An ethical review of information systems development
The Australian Computer Society's code of ethics and SSADM

Simon Rogerson
Centre for Computing and Social Responsibility, De Montfort University, Leicester, UK
John Weckert
ARC Commonwealth Special Research Centre for Applied Philosophy and Public Ethics, Charles Sturt University, Wagga Wagga, Australia and
Chris Simpson
School of Information Technology, Swinburne University of Technology, Hawthorn, Australia

Keywords *Ethics, Information systems, Systems development, Australia*

Abstract *The rapid advance of computer-based technology has led to social policy vacuums. Most information systems development tools concentrate upon technical issues, and offer few if any guidelines that address the moral issues inherent in new application possibilities. It is argued that extension of such tools to include ethical and moral, human and environmental issues is possible. A good starting point is provided by mapping relevant clauses of professional codes of ethics upon each stage of the development methodology. We use as examples the Australian Computer Society Code of Ethics and the structured systems analysis and design method (SSADM).*

Introduction

Moor (1985) asserted that computers are "logically malleable" in the sense that "they can be shaped and moulded to do any activity that can be characterised in terms of inputs, outputs, and connecting logical operations". This might lead to policy vacuums caused by possibilities that did not exist before computers. In these situations there are often no useful analogies to draw upon for help. Maner (1996) states that:

> Lack of an effective analogy forces us to discover new moral values, formulate new moral principles, develop new policies, and find new ways to think about the issues presented to us.

While this may be a little strong – it is not altogether clear just what a new moral principle or value might be – we must certainly apply principles in new situations. In such situations Gotterbarn (1992) suggests that professionals must be aware of their professional responsibilities, have available methods for resolving non-technical ethics questions and develop proactive skills to reduce the likelihood of ethical problems occurring. Those involved in information

Information Technology & People, Vol. 13 No. 2, 2000, pp. 121-136.

systems (ISs) development are one group significantly affected by this ethically volatile situation.

Much of the work in computer ethics has been concentrated in the philosophy and sociology disciplines with restricted input from the ISs and computer science disciplines. This may be the reason why Walsham (1996) claims that "... there is little published work which directly relates these [specific IS-related ethical] issues to more general ethical theory ...". The work has tended to be conceptual and more of a commentary on computer phenomena than an attempt to develop strategies to identify and address societal and ethical issues associated with ISs and the underpinning information technology (IT) (Rogerson and Bynum, 1996). There are some notable exceptions, however, including Laudon (1995), Mason (1995), Anderson *et al.* (1993), Mason *et al.* (1995), de Raadt (1997), Harris *et al.* (1997) and Martin and Schinzinger (1996).

The purpose of this paper is to address some of the ethical challenges associated with ISs development. Specifically, various principles from the Code of Ethics of the Australian Computer Society (ACS) will be applied to a particular methodology. The structured systems analysis and design method (SSADM) will be used as an illustrative example because it is felt that if ethical enrichment can be achieved in a "hard" systems approach then it is likely to be achievable in most approaches. The ethical enhancement of SSADM is a considerable task that is beyond the scope of the present study. This discussion is confined to making some suggestions regarding how that task might be achieved employing the aforementioned code of ethics. Several important assumptions underlie the paper. One is that both deontological and consequential theories of ethics are important here. For practical purposes we must often follow rules, but consequences are important in the formulation of rules, and for making decisions when rules clash. A second assumption is that ethical principles are, in an important sense, objective. Both of these assumptions are contentious, but detailed examination of them would allow no time for the issue at hand.

ISs development

Turning to the overall development process, there are numerous methodological approaches to ISs development. Few deal adequately with the ethical dimensions of systems development. Avison (1995) criticises the development methodologies, such as SSADM, Merise and Yourdon, that are adopted by most organisations today because they tend to stress formal and technical aspects. He argues that:

> The emphasis ... must move away from technical systems which have behavioural and social problems to social systems, which rely to an increasing extent on information technology.

He suggests that the human, social and organisational aspects are often overlooked. The consideration of this broader perspective only seems to occur in the event of systems failure or under-performance. This issue is addressed by

Wood-Harper *et al.* (1996) who identify a number of dilemmas which a systems analyst might face when undertaking a systems development activity using a methodological approach. These dilemmas are summarised as:

- whose ethical perspective will dominate the study of the situation and the development of the IS?
- will ethical viewpoints be included in the study?
- what methodology should be used for the study?
- what approach should the analyst use if there is an obvious conflict of interests?

It is important to recognise that there are a few methodological approaches, notably ETHICS from Mumford (1983), Soft Systems Methodology from Checkland (1981) and Multiview from Avison and Wood-Harper (1990) that attempt to include consideration of ethical and societal issues. In evaluating ETHICS, Jayaratna (1994) suggests that it:

> offers many design guidelines useful for the understanding and the design of human-centred systems, but . . . does not offer models or ways for performing . . . the steps. Nor does it offer any models for handling interpersonal and political conflicts.

He concludes that:

> ETHICS is committed to a particular ethical stance [and] does not offer any means of discussing or resolving many of the ethical dilemmas . . . in systems development.

This appears to be a recurrent criticism of such methodologies. Whilst it is laudable that ethical sensitivity is raised as an issue worthy of investigation, the manner in which investigation is undertaken and an ethically defensible position derived is vague.

Methodologies need to be enhanced to address these criticisms. It seems clear that research is required into the generalised manner in which ethical issues can be accounted for together with enhancement of specific methodologies. This paper specifically addresses the question, "Will ethical viewpoints be included in the study?" The approach explored here is to use the relevant elements of a code of conduct to create an analytical tool that can be used to identify where and how to ethically enrich an ISs development approach. The code of the ACS has been chosen for this purpose (see the Appendix).

The ACS code as an analytical tool

ISs practitioners will readily acknowledge that they are professionals who use computers to resolve organisational problems. As professionals they should be willing to adhere to a professional code of conduct that embeds ethical principles (Rogerson, 1997). The ACS has developed a code of ethics, which constitutes a comprehensive and appropriate tool to provide guidance in computing practice.

The ACS Code of Ethics and Standard of Conduct is largely reproduced in the Appendix to this paper. It is segmented into six areas of concern (priorities,

competence, honesty, social implications, professional development, and information technology profession). Although this contrasts with the British Computer Society (BCS) code, which focuses upon the public interest, duty to the employer and clients, duty to the profession and professional competence and integrity, in fact the detail when viewed overall is quite similar. Still fewer headings are used in the Association for Computing Machinery (ACM), namely, general moral imperatives, more specific professional responsibilities, organizational leadership imperatives and compliance with the code.

In reality, it could be argued that the headings used in the ACS code lend themselves better to the "hard" checklist approach of SSADM than the other codes, for example, to achieve a review of proper priorities and likewise for social implications.

The introduction to the code states the following:

> *A Requirement*
>
> An essential characteristic of a profession is the need for its members to abide by a code of ethics. The Society requires its members to subscribe to a set of values and ideals which uphold and advance the honour, dignity and effectiveness of the profession of information technology. The member is required at all times to be honest, forthright and impartial, to loyally serve employers, clients and the public, to strive to increase the competence and prestige of the profession and to use special knowledge and skill for the advancement of human welfare.

The guidelines clearly imply that the future of the computing profession depends on both technical and ethical excellence and consequently any violation has to be considered as unprofessional practice. It is worth noting here that, in certain circumstances, some violation may be impossible to avoid. It is not too difficult to imagine a situation in which it is not possible to "loyally serve employers, clients and the public" simultaneously. This is recognised, at least minimally, in 1.5. of the code: "I will advise my client or employer of any potential conflicts of interest between my assignment and legal or other accepted community requirements". No clear guidelines are given, however, on the best methods of resolving these conflicts.

Particular areas of the code exist that should be considered during an IS development project. These can be identified and used to measure how methodologies attend to ethical considerations. They become tools for deciding how methodologies might have to be modified. This paper indicates how this may be done.

Compared with other codes, the architecture of the ACS code is categorised into areas of concern that make it relatively easy to create pragmatic checklists for adding to the SSADM procedures, particularly when considering "priorities" and "social implications".

Clearly there are still gaps in the code when it comes to detailed considerations of, for example, flow-on effects of one's product and services to clients and further into the community. No imperative is stated to consider far-reaching secondary effects or the desirability of resulting cultural changes. The closest we get to this is in the ACM code: "1.1 Contribute to society and human

wellbeing", "2.5. Give comprehensive and thorough evaluations of computer systems and their impacts, including analysis of possible risks", and "2.7. Improve public understanding of computing and its consequences". These are rather vague. "Public health and safety" and "environment" are mentioned in both BCS (1.) and ACS (4.1) codes. BCS (4.) mentions "basic human rights" and ACS refers to "people's privacy affected by my work" in (4.2). ACS section (4.5) is even less specific: "increase the feelings of personal satisfaction, competence and control of those affected by my work". We return to the issue of far-reaching secondary effects further on in this paper.

In producing an extension to the SSADM checklist, the essential ethical questions can be explicitly detailed and revised periodically as professional wisdom and sensitivity grows.

An overview of SSADM version 4

SSADM is now used as an illustrative method to further discussion of the ethics of ISs development. SSADM is a set of procedural, technical and documentation standards for ISs development (Skidmore *et al.*, 1994). SSADM comprises five core modules: feasibility study, requirements analysis, requirements specification, logical systems specification, and physical design module. The position of these modules in the development life cycle is shown in Figure 1 (Skidmore *et al.*, 1994). SSADM adopts a product oriented approach where each element, be it a module, stage or step, produces predictable outputs from given inputs.

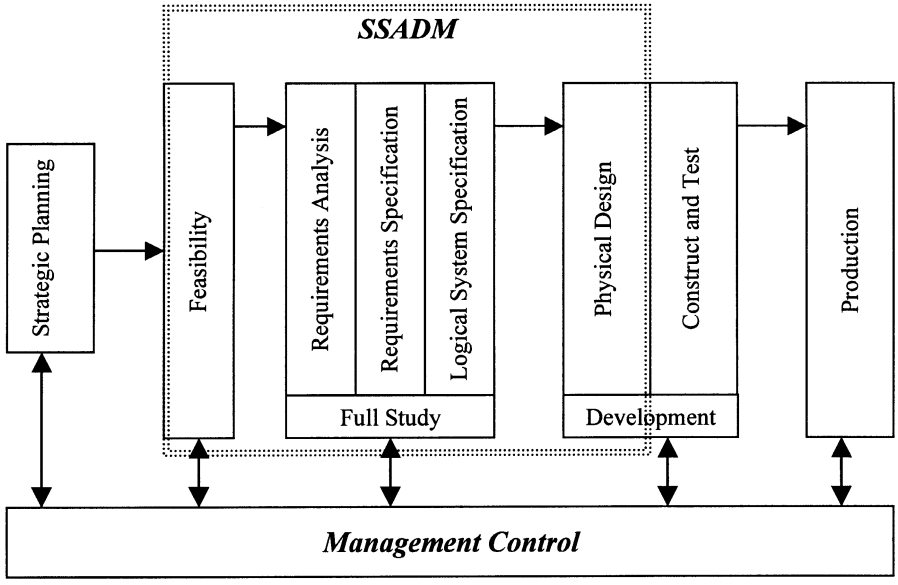

Figure 1.
Position of SSADM in the systems development life cycle

Core modules

Each of the five self-contained modules has a set of objectives:

(1) Feasibility study:
- to determine whether systems can be developed to meet defined business and technical objectives within specified financial and operational constraints.

(2) Requirements analysis:
- to determine the application scope;
- to establish how to integrate IT with other needs;
- to form an overall view of system costs and benefits;
- to confirm the viability of continuing further;
- to achieve user buy-in of the requirement.

(3) Requirements specification:
- to produce a complete specification that informs the subsequent logical systems specification.

(4) Logical systems specification:
- to enable management to select the technical environment that offers the best value for the money;
- to provide an independent non-procedural specification of system functionality.

(5) Physical design module:
- to develop a physical design that defines data, processes, inputs and outputs whilst incorporating installation standards;
- to form the basis for system construction.

Modelling perspectives

There are three modelling perspectives within SSADM:

(1) *Functions*: this is an attempt to capture the user's view of system processing. Techniques used include data flow modelling.

(2) *Events*: these are business activities that trigger processes to update system data. Techniques involved include entity life histories and affect correspondence diagrams.

(3) *Data*: this is a description of the data used in the system utilising a logical data model.

Products

Products are defined within a product breakdown structure hierarchy, the top level of which has three elements: management products, technical products, and quality products. These three categories are designed to be complementary

to ensure that a high quality solution is provided in a managed and controlled way (CCTA, 1990). Management products are used in planning and controlling the project, technical products document how the project proposes to realise the project objectives, and quality projects demonstrate how quality has been built into the system.

Reviewing SSADM

Winter *et al.* (1995) explain that computer-based ISs are systems that serve purposeful human actions. They argue that there is a very heavy emphasis on the serving IS in the systems development life cycle with, in the main, only limited and implicit account taken of the purposeful human action or the so-called served system. This is particularly the case in the "hard" systems thinking approaches of structured methods that are highly technical and rational. In examining SSADM, Winter *et al.* (1995) found no explicit organisational activity model. There also appeared to be an acceptance without question of the image, activities and performance aspects of the organisation in question. Furthermore, they were critical of the information requirements of SSADM's concept, which concerned the information required for the system to function rather than the information needed by the people undertaking organisational activities. Upon analysis of the ACS code, several of the articles detailed in its Standards of Conduct align with the areas of criticism within SSADM. It is argued that a method needs to address effectively these areas before it can be considered to be a professional approach. Clearly the authors have reservations about SSADM. If it is to be a professional approach it must be modified in some way, or used in conjunction with a complementary method, in order to overcome its deficiencies.

The approach adopted by Winter *et al.* (1995) is organisationally oriented. It does not appear to go beyond the environs of the organisation. Potential well-being of many individuals is likely to be at risk unless an ethically sensitive horizon is established for the scope of consideration. Issues beyond the horizon are deemed not to influence the project or be influenced by the project. The scope of consideration is an ethical hot-spot that is influenced by the identification and involvement of all stakeholders including those beyond the organisational boundary (Rogerson, 1997). It is this that is particularly addressed in articles 1.5, 4.1, 4.2 and 4.5 of the ACS code. This is an area not effectively covered by SSADM and it is unclear whether distant stakeholders will be identified let alone involved. The implications of such restricted stakeholder involvement on achieving a socially and ethically sensitive approach are obvious.

The ACS code, in the last three articles just mentioned, talks of "those affected by my work". The "ethically sensitive horizon" is clearly relevant here. In order to make clearer some of the issues regarding stakeholders, it will be useful to identify the main groups, following Collins *et al.* (1994). They identify four principal groups: developers who develop and sell the system; buyers who purchase and own the system; users who use the system; and the penumbra, comprising anyone affected by the use of the system. These groups are not

mutually exclusive of course. The developers and buyers could be the same people, and might also be users of the system. Any of the first three groups could also overlap with the penumbra. This identification assists in the clarification responsibilities and tensions in the software development enterprise. Developers want to make profits, but products need to be affordable by potential buyers. Buyers want products to be as inexpensive as possible, but they also want reliable products. Users also want the products to be reliable but they also desire them to be "friendly". The penumbra in general just want software which is not going to harm them or make their lives less pleasant. The basic tension is clearly between cost, on the one hand, and reliability and ease of use on the other. The more time spent developing a product, the better it can be, but the more expensive it will be. Typically there will be some trade-off between these two. The important question then is what criteria should be used to determine where the line will be drawn. Collins *et al.* suggest three:

(1) Do not increase harm to the least advantaged.

(2) Do not risk increasing harm in an already risky environment.

(3) Use the publicity test for difficult trade-offs.

The first criterion in most cases would refer to members of the penumbra, that is, those who are affected by the system but who may derive little or no benefit from it. The second criterion is unlikely to help much in identifying who will be harmed, but it does emphasise that more care needs to be taken in some situations than in others. The third criterion is explained thus:

> trade-offs between financial benefits to the one party and personal (i.e. non-financial) harm to another party should be made on the basis of a cost-benefit ratio that, if made public, would not outrage most members of the public (Collins *et al.,* 1995, p. 86).

This does something to help set the horizon. It seems intuitively plausible to assume that there would normally be more public disquiet, if not outrage, if financial benefits took precedence over direct and fairly immediate harm than over more vague harm in the future, or further removed from the system's direct effects. While these three criteria have some plausibility and usefulness, there are problems. Public outrage, for example, can be partly a result of the level of information given, so might not always identify the unethical, and the majority might not be outraged at some injustice done to a small minority, when it ought to be. A rights based alternative will sometimes be more appropriate than the consequentialism of Collins *et al.* (1995).

Identifying the stakeholders in the manner above certainly assists in making some of the issues clearer. There is, however, still some problem with the ideas of "distant stakeholders" and the "ethically sensitive horizon". The problem is not new, and any consequentialist theory of ethics must face it. Mill in *On Liberty* (1859), arguing that individuals should be free to do as they like with respect to actions which affect only themselves, acknowledged that most actions affect others as well. This is the same problem of scope as that which must be faced by software developers. Which effects should be taken into account and

which need not? Mill required an answer in order to set limits to the state's legitimate coercion of individuals. Software developers require it to make decisions regarding their work, where their own legitimate interests must be considered alongside the interests of those affected by their work. Mill's suggested solution was to say that people should be allowed to do as they want with respect to actions which affect themselves "directly and in the first instance", or primarily only affect themselves. Applying this to the software development environment, perhaps the horizon should be set so that all those affected by the software "directly and in the first instance", or who are primarily affected, have their interests taken into account. This would indicate that the interests of those using, say, an unfriendly ATM should be taken into account, but not the interests of their families on whom they vent their spleen. This is still vague, and it may not always draw the line in the right place, but it does give some guidance. It must be noted too that, in practice, there may be different stakeholders and horizons at each phase of the development process. The ethical guidelines must take account of this. For example, at the feasibility study phase there will be a host of social and environment concerns involving many groups in society, while at the logical system phase the focus will be more on management, technical staff, users and company shareholders.

de Raadt (1997), in a series of papers about multi-modal systems thinking, proposes a number of modes of action, analysed into a high to low sequence from wholly inspirational to wholly foundational. In order, these are: credal, ethical, juridical, aesthetic, economic, operational, social, epistemic, informatory, historical, psychic, biotic, physical, kinematic, spatial, numeric, logical. The professional codes of ethics seem to be concerned about the modes from logical up to and including economic, sometimes embracing aesthetic, often juridical (rules) but seldom venturing into the so-defined ethical (love) mode, which is of a higher order than the common view of ethics, and never into the credal (faith) mode. This "multi-modal systems approach" inherently embraces the "ethically sensitive horizon" and therefore promises to be a useful tool for future extensions to SSADM. At the same time, it invites us to rethink higher order professional standards.

Ethics and SSADM

As noted earlier, the ethical enhancement of SSADM is no small task, and hence this discussion is limited to outlining some methods for its achievement. It seems obvious that prescriptive rules for action are required, together with consideration of consequences. Such rules are needed if for no other reason than that in practice we cannot always consider all of an action's likely outcomes. Consequences must be taken into account because sometimes rules clash, and sometimes they need to be overridden for a greater good.

These comments partly mirror more fundamental discussions in ethics. Is something morally right or wrong because of its consequences, or because it follows or breaks some rule or violates some right? This is not the place to argue about whether ethics is essentially consequentialist, deontological, rights or

virtue based, or something else, important as these positions are for ethics in general. For everyday, practical purposes, we must frequently follow rules, albeit, defeasible ones; hence our reliance on a code of ethics in this paper. In everyday life, if we are to get anything done, we cannot examine in detail the consequences of all of our actions to see whether they cause, or are likely to cause, more or less harm than alternative actions. Nor can we in every case decide whether someone's rights have been violated. In important instances, and whenever possible, and particularly when rules clash, such thought should be given, but many decisions are made "on the run", with little time for reflection. This is not to suggest that ethical theory is not important in everyday life, it certainly is important, and it plays a central role in the formulation of codes of ethics such as the one used here.

Another basic issue in ethics is whether ethical principles are objective or relative. Our assumption, also noted earlier, is that they are objective in an important sense, and are not just matters of opinion. Nor is ethics merely relative to a culture. While many ethical practices are certainly related to cultures, it is much less clear that basic ethical principles are (see Weckert and Adeney, 1997, for discussion).

With this in mind it is important to address both the process and the product of SSADM. The product breakdown structure provides the impetus to address issues in a certain way. The completeness of these products needs to be considered given their powerful influence in the development. Consideration must also be given to how the systems developers should think when undertaking the various development tasks. The SSADM culture explicitly promotes technological and economic thinking, but not ethical thinking. The new version 4+ appears to be moving to a more balanced view with more emphasis on business orientation and the inclusion of user culture in the systems development template (Hall, 1995).

Changing the structure

It is now recognised that quality should permeate the whole of the ISs development process and not simply be considered at discrete points within the process. This is reflected in SSADM by the inclusion of a quality products set comprising a number of files that demonstrate that quality has been built into the system (CCTA, 1990). Product descriptions are part of this set. There are product descriptions for all the products specified in SSADM. The details include quality criteria against which the product can be checked. Product descriptions are used to monitor progress and success of the project. Including quality in each description ensures that it permeates the whole process and promotes a quality culture throughout the development team. Similarly ethical and societal consideration should permeate the whole process, and in many cases quality issues overlap with ethical ones, for example, in safety critical systems. It follows that each product description should include ethical and social criteria to promote this awareness and consideration. This might be

systematically addressed by using the ACS articles to form the ethical criteria for products within each of the core modules. This is illustrated in Table I which shows which articles apply to each of the modules.

The clauses below are selected from the ACS code of ethics as the most relevant to ethical issues in systems development and hence to extending a methodology such as SSADM. During the construction of Table I, there emerged a number of grey areas that depend upon point of view. The purpose here is not definitive accuracy, but a possible approach to adding checkpoint questions where they would be most effective. Future developments of the Table could well differ.

As an example, consider the application style guide, which is one of the products from stage 4, "technical environment" options within "logical systems specification". This sets the standards for the user environment within the particular project and includes ergonomic details and system based requirements. The ethical criteria for this product based on the appropriate ACS articles might be:

- Does the system protect and promote the health and safety of those affected by it? (4.1)
- Is the application style such that integrity and security of individuals is not compromised? (1.2)
- Am I happy that this product is suitable? (3.1)
- Has a walk-through of the application style taken place? (4.5)
- Is the working environment enhanced? (4.1, 4.5).

It is suggested that the top level of the product breakdown structure be extended to include ethics products. In this way ethical issues would be explicitly

	SSADM core modules				
ACS clauses	Feasibility study	Requirements analysis	Requirements specification	Logical systems specification	Physical design
1.2	X	X	X	X	X
1.3	X	X	X	X	X
1.5	X	X	X		
2.1		X	X		
2.4			X	X	
2.6	X	X			
3.1	X	X	X	X	
3.4	X				
3.5	X	X	X	X	X
4.1	X	X	X	X	X
4.2	X	X			
4.4	X				
4.5	X	X	X	X	X
6.1	X	X	X	X	X

Table I. Mapping ACS articles on to the core modules

considered within the SSADM method. These would comprise enhanced product descriptions, described above, plus appropriate outputs from the ethical instruments within SSADM. Three examples of these instruments are now briefly outlined.

Example instruments

The move from the current data flow diagram to the required logical data model is a demanding process from requirements analysis through requirements specification to logical systems specification. The process logically reduces the requirements definition by removing data redundancy and irrelevance. This is done from an economic and technological standpoint (and possibly legal). The question of whether items are redundant from ethical and societal standpoints needs also to be addressed. There is also the issue of ethical and societal verification of data item inclusion. Both might be addressed by developing a criterion reference model that enables the ethically and socially charged items to be systematically identified. A course of action can then be chosen based on balanced and comprehensive information.

An approach that could be incorporated both in the feasibility study and requirements analysis modules of SSADM is that suggested by Rogerson and Bynum (1995). They developed a four-perspectives model, based on Aristotle's model of ethical decision making, for the preliminary analysis and decision making at the beginning of any systems development project. This entails the integration of ethics, based upon the concepts and guidelines already outlined, in two different ways. A comprehensive analysis is undertaken from four perspectives; technical, economic, legal, and ethical. The four-perspectives analysis is the major input to the decision-making process which in turn results in a tentative action plan. It is essential that an ethical control loop exists providing the opportunity to review the decision made.

A feedback loop based on the outcome of the action is inappropriate as it will then be too late to influence that outcome. For this reason a feed forward loop is included which evaluates the tentative decision before the action takes place. This ethical evaluation provides a prediction of the likely outcome should action be undertaken. A predicted unethical outcome will result in a re-consideration of the project definition or the four-perspectives analysis or both. A series of analytical iterations could take place until the ethical feed forward analysis predicts an ethically acceptable outcome of the proposed practical action.

Given the scientific nature of SSADM that encourages rational logic, it is worth considering how ethical modelling might be incorporated. The SSADM process is based on logical decomposition that identifies the "best" way of satisfying a user requirement. It is oriented to the end product with the analytical journey simply being the means to that end result. A consequentialist approach is attractive in this context. Indeed it might be possible to develop an ethical modelling approach based on Bentham's hedonistic calculus. The seven aspects in the calculus used to compare relative outcomes can be reinterpreted in six terms more appropriate for ISs development as:

(1) the value of the benefit;
(2) the duration of the benefit;
(3) the degree of certainty in benefit realisation;
(4) when the benefit will be realised;
(5) whether the system will lead to secondary or indirect benefits; and
(6) the individuals, business and communities affected.

Note that the last two items require consideration of all stakeholders, alterations to community ethos, and effects upon environment. A significant reduction in complexity of these avenues could be achieved through the use of checklists (which as suggested, could be quite extensive, yet dynamic, being revised as professional wisdom grows).

This consequentialist approach has merit in that it links the ethical character of actions to their practical outcome. However, this can be extremely difficult. As White (1993) points out:

> a full account of an action's results means not only careful analysis of the immediate consequences to all involved and astute discernment of the quality and comparative value of the sensations experienced, but also an uncovering of the subtle, indirect, far-reaching, and long-term results as well.

This is particularly relevant for ISs where it is usually the case that systems interact resulting in long-term synergistic effects that are difficult to predict and measure. Yet we have to try.

Conclusion

This paper has discussed the ethical problems associated with ISs development. Little research has been undertaken in this area to date. It is has been demonstrated that approaches like SSADM do not encompass ethical considerations effectively. The ACS code has been used to identify where and how SSADM could be modified and some examples were used to illustrate this approach. This is only the beginning. Ethical enrichment of development methods must be carefully undertaken and validated through field work. Only then will there be an increase in ethically sensitive ISs.

References

Anderson , R.E., Johnson, D.G., Gotterbarn D. and Perrolle, J. (1993), "Using the new ACM Code of Ethics in decision making", *Communications of the ACM*, February, Vol. 36, pp. 98-107.

Avison, D.E. (1995) "What is IS? an inaugural lecture delivered at the University of Southampton", 3 November 1994.

Avison, D.E. and Wood-Harper, A.T. (1990), *Multiview: An Exploration in Information Systems Development*, Alfred Waller.

CCTA (1990), *SSADM Version 4, Reference Manual: Volume 4, Dictionary*, NCC Blackwell.

Checkland, P.B. (1981), S*ystems Thinking, Systems Practice*, Wiley.

Collins, W.R., Miller, K.W., Speilman, B.J. and Wherry, P. (1994), "How good is good enough? An ethical analysis of software construction and use", *Communications of the ACM*, Vol. 37, January, pp. 81-91.

de Raadt J.D.R. (1997), "Faith and the normative foundation of systems science", *Systems Practice*, Vol. 10, pp. 13-35.

Gotterbarn, D. (1992), "The use and abuse of computer ethics", in Bynum, T.W., Maner, W. and Fodor, J.L. (Eds), *Teaching Computer Ethics*, Research Center on Computing and Society, Southern Connecticut State University, pp. 73-83.

Hall, J. (1995), "How is SSADM4.2 different from SSADMV4?", *SSADM4+ for Academics*, CCTA & International SSADM User Group Workshop, Leeds Metropolitan University.

Harris, C.E. Jr, Prichard, M.S. and Rabin, M.J. (1997), *Practicing Engineering Ethics*, IEEE.

Jayaratna, N. (1994), *Understanding and Evaluating Methodologies*, McGraw-Hill.

Laudon, K.C. (1995), "Ethical concepts and information technology", *Communications of the ACM*, December, Vol. 38, pp. 33-9.

Maner, W. (1996), "Unique ethical problems in information technology", *Science and Engineering Ethics*, Vol. 2 No. 2, pp. 137-55.

Martin, M.W. and Schinzinger, R. (1996), *Ethics in Engineering*, McGraw-Hill.

Mason, R.O. (1995), "Applying ethics to information technology issues", *Communications of the ACM*, December, Vol. 38, pp. 55-7.

Mason, R.O., Mason, F.M. and Culnan, M.J. (1995), *Ethics of Information Management*, Sage Publications.

Mill, J.S. (1859), *On Liberty*, edition of Spitz, D. (1975), *John Stuart Mill,* On Liberty; *Annotated text, Sources and Background Criticism*, W.W. Norton & Co.

Moor, J. (1985), "What is computer ethics?", *Metaphilosophy*, Vol. 16 No. 4, pp. 266-79.

Mumford, E. (1983), *Designing Participatively*, Manchester Business School.

Rogerson, S. (1997), "Software project management ethics", in Myers, C., Hall, T. and Pitt, D. (Eds), *The Responsible Software Engineer*, Springer-Verlag, pp. 100-6.

Rogerson, S. and Bynum, T.W. (1995), "Towards ethically sensitive IS/IT projected related decision making", submitted to COOCS'95.

Rogerson, S. and Bynum, T.W. (1996), "Information ethics: the second generation", UK Academy for Information Systems Conference, UK.

Skidmore, S., Farmer, R. and Mills, G. (1994), *SSADM Version 4 Models and Methods*, 2nd ed., NCC Blackwell.

Walsham, G. (1996), "Ethical theory, codes of ethics and IS practice", *Information Systems Journal*, Vol. 6, pp. 69-81.

Weckert, J. and Adeney, D. (1997), *Computer and Information Ethics,* Greenwood Press.

White, J. (1993), *Money isn't God. So why is the Church worshipping it?*, Inter-Varsity Press, Downers Grove, IL.

Winter, M.C., Brown, D.H. and Checkland, P.B. (1995), "A role for soft systems methodology in information systems development", *European Journal of Information Systems*, Vol. 4 No. 3, pp. 130-42.

Wood-Harper, A.T., Corder, S., Wood, J.R.G. and Watson, H. (1996), "How we profess: the ethical systems analyst", *Communications of the ACM*, March, Vol. 39 No. 3, pp. 69-77.

Appendix – The Australian Computer Society Code of Ethics

Following is the Australian Computer Society (ACS) Code of Ethics, extracted from the ACS WWW site http://www.acs.org.au/ as it existed in January 1998. Apart from omission of the initial introduction and the introductory paragraph to the Standard of Conduct section, the Code is reproduced in full. The bracketed words have been added.

Code of Ethics

I will act with professional responsibility and integrity in my dealing with clients, employers, employees, students and the community generally. By this I mean:

1. [Priorities] I will serve the interests of my clients and employers, my employees and students, and the community generally, as matters of no less priority than the interests of myself or my colleagues.
2. [Competence] I will work competently and diligently for my clients and employers.
3. [Honesty] I will be honest in my representations of skills, knowledge, services and products.
4. [Social implications] I will strive to enhance the quality of life of those affected by my work.
5. [Professional development] I will enhance my own professional development, and that of my colleagues, employees and students.
6. [Computing profession] I will enhance the integrity of the Computing Profession and the respect of its members for each other.

Standard of Conduct

1. *Priorities*
   I will serve the interests of my clients and employers, my employees and students, and the community generally, as matters of no less priority than the interests of myself or my colleagues.

1.1 I will endeavour to preserve continuity of computing services and information flow in my care.

1.2 I will endeavour to preserve the integrity and security of others' information.

1.3 I will respect the proprietary nature of others' information.

1.4 (missing in original).

1.5 I will advise my client or employer of any potential conflicts of interest between my assignment and legal or other accepted community requirements.

1.6 I will advise my clients and employers as soon as possible of any conflicts of interest or conscientious objections which face me in connection with my work.

2. *Competence*
   I will work competently and diligently for my clients and employers.

2.1 I will endeavour to provide products and services which match the operational and financial needs of my clients and employers.

2.2. I will give value for money in the services and products I supply.

2.3. I will make myself aware of relevant standards, and act accordingly.

2.4. I will respect and protect my clients' and employers' proprietary interests.

2.5. I will accept responsibility for my work.

2.6. I will advise my clients and employers when I believe a proposed project is not in their best interests.

2.7. I will go beyond my brief, if necessary, in order to act professionally.

3. *Honesty*
   I will be honest in my representation of skills, knowledge, services and products.

3.1. I will not knowingly mislead a client or potential client as to the suitability of a product or service.

3.2. I will not misrepresent my skills or knowledge.

3.3. I will give opinions which are as far as possible unbiased and objective.

3.4. I will give realistic estimates for projects under my control.

3.5. I will qualify professional opinions which I know are based on limited knowledge experience.

3.6. I will give credit for work done by others where credit is due.

4. *Social implications*
I will strive to enhance the quality of life of those affected by my work.

4.1. I will protect and promote the health and safety of those affected by my work.

4.2. I will consider and respect people's privacy which might be affected by my work.

4.3. I will respect my employees and refrain from treating them unfairly.

4.4. I will endeavour to understand, and give due regard to, the perceptions of those affected by my work, whether or not I agree with those perceptions.

4.5. I will attempt to increase the feelings of personal satisfaction, competence, and control of those affected by my work.

4.6. I will not require, or attempt to influence, any person to take any action which would involve a breach of this Code.

5. *Professional development*
I will enhance my own professional development, and that of my colleagues, employees and students.

5.1. I will continue to upgrade my knowledge and skills.

5.2. I will increase my awareness of issues affecting the computing profession and its relationship with the community.

5.3. I will encourage my colleagues, employees and students to continue their own professional development.

6. *Computing profession*
I will enhance the integrity of the Computing Profession and the respect of its members for each other.

6.1. I will respect, and seek when necessary, the professional opinions of colleagues in areas of competence.

6.2. I will not knowingly engage in, or be associated with, dishonest or fraudulent practices.

6.3. I will not attempt to enhance my own reputation at the expense of another's reputation.

6.4. I will co-operate in advancing information processing by communication with other professionals, students and the public, and by contributing to the efforts of professional and scientific societies and schools.

6.5. I will distance myself professionally from someone whose membership of the ACS has been terminated because of unethical behaviour or unsatisfactory conduct.

6.6. I will take appropriate action if I discover a member, or a person who could potentially be a member, of the ACS engaging in unethical behaviour.

6.7. I will seek advice from the ACS when faced with an ethical dilemma I am unable to resolve by myself.

6.8. I will do what I can to ensure that the corporate actions of the ACS are in accordance with this Code.

6.9. I acknowledge my debt to the Computing Profession and in return will protect and promote professionalism in computing.