Bangladesh Journal of Public Administration Volume XXII Number II 2013 ISSN 1563-5032

Computer Software, Moral Theories and Policy Vacuums: An Analysis

Dr. Md. Munir Hossain Talukder [1]

Abstract

Computer software is a unique technological advancement which creates enormous possibilities, but also raises complex moral problems. In one way, these moral problems are not entirely new in character; rather a new dimension of traditional moral dilemmas. On the other, we are not ready yet to tackle them with traditional moral norms andprinciples. Several new normative guidelines are required for computer technology related issues, such as, privacy, property rights, internet control, confidentiality, cyber harm, and cyber risk. More importantly, software ownership right, a right which is absent in traditional ethical discussions, is necessary to protect. However, as software is a completely new creative work, traditional legal framework (e.g. copy right, patent, and trade secrecy laws) and philosophical ownership theories (e.g. labour theory, utilitarian theory) are inappropriate to protect the right of software owner or its inventor: A lack of effective ownership laws, inadequate ethical principles as well as traditional philosophical grounding have created policy vacuum in computer software property rights.

Keywords: Free software, Policy vacuums, Property rights, Software ownership, Traditional moral principles, Uniqueness

1.0 Introduction

Technological revolutions have significant impact on society and human behaviour. Technology has gained the power to influence at every stages of our life. We create new technology to live a more and more comfortable life. Computer, information technology, and coinmunication engineering, are some of the recent intellectual developments of human knowledge. In fact, our economy, health, national security, research, communication, entertainment, and so

[1] **Associate Professor, Department of Philosophy, Jahangirnagar University, Savar, Dhaka-1342**
Email: mdmhtalukder@gmail.com

forth, all are now regulated trough computers. We are leading a more sophisticated life than before. However, computer technology brings good as well as some bad consequences on society and individuals. For this reason, ethical justification is associated with this type of technology. Computer ethics analyzes the ethical problems which arise by using computer technology like privacy, security, intellectual property right, etc. Since computer technology is rapidly changing, new developments and inventions are taking place even within a day. Ethicists are in dilemma to determine whether traditional ethical theory is sufficient, or completely new ethical framework is necessary for solving these new problems.

Policy makers are also facing troubles of continuous conceptual development in these fields. By conceptual development, we mean the modifications or addition of new knowledge in software. For example, MS Office 97, MS Office 98, MS Office 2000, MS Office XP, etc. are the gradual developments of Office software. Although computer ethical problems are not wholly new in character they are new in dimensions. Conversely, the ethical problems related to computer are traditional like privacy, risk, harm, property rights, etc. However, they are ***not*** exactly same as these issues discussed since the long time. They possess new explanations while associate with computer technology. For example, ownership right is traditional, but ownership of software is new. As a result, existing policies may not be sufficient to cover all the issues and artifacts in computer technology. For instance, what new laws, policies, protocols, guidelines, are appropriate to own software as intellectual property, are still being discussed. According to some ethicists (e.g. Debrah Johnson), existing forms of copy right, patent, and trade secrecy laws are insufficient to protect this intellectual rights. Therefore, computer software produces policy vacuums. I will explore these policy vacuums, and argue that we need a new ethical approach to fill them.

2.0 The uniqueness of software ownership rights

By the term 'unique', here we refer to unique technology, and not unique ethical problems. Software as a technology is unique, and the claim to own software as a private intellectual property is also unique because traditional philosophical theories, and also legal form of property rights are insufficient to explain and protect software property rights in a satisfactory way. However, ethical issues adjacent to software property are privacy, confidentiality, risk, etc. which are traditional and not unique in character.

There is an on going debate whether computer technology creates unique ethical problems or not. Some ethicists (e.g. Johnson) argue that ethical issues about computer technology are not new or unique at all. For example, privacy, uncertainty, intellectual property rights, etc. are all issues which are traditional. We have to face these issues when any new technology (e.g. gene technology, nuclear power plant technology) is introduced to the society. Johnson mentions that computer and information technology are not the first technologies raising such types of ethical problems. According to her, computer ethical problems are not unique, but these are new variations of old problem. She writes,

> I propose that we think of the ethical issues surrounding computer and information technology as ***new*** species of ***general, or traditional moral issues.*** The idea is that the ethical issues surrounding computer and information technology can be understood as variations of traditional ethical problems or issues. (Johnson 2001, p.16)

We may refer to this position as "traditional view" which holds that traditional ethical theories or moral framework is sufficient to solve computer ethical problems. Johnson supports this traditional view.

Contrary to traditional view, proponents of "uniqueness view" argue that computer technology deserves special status, and creates completely new ethical problems because this technology does not exist before. We are not able to understand and analyze these new problems through traditional moral principles or theories. So, they claim a new ethical foundation is necessary. As Walter Maner writes,

> ...there must be a unique domain for computer ethics distinct from the domain for moral education, distinct even from the domains of other kinds of professional and applied ethics. Like James Moor, I believe computers are special technology and raise special ethical issues, hence that computer ethics deserves special status. (Maner 2004, p.41)

Maner holds that there is a gap between traditional ethical principles and new ethical problems. Uniqueness advocates have put forwarded 'software ownership rights' issue to consider their claim. Computer software is a new form of intellectual property. It does not exist before computer technology appeared. Since software is not a physical property this unique entity raises complex ethical and legal dispute. Software has special type of character, and hence, present forms of legal rules are insufficient. Should software be treated as private property? Do we have the right to own algorithmic and numerical data? All these moral questions are complex and new as software is unique, and therefore, software ownership right is unique.

In the next section, I will show that present policies and regulations are inadequate.

3.0 Present software policy is inadequate

Computer software brings new opportunities and possibilities for us. It is now an important component of today's science and technology. One example is that from food processing to sending rockets everything is dependent on computers. Lundestad and Hommels write, "Our professional and personal lives have become unthinkable without the possibilities these technologies offer us to communicate and do business" (Lundestad and Hommels 2007, p.90).

As technology carries new hopes as well as some new risk, software is not without risk. For. instance, someone may able to collect all of the data and information about national security of a country by using software. Or, someone may able to collect personal information without taking consent. Software may provide influential economic risk at individual and organizational level. If one copies a newly invented software, and sell it to those countries where law and order situation are not so strict, the originator company will face extreme financial loss. However, at present software is protected by laws.

The aim of present software policy and laws is to inspire invention or creation of new knowledge in this field. Copy right, patent, and trade secrecy laws, are primarily based on traditional utilitarian principle. The basic formulation of this principle is: *"Everyone ought to act so as to bring about the greatest amount of happiness for the greatest number of people"* (Johnson *200*1, p.36). When programmers or developers create or develop new software, they expect to get some financial benefit from their invention. Or, at least they want to recover production cost. Financial incentive will encourage them to do better work next.

Copy right laws are applied to protect creative works like **literature,** music, film, **etc.** These laws are also applicable to computer software. Authors can protect their work for whole life time, and further 70 years by the copy right laws. In this timeframe no one can copy, or reproduce this work without author's permission. Copy right laws can only protect expression of ideas, but not ideas themselves. So, when applying to computer software these laws become weak to protect programmers or developers right since software is mainly based on mathematical algorithm which is an idea, not expressions of ideas. Another limitation is that if someone copies software, it is the responsibility of that software creator to prove.

Patent is rather stronger type of ownership laws. These laws give full control to the programmers or developers up to 17 years to 22 years from creation. Patent protection gives the right to inventors to sell their work as well as to protect others from copy or reproduction. Limitation of patent protection is that it is too costly.

Only big companies are able to afford it. In order to give patent licenses the court has to find out whether someone has already patented this work, or not. Trade secrecy protection is also applicable to big firm or companies. They can protect any business secret such as, formula, procedure, by law. Any software company may keep its software secret than sell by these laws. The limitation of trade secrecy law is that if someone is able to know the secret then the company will loss the whole investment (Johnson 2001).

As mentioned earlier, the aim of these laws is to inspire creation and help to make original contribution to the advancement of science and technology. Software is an intellectual creativity. Everybody wants to get recognition and inspiration for their creation. If programmers and developers creations are not acknowledged then they may be disinterested for further development. So, incentive as well as inspiration is necessary for software development. As software is now protected by laws, more and more people are interested to contribute to this field. Many inventions are now protected by copy right, patent and even trade secrecy laws. Nonetheless, there is a reverse impact of these legal systems. Johnson identifies this situation as "Building Blocks". She writes, "Ownership of the building blocks would interfere with invention insofar as new inventors would have to seek permission to use these building blocks from private owners" (Johnson 2004, pp.289-290).

According to some ethicists, building blocks are contra to intellectual freedom. They argue that although new researchers and programmers might buy the right to use, sometimes it might be costly, or unavailable to them. Another reverse impact of building blocks is that most companies employ software developers or programmers and own the right of their creation. As a result, on the one hand, programmers are selling labour as well as losing their intellectual rights and on the other hand, a lot of people in the world are unable to buy this software for high price. Lau writes, "*The price of original software is positively correlated with lenient attitudes to software piracy*" (Lau 2003, p.236).

The major aim of present utilitarian policy seems to make profit, and to recover production cost for the software companies. Big companies are financially solvent, and able to take financial plus technological risk. But small companies and individuals are also affected by this policy. These companies are incapable to sustain in the market any more. They are feeling pressure to merge with computer giants or stop their creative production. As Spinello points out:

> Microsoft's intention was to leverage its Windows monopoly in order to defeat Netscape...Further, Microsoft had no intention of competing on a level playing field: Gates' own words unambiguously confirm this conclusion. (Spinello 2003, p.130)

Moreover, big companies are also facing financial risk for increasing software duplicity world wide. Lau mentions,"The revenue loss of the worldwide software industry due to software piracy amounted to S$10.97 billion in 2001" (Lau 2003, p.233).

Therefore, present software policy is inadequate and many people are arguing for free software.

4.0 Free software or owned software?

One of the strong arguments for inadequacy of present software property rights policy is that there is no 'ethical agreement' whether software should be free or owned. Some ethicists (e.g. Johnson) argue that software should be owned as this is an intellectual property. We should encourage and facilitate creativity or invention in computer software by giving programmers or developers ownership rights. Conversely, others (e.g. Stallman) argue that software ownership has negative impact on the society. According to them, programmers should not do only what is profitable for them. Programmers have a duty to inspire others by sharing their inventions. Law should not protect the right of profit, but rather ensure morality. However, supporters of free software are unable to give justification for imposing this special duty to programmers. In particular, why do software programmers have a duty unlike other profession? We will discuss both of these theses in the light of philosophical ownership theories. But before that, we need to analyze the status of software as a commodity.

Apparently, software is a commodity or product like other commodities (e.g. car). Nevertheless, because of its use and influence on our lives, software seems more than a commodity. Software is using in education, life saving medical equipment, research and academic purposes, etc. Cars are luxurious goods whereas software may be public good, or at least quasi-public good, as Hawkins refers. He writes,

> Public goods are of interest because while society benefits from the provision of the good, no individual entity receives enough benefit to provide the good....When software is open sourced, is very close to meeting the definition of a public good.(Hawkins 2004, p.114)

From this perspective big companies (e.g. IBM) are investing on open source software because open source software is also profitable as propriety software. So, there is a new trend in software property rights and software business policy. As Hawkins writes, "...in which firms have turned (or attempted to turn) proprietary software into open source software, presumably expecting greater profits than if the code were kept proprietary" (Hawkins 2004, p.104).

There arc different types of ownership theory in philosophy. Among these theories, labour theory and utilitarian theory have wide implications on software ownership. But, what is ownership? We may 'own' a car, a house, or even a computer. By the word 'own' we meant here the right to control, right to use, right to sell, etc. However, ownership does not imply absolute right i.e. we may not burn the car as it may harm others (Bynum & Rogerson 2004).

Philosopher John Locke constructed the labour theory. He argued that someone who produces or creates a product has obtained the right to own of that product, because of her or his investment of labour to create that production. The law ought to protect this ownership right of the labourer. The labourer will not destroy this product, rather leave it for the next generation. The aim of labour theory was to protect natural resources like land, crops, etc. However, this theory may easily be extended to the software property rights. A programmer or developer may claim that created software is her or his property as she or he invested own labour to create it. Although labour theory has strong position to protect ownership rights, it has some limitations. Specially, when it applies to software property rights. According to Johnson, labour theory does not distinguish between 'tangible' and 'nontangible' property. By tangible property we mean physical property like house, car, computer, etc. and nontangible property denotes intellectual property like poem, music, software, etc. When a labourer loses tangible property, she or he loses whole labour given for it. But, when someone duplicates a software or performs others music, the labourer does not lose her or his property because two people may sing the same song differently.

Considering this limitation of labour theory Johnson writes, "Because of the reproducibility of computer software, the labor theory of property cannot be used to justify the assignment of property rights to software developers" (Johnson 2001, p. 156). The unique character of software as Johnson mentions is "reproducibility". If we follow labour theory then it gives rights to the programmers, but unable to protect reproducibility. As a result, programmers are not losing the property rather losing income which should be protected. Johnson also mentions this point:

> Even though Locke's labor theory does not provide a justification for property rights in computer software...while software developers do not lose their software, they do lose something very valuable, they lose *the capacity to sell(and make money from) their creations.* (Johnson 2001, p. 156)

The utilitarian theory has some advantage over labour theory. Software programmers or developers want to own their invention as well as the right to sell their product. So, they have claim for not only property rights but also economic rights. Economic right is "... a right that gives them the capacity to sell and, if successful at selling, make a profit from their creation" (Johnson 2001, p. 157). According to utilitarian principle, "...property rights should be recognized, promoted, and protected in order to maximize happiness and well-being within the community and minimize pain and sorrow" (Bynum & Rogerson 2004, p.280).

Notice, arguments for and against software ownership both are based on utilitarian principle. Johnson argues for software ownership. According to her, if software is not owned then developers and companies will be disinterested to develop or invent new software. As a result, innovation and creativity will be stopped. Lack of incentive may create bad consequences on the society. For promoting software knowledge, incentive is necessary. Another argument is that if we protect software ownership then it will inspire future developers. She argues for sufficient legal frame work to protect software ownership that will eventually encourage invention and creativity.

In contrast, Stallman argues against software ownership from the same utilitarian perspective. His claim is that software should be free, because the restriction of software property rights will create material as well as psychosocial harm. Material harm includes "Fewer people use the program, none of the users can adopt or fix the program, other developers cannot learn from the program, or base new work on it" (Stallman 2004, p.297). Psychosocial harm includes "...effect that people's decisions have on their subsequent feelings, attitudes, and predispositions" (Stallman 2004, p.297).

Stallman rejects the earlier argument 'If there is no incentive there will be no software'. According to him, this argument is not satisfactory because this argument is based on only two possibilities (one is proprietary software and another is no software) .There must be other possibilities. These are open source software, free software, etc.

From the above discussion, we may say that there is no moral agreement whether software should be owned or free. We have also discussed that existing intellectual property right laws are insufficient as well. All these issues create policy vacuums in computer software.

5.0 What should we do to fill policy vacuums in computer software property rights?

It is analyzed that computer software is a unique entity. This is unique as a sense of technology, and complex as a moral problem. Both philosophical theories and legal practice have some limitations to handle this technology in a satisfactory way. These limitations generate policy vacuums in computer software property rights. Existing software policy is taking software as other business commodities, and weighing on cost-benefit analysis. Freedom of intellectuality is facing some restrictions. Companies want to be benefited as much as possible, while users want to buy software at a minimum cost. Many people in the world are unable to buy genuine software for some reasons, such as social structure, price, availability, lack of knowledge, and so forth. As a result, illegal software business is growing day by day and creating market all over the world, specially in the developing countries. These illegal businessmen are the most beneficiary agents with minimum investment. We need to resist this illegal software. To do so, legal framework may not be sufficient. Only law will not change people's behaviour, moral belief, moral practice, etc. We must have to have a good policy based on sufficient ethical principles. Mere cost-benefit analysis which based on utilitarian principle should be changed to fill policy vacuums in computer software property rights.

James Moor also points out these policy vacuums in software, and argues for 'Just Consequentialism' to fill up. Just Consequentialism emphasizes on justice. It is a unification of consequentialist and deontological ethical theories. Moor holds that traditional consequentialist ethical theory has significant limitations (e.g. it does not consider justice seriously). He takes impartiality as one of the main elements of justice. According to him, policies should be impartial, what we believe harmful for us should believe harmful for others. Existing consequentialist ethics ignores impartiality as well as justice (Moor 1999). However, Horner criticizes Moor's theory by pointing that if we follow Just Consequentialism then we have to have an ability to predict the outcomes of consequences in advance which is "problematic" and unrealistic (Horner 2005).

Therefore, no existing or proposed ethical approach is appropriate to fill the policy vacuums in software property rights. We need a new ethical approach which will be able to combine rights, utility, cost-benefit, justice, intellectual freedom, and so forth.

6.0 Conclusion

Computer software is a unique entity. Existing forms of property rights laws, and ownership theories, are insufficient to handle this unique entity in a satisfactory way. As a result, there is a policy vacuum in computer software property rights. Present policy is based on traditional utilitarian principle and cost-benefit analysis. This policy is inadequate as software duplicity is growing day by day. There are also moral arguments for and against software ownership rights. Some ethicists argue that software property rights should be owned, whereas others ague that software property rights should be free. Traditional ethical approach is not sufficient to answer what laws or theories we should formulate to protect software property rights, how software duplicity could be protected, etc. Therefore, we need a new ethical approach to fill policy vacuums in computer software property rights.

Acknowledgements: Main idea of this paper was presented at Ethics Institute, Utrecht University, the Netherlands. I thank all discussants for their valuable questions and feedback.

REFERENCES

Bynum T.W., Rogerson S. (eds.), *Computer Ethics and Professional Responsibility.* Oxford: Blackwell Publishing Limited, 2004.

Hawkins, R.E., "The economics of open source software for a competitive firm why give it away for free?", *.Netnomics* 6, 2004, 103-117.

Himma, K.E., "The relatinship between the uniqueness of computer ethics and its independence as a discipline in applied ethics", *Ethics and Information Technology* 5, 2003, 225-237.

Horner, D.S. (2005), "Policies for a 'nanosociety': Can we learn now 'from our future mistakes?". In: *Reader* Ethics and Technology, 2007, Utrecht University, [1-12].

Johnson, D.G., *Computer Ethics,* 3rd ed. New Jersey: Prentice-Hall, 2001.

Johnson, D.G., "Property rights in computer software: individual and policy issues". In: Bynum T.W., Rogerson S. (eds.), *Computer Ethics and Professional Responsibility,* Oxford, Blackwell Publishing Limited, 2004, 285-293.

Lau, E.K.W, "An empirical study of software piracy", *Business Ethics: A European Review* 12 (3), 2003,233-245.

Lundestad C.V. and Hommels A., "Software vulnerability due to practical drift", *Ethics and Information Technology* 9, 2007, 89-100.

Maner,W., "Unique ethical problems in information technology". In: Bynum T.W., Rogerson S. (eds.), *Computer Ethics and Professional Responsibility,* Oxford, Blackwell Publishing Limited, 2004, 39-59.

Moor, J.H., "Just consequentialism and computing", *Ethics and Information Technology* 1, 1999, 65-69.

Moor, J.H., "Reason, relativity and responsibility in computer ethics". In: Bynum T.W., Rogerson S. (eds.), *Computer Ethics and Professional Responsibility.* Oxford, Blackwell Publishing Limited, 2004, 21-38.

Spinello, R.A., "The case against Microsoft: an ethical perspective", *Business Ethics: A European Review* 12 (2), 2003, 116-132.

Stallman, R., "Why software should be free". In: Bynum T.W., Rogerson S. (eds.), *Computer Ethics and Professional Responsibility,* Oxford, Blackwell Publishing Limited, 2004, 294-310.

Tavani, H. T., "The uniqueness debate in computer ethics: what exactly is at issue, and why does it matter?", *Ethics and Information Technology* 4, 2002, 37-54.