical or emergent. What this classification shows is that embedded values are not necessarily a reflection of the values of designers. When they are, moreover, their embedding often has not been intentional. However, their embedding *can* be an intentional act. If designers are aware of the way in which values are embedded into artefacts, and if they can sufficiently anticipate future uses of an artefact and its future context(s) of use, then they are in a position to intentionally design artefacts to support particular values. Several approaches have been proposed in recent years that aim to make considerations of value part of the design process. In Section 3.4, the most influential of these approaches, called value-sensitive design, is discussed. But first, let us consider a more philosophical approach that also adopts the notion of embedded values.

3.3 Disclosive computer ethics

The approach of *disclosive computer ethics* (Brey 2000, 1999a) intends to make the embedded values approach part of a comprehensive approach to computer ethics. It is widely accepted that the aim of computer ethics is to morally evaluate *practices* that involve computer technology and to devise ethical policies for these practices. The practices in question are activities of designing, using and managing computer technology by individuals, groups or organizations. Some of these practices are already widely recognized in society as morally controversial. For example, it is widely recognized that copying patented software and filtering Internet information are morally controversial practices. Such practices may be called *morally transparent* because the practice is known and it is roughly understood what moral values are at stake in relation to it.

In other computer-related practices, the moral issues that are involved may not be sufficiently recognized. This may be the case because the practices themselves are not well known beyond a circle of specialists, or because they are well known but not recognized as morally charged because they have a false appearance of moral neutrality. Practices of this type may be called *morally opaque*, meaning that it is not generally understood that the practice raises ethical questions or what these questions may be. For example, the practice of browser tracking is morally opaque because it is not well known or well understood by many people, and the practice of search engine use is morally opaque because, although the practice is well known, it is not well known that the search algorithms involved in the practice contain biases and raise ethical questions.

Computer ethics has mostly focused on morally transparent practices, and specifically on practices of using computer systems. Such approaches may be called *mainstream computer ethics*. In mainstream computer ethics, a typical study begins by identifying a morally controversial practice, like software theft, hacking, electronic monitoring or Internet pornography. Next, the practice is described and analysed in descriptive terms, and finally, moral principles and judgements are applied to it and moral deliberation takes place, resulting in a moral evaluation of the practice and, possibly, a set of policy recommendations. As Jim Moor has summed up this approach, 'A typical problem in computer ethics arises because there is a policy vacuum about how computer technology should be used' (1985, p. 266).

The approach of *disclosive computer ethics* focuses instead on morally opaque practices. Many practices involving computer technology are morally opaque because they include operations of technological systems that are very complex and difficult to understand for laypersons and that are often hidden from view for the average user. Additionally, practices are often morally opaque because they involve distant actions over computer networks by system operators, providers, website owners and hackers and remain hidden from view from users and from the public at large. The aim of disclosive ethics is to identify such morally opaque practices, describe and analyse them, so as to bring them into view, and to identify and reflect on any problematic moral features in them. Although mainstream and disclosive computer ethics are different approaches, they are not rival approaches but are rather complementary. They are also not completely separable, because the moral opacity of practices is always a matter of degree, and because a complex practice may include both morally transparent and opaque dimensions, and thus require both approaches.

Many computer-related practices that are morally opaque are so because they depend on operations of computer systems that are value-laden without it being known. Many morally opaque practices, though not all, are the result of undisclosed embedded values and norms in computer technology. A large part of the work in disclosive computer ethics, therefore, focuses on the identification and moral evaluation of such embedded values.

3.3.1 Methodology: multi-disciplinary and multi-level

Research typically focuses on an (alleged) morally opaque practice (e.g., plagiarism detection) and optionally on a morally opaque computer system or software program involved in this practice (e.g., plagiarism detection software). The aim of the investigation usually is to reveal hidden morally problematic features in the practice and to provide ethical reflections on these features, optionally resulting in specific moral judgements or policy recommendations. To achieve this aim, research should include three different kinds of research activities, which take place at different levels of analysis. First, there is the *disclosure level.* At this level, morally opaque practices and computer systems are analysed from the point of view of one or more relevant moral values, like privacy or justice. It is investigated whether and how the practice or system tends to promote or demote the relevant value. At this point, very little moral theory is introduced into the analysis, and only a coarse definition of the value in question is used that can be refined later on into the research.

Second, there is the *theoretical level* at which moral theory is developed and refined. As Jim Moor (1985) has pointed out, the changing settings and practices that emerge with new computer technology may yield new values, as well as require the reconsideration of old values. There may also be new moral dilemmas because of conflicting values that suddenly clash when brought together in new settings and practices. It may then be found that existing moral theory has not adequately theorized these values and value conflicts. Privacy, for example, is now recognized by many computer ethicists as requiring more attention than it has previously received in moral theory. In part, this is due to reconceptualizations of the private and public sphere, brought about by the use of computer technology, which has resulted in inadequacies in existing moral theory about privacy. It is part of the task of computer ethics to *further develop and modify existing moral theory* when, as in the case of privacy, existing theory is insufficient or inadequate in light of new demands generated by new practices involving computer technology.

Third, there is the *application level*, in which, in varying degrees of specificity and concreteness, moral theory is applied to analyses that are the outcome of research at the disclosure level. For example, the question of what amount of protection should be granted to software developers against the copying of their programs may be answered by applying consequentialist or natural law theories of property; and the question of what actions governments should take in helping citizens have access to computers may be answered by applying Rawls' principles of justice. The application level is where moral deliberation takes place. Usually, this involves the joint consideration of moral theory, moral judgements or intuitions and background facts or theories, rather than a slavish application of preexisting moral rules.

Disclosive ethics should not just be multi-level, ideally it should also be a *multi-disciplinary* endeavour, involving ethicists, computer scientists and social scientists. The disclosure level, particularly, is best approached in a multi-disciplinary fashion because research at this level often requires considerable knowledge of the technological aspects of the system or practice that is studied and may also require expertise in social science for the analysis of the way in which the functioning of systems is dependent on human actions, rules and institutions. Ideally, research at the disclosure level, and perhaps also at the application level, is best approached as a cooperative venture between computer scientists, social scientists and philosophers. If this cannot be attained, it should at least be carried out by researchers with an adequate interdisciplinary background.

3.3.2 Focus on public values

The importance of disclosive computer ethics is that it makes transparent moral features of practices and technologies that would otherwise remain hidden, thus making them available for ethical analysis and moral decision-making. In this way, it supplements mainstream computer ethics, which runs the risk of limiting itself to the more obvious ethical dilemmas in computing. An additional benefit is that it can point to novel solutions to moral dilemmas in mainstream computer ethics. Mainstream approaches tend to seek solutions for moral dilemmas through norms and policies that regulate usage. But some of these moral dilemmas can also be solved by redesigning, replacing or removing the technology that is used, or by modifying problematic background practices that condition usage. Disclosive ethics can bring these options into view. It thus reveals a broader arena for moral action, in which different parties responsible for the design, adoption, use and regulation of computer technology share responsibility for the moral consequences of using it, and in which the technology itself is made part of the equation.

In Brey (2000) I have proposed a set of values that disclosive computer ethics should focus on. This list included justice (fairness, non-discrimination), freedom (of speech, of assembly), autonomy, privacy and democracy. Many other values could be added, like trust, community, human dignity and moral accountability. These are all public values, which are moral and social values that are widely accepted in society. An emphasis on public values makes it more likely that analyses in disclosive ethics can find acceptance in society and that they stimulate better policies, design practices or practices of using technology. Of course, analysts will still have disagreements on the proper definition or operationalization of public values and the proper way of balancing them against each other and against other constraints like cost and usability, but such disagreements are inherent to ethics.

The choice for a particular set of values prior to analysis has been criticized by Introna (2005), who argues that disclosive computer ethics should rather focus on the revealing of hidden politics, interests and values in technological systems and practices, without prioritizing which values ought to be realized. This suggests a more descriptive approach to disclosive computer ethics opposed to the more normative approach proposed in Brey (2000).

3.4 Value-sensitive design

The idea that computer systems harbour values has stimulated research into the question how considerations of value can be made part of the design process (Flanagan, Nissenbaum and Howe 2008). Various authors have made proposals for incorporating considerations of value into design methodology. *Value-sensitive design* (VSD) is the most elaborate and influential of these approaches. VSD has been developed by computer scientist Batya Friedman and her associates (Friedman, Kahn and Borning 2006, Friedman and Kahn 2003) and is an approach to the design of computer systems and software that aims to account for and incorporate human values in a comprehensive manner throughout the design process. The theoretical foundation of value-sensitive design is provided in part by the embedded values approach, although it is emphasized that values can result from both design and the social context in which the technology is used, and usually emerge in the interaction between the two.

The VSD approach proposes investigations into values, designs, contexts of use and stakeholders with the aim of designing systems that incorporate and balance the values of different stakeholders. It aims to offer a set of methods, tools and procedures for designers by which they can systematically account for values in the design process. VSD builds on previous work in various fields, including computer ethics, social informatics (the study of information and communication tools in cultural and institutional contexts), computer-supported cooperative work (the study of how interdependent group work can be supported by means of computer systems) and participatory design (an approach to design that attempts to actively involve users in the design process to help ensure that products meet their needs and are usable). The focus of VSD is on ‘human values with ethical import’, such as privacy, freedom from bias, autonomy, trust, accountability, identity, universal usability, ownership and human welfare (Friedman and Kahn 2003, p. 1187).

VSD places much emphasis on the values and needs of *stakeholders*. Stakeholders are persons, groups or organizations whose interests can be affected by the use of an artefact. A distinction is made between direct and indirect stakeholders. Direct stakeholders are parties who interact directly with the computer system or its output. That is, they function in some way as users of the system. Indirect stakeholders include all other parties who are affected by the system. The VSD approach proposes that the values and interests of stakeholders are carefully balanced against each other in the design process. At the same time, it wants to maintain that the human and moral values it considers have standing independently of whether a particular person or group upholds them (Friedman and Kahn 2003, p. 1186). This stance poses a possible dilemma for the VSD approach: how to proceed if the values of stakeholders are at odds with supposedly universal moral values that the analyst independently brings to the table? This problem has perhaps not been sufficiently addressed in current work in VSD. In practice, fortunately, there will often be at least one stakeholder who has an interest in upholding a particular moral value that appears to be at stake. Still, this fact does not provide a principled solution for this problem.

3.4.1 VSD methodology

VSD often focuses on a technological system that is to be designed and investigates how human values can be accounted for in its design. However, designers may also focus on a particular value and explore its implications for the design of various systems, or on a particular context of use, and explore values and technologies that may play a role in it. With one of these three aims in mind, VSD then utilizes a tripartite methodology that involves three kinds of investigations: conceptual, empirical and technical. These investigations are undertaken congruently and are ultimately integrated with each other within the context of a particular case study.

Conceptual investigations aim to conceptualize and describe the values implicated in a design, as well as the stakeholders affected by it, and consider the appropriate trade-off between implicated values, including both moral and non-moral values. *Empirical* investigations focus on the human context in which the technological artefact is to be situated, so as to better anticipate on this context and to evaluate the success of particular designs. They include empirical studies of human behaviour, physiology, attitudes, values and needs of users and other stakeholders, and may also consider the organizational context in which the technology is used. Empirical investigations are important in order to assess what the values and needs of stakeholders are, how technological artefacts can be expected to be used, and how they can be expected to affect users and other stakeholders. *Technical* investigations, finally, study how properties of technological artefacts support or hinder human values and how computer systems and software may be designed proactively in order to support specific values that have been found important in the conceptual investigation.

Friedman, Kahn and Borning (2003) propose a series of steps that may be taken in VSD case studies. They are, respectively, the identification of the topic of investigation (a technological system, value or context of use), the identification of direct and indirect stakeholders, the identification of benefits and harms for each group, the mapping of these benefits and harms onto corresponding values, the conduction of a conceptual investigation of key values, the identification of potential value conflicts and the proposal of solutions for them, and the integration of resulting value considerations with the larger objectives of the organization(s) that have a stake in the design.

3.4.2 VSD in practice

A substantial number of case studies within the VSD framework have been completed, covering a broad range of technologies and values (see Friedman and Freier 2005 for references). To see how VSD is brought into practice, two case studies will now be described in brief.

In one study, Friedman, Howe and Felten (2002) analyse how the value of informed consent (in relation to online interactions of end-users) might be better implemented in the Mozilla browser, which is an open-source browser. They first undertook an initial conceptual investigation of the notion of informed consent, outlining real-world conditions that would have to be met for it, like disclosure of benefits and risks, voluntariness of choice and clear communication in a language understood by the user. They then considered the extent to which features of existing browsers already supported these conditions. Next, they identified conditions that were supported insufficiently by these features, and defined new design goals to attain this support. For example, they found that users should have a better global understanding of cookie uses and benefits and harms, and should have a better ability to manage cookies with minimal distraction. Finally, they attempted to come up with designs of new features that satisfied these goals, and proceeded to implement them into the Mozilla browser.

In a second study, reported in Friedman, Kahn and Borning (2006), Kahn, Friedman and their colleagues consider the design of a system consisting of a plasma display and a high-definition TV camera. The display is to be hung in interior offices and the camera is to be located outside, aimed at a natural landscape. The display was to function as an ‘augmented window’ on nature that was to increase emotional well-being, physical health and creativity in workers. In their VSD investigation, they operationalized some of these values and sought to investigate in a laboratory context whether they were realized in office workers, which they found they did. They then also identified indirect stakeholders of the system. These included those individuals that were unwittingly filmed by the camera. Further research indicated that many of them felt that the system violated their privacy. The authors concluded that if the system is to be further developed and used, this privacy issue must first be solved. It may be noted, in passing, that, whilst in these two examples only a few values appear to be at stake, other case studies consider a much larger number of values, and identify many more stakeholders.

3.5 Conclusion

This chapter focused on the embedded values approach, which holds that computer systems and software are capable of harbouring embedded or 'built-in' values, and on two derivative approaches, disclosive computer ethics and value-sensitive design. It has been argued that, in spite of powerful arguments for the neutrality of technology, a good case can be made that technological artefacts, including computer systems, can be value-laden. The notion of an embedded value was defined as a built-in tendency in an artefact to promote or harm the realization of a value that manifests itself across the central uses of an artefact in ordinary contexts of use. Examples of such values in information technology were provided, and it was argued that such values can emerge because they are held by designers or society at large, because of technical constraints or considerations, or because of a changing context of use.

Next, the discussion shifted to disclosive computer ethics, which was described as an attempt to incorporate the notion of embedded values into a comprehensive approach to computer ethics. Disclosive computer ethics focuses on morally opaque practices in computing and aims to identify, analyse and morally evaluate such practices. Many practices in computing are morally opaque because they depend on computer systems that contain embedded values that are not recognized as such. Therefore, disclosive ethics frequently focuses on such embedded values. Finally, value-sensitive design was discussed. This is a framework for accounting for values in a comprehensive manner in the design of systems and software. The approach was related to the embedded values approach and its main assumptions and methodological principles were discussed.

Much work still remains to be done within the three approaches. The embedded values approach could still benefit from more theoretical and conceptual work, particularly regarding the very notion of an embedded value and its relation to both the material features of artefacts and their context of use. Disclosive computer ethics could benefit from further elaboration of its central concepts and assumptions, a better integration with mainstream computer ethics and more case studies. And VSD could still benefit from further development of its methodology, its integration with accepted methodologies in information systems design and software engineering, and more case studies. In addition, more attention needs to be invested into the problematic tension between the values of stakeholders and supposedly universal moral values brought in by analysts. Yet, they constitute exciting new approaches in the fields of computer ethics and computer science. In ethics, they represent an interesting shift in focus from human agency to technological artefacts and systems. In computer science, they represent an interesting shift from utilitarian and economic concerns to a concern for human values in design. As a result, they promise both a better and more complete computer ethics as well as improved design practices in both computer science and engineering that may result in technology that lives up better to our moral and public values.