# The role of visibility in systems

**Richard Veryard**

First published in Human Systems Management 6 (1986) pp 167-175

*Keywords*

Design, Participation, Modelling, Systems, Visibility, Open Control.

*Abstract*

Modern man is surrounded by procedural systems, often highly formalized and at least partially computerized. Systems are by their nature intangible and not directly visible. Indirect representation (via system models) is required to make a system visible, so that it can be properly understood, shaped and controlled.

*Acknowledgements*

In draft form, this paper has been delivered at an internal Technical Conversation for Data Logic in 1984, and circulated as a working paper dated 1985. Love Bhabuta read (and decimated) the whole paper and Colin Tully pointed out the significance of the audit trail. Thanks are also due to other colleagues (and an anonymous referee) who provided feedback.

*Biography (1986)*

Richard Veryard is a computer systems analyst, specializing in modelling techniques and formal methodologies. He is the author of *Pragmatic Data Analysis* (1984) and a number of articles, on both technical and social aspects of information technology. He studied mathematics, philosophy and computing science at the universities of Oxford and London, and has worked for computer systems houses in London, Hamburg and Dublin. He is a member of the British Computer Society's Working Party on Information Systems Analysis and Design.

*Dedication*

To the memory of Dora Russell.

*Current Author Details (1999)*

email: richard@veryard.com

web: http://www.veryard.com

# Introduction

## *Steam Power*

This paper is about the relationship between systems (regarded as a branch of technology) and people.

I can trace an uncertainty about technology back to an incident in my childhood. Because I was a boy it was natural that I should be given a train set. (For some reason, it was acceptable in those days for middle-class parents to instil in their sons the fantasy of being an engine-driver.) What was unnatural about it was that the locomotives were based on a deliberate fraud. Inside a basic shape dating from the nineteenth century and modelled on one of the classic steam trains, there was a simple electric motor, emitting a strange short-circuit smell. More sophisticated electric toy trains even had the facility to emit phoney puffs of smoke.

The first real steam train I ever saw working was a much larger model, pulling open carriages of ice-cream cheeked children. The engine-driver sat astride the engine, collecting fares from the parents. I could see no difference between that and a seaside donkey. Perhaps it is the fate of out-of-date inventions to revert to the status of amusing gimmicks. Anyway I was for a long time unsympathetic to the romance of steam.

The steam train aroused passions that its successors never have. This was at least partly due to the fact that the working parts were visible. People watched in awe as the power of the steam was transmitted to the wheels. But the workings of a diesel-electric train are concealed behind a modesty curtain of metal.

It appears to be a general trend of technological progress that the workings of machines are increasingly hidden from the user, perhaps because of size (one can't watch quartz as one can watch clockwork), perhaps because of design ideology (so-called functional design, a preference for smooth unfussy surfaces). Industrial design expresses a set of values. One generation was impressed by power, the next generation by speed, and the following one by efficiency. Our fathers were impressed by bigness, their children by smallness.

'From the late 1920s to the mid-1930s, industrial designers found themselves faced with machinery whose function could no longer be derived visually through common sense or logic, and began encasing these constructs in so-called shrouds. A method or work that no longer made sense was tucked away behind metal or wooden covers. This shrouding performed a secondary and equally important job - it protected delicate engine parts from dust and other contaminations. In the 1970s, chip technology, minicomputers, and microprocessors emerged as small, black-box concepts, which many users did not understand. Linked to this, extreme microminiaturization and a wealth of new materials have made it possible for designers and manufacturers to "make anything look like anything we want, make anything look like something else."' [Papanek, p71]

Another example can be found among the architectural wars of the 1980s. Architects of the 'post-modern' school react against the allegedly stale and anonymous glass boxes of the modernists by painting ironic jokes on the sides of the same boxes, or bolting incongruous elements together. Architects of the 'hi-tech' school, on the other hand, turn their buildings inside out, to make the structure visible rather than witty.

## *Thesis*

So this paper is about the visibility of systems. Systems are more difficult to make visible than the machinery described in the previous section. Some of the reasons for this will become clear during the paper. (Because of the differences between systems and machines, the concept of

0167-2533/86/$3.50  

visibility is metaphorical. We need some idea how far the metaphor can be stretched.) Visibility is a useful, often essential property of human activity systems. Business information and computer systems provide some examples; it is also enlightening to consider socio-political systems such as Criminal Justice and Parliamentary Democracy.

The argument runs as follows. First I try to delineate what I mean by visibility, and justify its importance. Then I discuss some aspects of systems and system models. Then I describe some mechanisms for providing visibility in systems. I will indicate how this brings not only social benefits but also enhances the effectiveness and efficiency of the system itself. I end with some unanswered questions, as possible research ideas.

## What is visibility?

### *Definition and examples*

Visibility is an important property of human activity systems. The intentions, workings and structure of any system should not be hidden away but be open to view. The system should not only work, it should be seen to work. There should be a manifest linking between cause and effect.

The alternative to visibility is mystery; if the system produces results by some magical process, then it becomes the property of an initiated elite, which is socially undesirable. The alternative to visibility is black-box systems. These arouse suspicion, dislike and distrust, because the uninitiated fear being manipulated. Black boxes are less reliable, because it is more difficult to check their workings.

Visibility therefore ranks alongside such other system criteria as effectiveness, efficiency, reliability, stability and measurability. These are the (interdependent) criteria by which systems of all kinds are evaluated, selected, designed and improved. The next sections discuss visibility in relation to other system properties.

Elegant design doesn't hide workings; form should follow function. In problem-solving, making the structure of the problem explicit allows the structure of the solution to be properly related to the structure of the problem. This is a familiar theme in knowledge engineering, that an 'expert' system should explain its reasoning. For example, a computerized medical diagnosis system should justify its results, in terms that will be understood at least by doctors, if not by patients. This provides a double reassurance to the user: that the right question has been answered using the right methods. There is a parallel with school mathematics: bright children always find the most difficult requirement to be that of reconstructing their calculations on paper for the benefit of the teacher.

'Visualization helps people's recognition and social acceptance of intangible objects. In Japan, visualization has helped social acceptance of many issues. Much of the widely envied improvement in quality control came from making the intangible concept "quality" explicit and visible.' [Negishi]

In political theory, we can recognize similar demands. Justice or democracy must be seen to be done. In other words, the legal system and the political system must both contain some degree of visibility.

### *Visibility versus simplicity*

Simplicity is an aid to visibility and comprehensibility.

'Simplification leads to elegance. Simplification also implies common sense, a quality that seems to get rarer as the world becomes bureaucratized. Elegance, as I use the word here, means that a problem has been solved directly, un-self-consciously, with minimal effort and in such a way that the object or product can be understood easily and is a good fit with its environment.' [Papanek, p 80]

'The simpler a data system is, the easier it will be to entertain a clear alertness of its omissions. A complicated data system might give users the false impression of covering everything worth knowing.' [Nissen]

Simplicity may require reducing the number of constructs, or relying on popular common sense instead of theoretical sophistication.

'A conception of justice is to be the public basis of the terms of social cooperation. Since common understanding necessitates certain bounds on the complexity of principles, there may likewise be limits on the use of theoretical knowledge (viz. in establishing principles of fairness).' [Rawls, p 142]

But simplicity or design elegance is only conducive to visibility; it is not always equivalent. An over-simple design may not be expressive enough. An un-self-conscious design may be insufficiently rational. [Simon]

### *Visibility versus transparency*

Many writers use the term 'transparency'. I avoid this for two reasons. Firstly it is ambiguous. Sometimes it means that the user sees *into* the system and understands its workings, which is roughly equivalent to what I am calling visibility. Sometimes it means that the user sees *through* the system and ignores its workings, which is roughly the opposite. Secondly, transparency is a property of an object alone; visibility is a property of an object in relation to an observer. In other words, visibility implies choice: the user *can* see the workings, but needn't. Transparency is used of glass or of television screens: one is unaware of the distortion.

### *Personal note*

The notion that a democratic system should be open and visible is hardly controversial. Even elitist politicians pay lip-service to it in principle, while thickly hedging it with brambles and 'safeguards' in practice.

However, *ad hominem* tactics are available against me. Sherry Turkle, in her analysis of the psychology of the computer hacker, explains why people who work with computers tend to hold such beliefs in 'pleasing populism'. 'Wayne', for example, is quoted as follows:

'Politics is a system, complex to be sure, but a system all the same. If people understand something as complicated as a computer, they will demand greater understanding of other things.' [Turkle, p 178]

'Wayne' is then dismissed as a social critic because he is a hacker. (I.e. a member of a subcult of computer fanatics.) He therefore (a) can be regarded as having taken refuge from normal society, and (b) underrates the complexity of society vis-à-vis the computer. The idea that political systems *ought* to be visible is presented as a psychological phenomenon, rather than as a serious moral.

I agree with 'Wayne'. I admit that my analysis of democracy is at least partly based on my experience as a computer systems analyst. I am not aware of confusing computers with people or computer systems with politics. (My unawareness perhaps confirms my inner confusion!) I believe it to be legitimate to base hypotheses upon analogy, generalization or abstraction from experience. (I challenge Turkle to dismiss or explain away this belief on the basis of my biography.)

## Importance of visibility

### *General comments*

The point of enhancing visibility is to improve the relationship between systems and people. What we are is determined by the tools we use (or what Marx called the ‘means of production’). What we are is determined by the symbols we use to represent the world, and our place in it (for Wittgenstein equated language with tools). Therefore in order to know ourselves, we have to be conscious of our machines, our systems, our language, how they work, how they work on us. Visibility means we don’t only perceive how the tool works (which means representing it somehow) but also how the representation works (which leads us into highly abstract and potentially circular discourse).

The design of a system, or the solving of a technical or technological problem in a new way, can be regarded at two levels. Firstly, it is (or should be) a specific improvement in the organization and working environment of a group of people. Secondly, it is a contribution, perhaps small but not insignificant, to general technological progress.

It matters both as professional ethics and as human ethics. Dehumanizing systems may be counter-productive; they are certainly morally wrong. And visibility is, in a manner of speaking, itself a tool: a tool for rehumanizing systems.

## *Open control*

The point of visibility is partly to improve people’s control over systems. How do staff control the business systems they work within? How do electors control a government? How does society control the judiciary? And how do system models help? Control has the following elements:

- setting expectations (standards, targets, etc.);
- measuring achievement and performance, and comparing with the expectations;
- taking corrective action where needed.

For example, let us consider political control over the Criminal Justice system. In the United Kingdom, this has the following elements:

- standard procedures are set by tradition and judicial whim (usually known as ‘precedent’), with a parliamentary override;
- performance is measured by the performers themselves (e.g. police, lawyers, etc.), with occasional interest shown by press and TV. Injustice should not be seen to be done, thus defendants should not appear bruised, confessions should seem plausible and fair, etc.;
- if enough public outcry is made or threatened, the Home Secretary can demand a retrial or pardon in a specific case.

In order for the ‘actors’ within a system to exercise control over it, the mechanisms of the system must be visible to them. A model provides this element of ‘democracy’, by explaining the system (descriptive modelling), what the system ought to do (prescriptive modelling) and what the actors ought to do (action modelling). Thus it should provide the following:

- more realistic expectations;
- a visible and comprehensible metric for evaluating achievement and performance;
- easier or better choice of appropriate action, to deal with any unsatisfactory situation.

Possible actions are as follows:

- changes in own behaviour or attitudes (i.e. fitting in with the system as it is, or putting up with its imperfections);
- carry out or demand minor changes to the system (i.e. ‘tuning’ the system or maintenance - this may involve a mere change in input parameters);
- carry out or demand wholescale changes to the system, or the implementation of a new system.

Retaining control over this last course of action may require some form of participation in the system design, which will be discussed in the next section. The importance of systems design follows from Mumford's analysis of technology.

'Choice, freedom, esthetic evaluation, are transferred from the process as a whole, where it may take place at any moment, to the initial stage of design. Once choice is made here, any further human interference, any effort to leave the human imprint, can only give impurity to the form and defeat the final result.' [Mumford, p 82]

## *Participative design*

Traditionally, systems design (and, for that matter, any problem-solving in management services) has been a battle-ground between professionals and amateurs. The professionals are experts in systems, organization and methods (O&M), computing, etc. The amateurs are the so-called end-users, who will be making regular use of systems when they are implemented. These users may have expertise in some business function, but are assumed to have no expertise in systems, O&M, computing, etc.

It is usually stated that the users would be unable to produce a good systems design unaided, at least if the system is to be at all complex. (This is because the professionals set the criteria by which the quality of a system design is judged, based usually on technical aspects.) So major systems design efforts are usually assigned to professionals, with or without the participation of users.

The task of systems design is itself complex. It has four inputs:

1. Subjective facts (e.g. targets, priorities, constraints, etc., obtained by interviewing key users and managers).
2. Objective facts (e.g. volumes and frequencies, obtained by direct observation and measurement).
3. Specific ideas (possible solutions to specific design problems, either invented or adopted/adapted from elsewhere).
4. Design methods and skills (based either on a formal methodology in which the professionals have been trained [Veryard 1985], or on their past experience and present intuition).

There are four levels of possible participation by the user, according to which of these four inputs s/he has control or influence over. Minimal participation occurs when the users are interviewed, acting as a passive source of subjective facts. Greater participation occurs when users can draw attention to particular facts, or hide facts believed irrelevant or harmful. (Note that facts are only effectively hidden if there is a consensus among users.) Still greater participation occurs when users are given the opportunity to describe possible systems, refer to similar systems elsewhere, or amend proposed system designs.

Full participation requires the user to be active in the design. This means s/he must understand the design *process* as well as the end-*result*. Visibility for full participation therefore has three aspects: three kinds of model are required:

- of the original or agreed system requirements
- of the designed system
- of the design process, i.e. a formal design methodology

It is not always clear how these three kinds of model are related. In information systems, the exact relationship between the design as process and the design as product remains speculative, pending further research.

## *Selfconsciousness*

Visibility in one place may improve understanding elsewhere. Alexander points out that the use of good models affects the users. (He is talking specifically about design problems.)

'The use of logical structures ... brings with it the loss of innocence. A logical picture is easier to criticize than a vague picture since the assumptions it is based on are brought out into the open. ... Once what we do intuitively can be described and compared with nonintuitive ways of doing the same things, we cannot go on accepting the intuitive method innocently.' [Alexander 1964, p 8]

Of course, 'bad' models may also affect people, but to different degrees and in different directions. There is a discussion of this point in [Nissen].

Thus we have to consider the impact of visibility not only as a local property of a single system, but also as a general heuristic device, as a tool for organizational and individual learning.

## Systems

### *Machines and systems*

It may be useful here to clarify some concepts. We start with some simple definitions. A *machine* is a physical tool or lump of machinery. A *system* is a connected group of mechanisms, activities, procedures, etc. Architects often regard a building as a machine; the occupancy and use of the building form a system. In computing, distinctions are drawn between hardware and software. In anthropological terms, the machine is 'dead', the system is 'alive'. (No value judgement implied, despite Alexander [1979].)

Visibility, which has a literal meaning for machines, has a metaphorical meaning for systems. In fact, discussion of systems generally is confused by having to rely on metaphor. This is partly due to a weakness in the English language for naming abstract entities. (This has infected writers in other languages, including German, who do have adequate mechanisms, but don't always use them.)

The following confusions will be ignored for the remainder of the paper:

1. A machine is generally named and described in terms of its *use*, but the use-of-machine involves human activity and is therefore a system.
2. There is a common figure of speech known as synechdoche, in which the container is confused with the thing contained. Whether the system contains the machine, or the machine the system, is a matter of perspective; it is certainly common for people to talk of systems as if they were identical with the physical machine.
3. Systems are often defined by location rather than by purpose and function. ('The Head Office system', etc.) In London, names of buildings and districts are used to refer elliptically to systems: Stock Exchange, Old Bailey, Bush House, Whitehall, Wimbledon.
4. In computing, the division between hardware and software is somewhat arbitrary. The designer can choose whether to implement a given mechanism by hardware or leave it to be programmed. (This is similar to the choice an architect has, to divide space with fixed partitions or to provide moveable screens.) The concept of 'abstract machine' has been introduced, allowing us to talk of a computer as if certain mechanisms were part of the hardware, although they are in fact part of the 'systems' software.

### *How do systems work - individuals and types*

A system deals with people or objects or both. Formal systems operate by classifying individuals into types. In order to illustrate this, let us consider two systems, a commercial sales order system and a criminal justice system. Each of these two systems involves people, who play particular roles. People, and other entities, are classified initially by role. On the one hand: customer, salesman, manufacturer, product, hireling. On the other hand: defendant, witness, policeman, juryman, judge. In formal systems there tends to be confusion when a single person attempts to play several roles at once, and often rules and procedures are invoked to prevent this.

Where each individual has been classified into a specific role, the system needs enough information to determine exactly what is to be done to or by the individual. The relevant attributes of a customer may be his credit limit, whether he wants overseas delivery, etc. The relevant attributes of a defendant may be the charge against him, previous convictions, legal aid entitlement, etc.

Since the system is usually processing several individuals in parallel, it needs to remember what stage each has reached. A sales order may be packed but not yet delivered, or delivered but not yet paid for. A defendant may be released on bail in order to prepare his case, or sentenced and awaiting appeal.

The system thus classifies by role-type, treatment-type and stage-type. (This also applies to non-human entities such as Delivery Note or Fingerprint.) The system deals with all entities according to their classification: entities that are classified alike will be treated alike.

For the system to be visible, it is necessary for this classification to be explicit and open. The procedures for deciding need not be laid down in advance, nor be crudely mechanical: the discount rate or length of sentence can be left to the discretion of the salesman or judge, within limits. But to avoid any suspicion of corruption, the decision criteria must be revealed and explained. And it must be clear which attributes of the individual may be relevant, and which not.

Party democracy has an extremely simple classification of individuals into types: Labour/Conservative, Christian/Social, Democrat/Republican. The individual oneself decides how to convert one's own political opinions (which may be very complex and sophisticated) into a simple vote.

## Models

### *Indirect visibility - through representation*

Models are used to provide visibility indirectly, or metaphorically.

A simple machine or system may be made visible simply by removing the skin (or using a glass skin), so that the pistons and cogwheels can be seen moving. A complex machine or system may remain incomprehensible, even after removing the skin. Consider the human body. An untrained observer would not be able to identify the innards of a vivisected person. The usual method of training surgeons and scientists is to first provide them with labelled line drawings. After learning the structure of nerves, tendons and vessels from such drawings, they are then ready to learn how these drawings correspond to reality, which they do by dissecting dead bodies. The drawing acts as a simplified, two-dimensional model of the anatomy, allowing the student to ignore the 'too too solid flesh' and focus on the relevant bits.

Note that each drawing is a partial model of reality: one depicts the nerves, another depicts the blood vessels, and so on. It is not necessary, and will probably be ineffective, to include everything in a single model.

These drawings are not only used for training doctors. A practising doctor may use similar drawings to explain a diagnosis to a patient, or to offer a choice of alternative treatments (perhaps involving different levels of risk, pain or inconvenience).

Thus a drawing provides visibility (in the sense defined above) not by direct looking but by indirect representation. It defines the scope of the system, probably better than words can.

And perhaps the first requirement of system visibility is the very existence of the system itself, which may need to be demonstrated. Other people may not recognize the same system (with the same purpose, scope, etc.) or may not recognize any system. Drawing something doesn't prove it exists, but it helps focus the argument.

### *Nature and type of models*

A model is a representation of reality. Alternatively, it may be a representation of projected reality, as when an architect makes a three-dimensional scale model of a future building.

Models can be two- or three-dimensional. (Models exist in physical space; they can represent a greater number of dimensions than they occupy.) They can be any combination of geometrical, mathematical, logical or verbal symbols. A model is in a *modelling language*; this defines which symbols are to be meaningful, and lays down rules for interpreting the model.

Geographical maps are models. Most maps are 'to scale', which implies that distances on the map are meaningful. (Apart from maps of the whole world, where the problems of mapping a sphere onto a flat surface are unavoidable, distances on the map are usually proportional to the distances on the ground.) But in some maps, the distances are meaningless. A common example is to be found in the underground railway maps in most cities; because people are more interested in the number of stations and the interconnections between lines, and more interested in journey times than actual distances, a simplified map with straight lines and clean curves is easier to read.

The model must share some properties with its subject, the reality being modelled. It cannot share all properties, because then the model would be indistinguishable from the subject. It may, for example, different in size or material, or rely on graphical and verbal symbols. (A town map is rich in such symbolism.) The conventions within which a map is drawn, or any other model made, dictate which properties must be shared between the model and its subject. (E.g., scale, topology.) And so they dictate what conclusions about the subject can validly be drawn from the model.

So a model of a system makes it visible by separating the relevant attributes of an entity (role-type, treatment-type, stage-type) from the irrelevant ones.

So far we have considered systems without reference to the observers. We must consider how the system is interpreted by observers and by actors within the system. In the rest of this section, I outline some useful types of system models and representations. These models are at present restricted to particular classes of observer/ actors, but have scope for wider use. A useful feature of all these models is that they can be displayed on a computer screen.

### *Structured computer design models*

Software and procedures are not directly displayable. Most people will find program code incomprehensible. Furthermore, it needs to be discussed before it exists. Computer programmers and systems designers have always used diagrams to support the formal (and usually incomprehensible) specification and documentation of computer software.

Such diagramming techniques form a modelling language; such a language should be independent both of the particular application and of the computer technology, to allow ease of communication and common understanding. (As technology becomes more complex, the limiting assumptions in a given modelling language become apparent, and a more sophisticated language must be developed; thus the language must be 'up-to-date' as well. Once it was the flowchart that was fashionable;

nowadays the models depict the data structure and data flow of a required computer system.) For details of a widely-used such modelling language, see [Veryard 1984].

These models are used to design specially-tailored software, or to select a suitable off-the-peg package. They are also sued for the long-term strategic planning of information systems within an organization. Their most important use is as a communications tool, for negotiating the scope and requirements of a projected computer system between the technical designers and the various groups of computer users.

### *Accounting and audit trails*

An audit trail is a device built into a financial accounting system, to enable the accountant or auditor to trace financial transactions backwards and forwards through the system. It therefore enables the auditor to understand how the system works, well enough at least to express confidence in the reliability of the accounts. Such models provide an alternative path to the visibility of computer systems.

### *Cognitive mapping*

Decision-support tools have developed from Operational Research techniques, allowing complex inter-relationships to be mapped and depicted on a computer screen. For example, the COPE system developed at Bath [Eden, Freyenfeld]. Such techniques allow a manager to understand how his department works and how it interacts with its environment. Thus he can be guided towards appropriate action. There is scope for wider application of such tools.

## Practical benefits

Enhancing the visibility of systems may be a ‘Good Thing’ in human terms. But arguments from human needs, human values, human progress, etc., do not persuade everyone. Furthermore, because the benefits are intangible, it is difficult to decide the amount of effort to be devoted to achieving such benefits. Therefore it will be useful to obtain practical gains from visibility. In general, the advantages to the enterprise of visibility in its systems are to do with maximizing the intellectual input from each employee.

Making a system more visible will often make it more effective or efficient. For example, in a computer system, inaccurate data are more likely to be corrected by staff who understand the significance of the data in the system, than by staff for whom the data are meaningless figures. And design faults in the system will be diagnosed more quickly if the users recognize the symptoms and can supply the programmers with complete details of the bug.

The effort of making a system more visible will often improve the simplicity of the system, and result in a better-documented system. Such systems are usually cheaper to maintain.

However, there will often be situations where such efficiency arguments do not work in the desired direction. Gramsci [p308ff] refers to the typesetter who may work faster if he doesn’t understand the text; similar arguments can be applied to data entry clerks. And for some organizations, security and secrecy may override efficiency: in order to minimize the consequences of any employee joining the opposition, no employee is allowed to know or understand very much.

Thus there will often be practical arguments in addition to the moral, human arguments. But sometimes we will only have the moral arguments.

## Conclusion and future research

'The great problem of our time is to restore modern man's balance and wholeness: to give him the capacity to command the machines he has created instead of becoming their helpless accomplice and passive victim.' [Mumford, p 11]

In this paper, I have tried to indicate some ways of making business information and computer systems more visible. The models that are available go some way towards the objective we started with, which was to act as a framework for 'democratic' control. However, much more work needs to be done, to develop better modelling techniques and languages, and to use the existing ones better.

More questions have been raised than answered, and I make no apology for this. I should like to mention some areas deserving further thought and research:

- Is visibility an important property of all man-made or designed objects? Scruton [p 12] asserts that there is a contradiction between visibility of the surface and visibility of the structure, and that therefore scientific explanation may create as much mystery as it dispels. This assertion requires some critical examination.
- How does a model-based concept of visibility relate to Boland's action-based concept of information systems?
- How far can or should visibility be taken? Is there an optimal quantity? What properties must a system model have to help and not hinder visibility?
- Nissen establishes a dialectic between information systems and people; he asks how the people can be made visible to the system. How can this idea of visibility be related (if at all) to the idea of visibility expounded in this paper?
- How does the topic of visibility relate to wider problems of technological change and control? What can we learn from computers in the improvement of our socio-political systems?
- What are the values expressed by a computer system? What are the implications of hiding the inner workings from the user? Can the concept of visibility help us towards a design aesthetic of human activity systems? If a system is visible, is it then easier to judge how 'good' or 'elegant' a design it is?

My main reason for writing the paper has been to open a debate about a property of human activity systems that I feel has been overlooked. I want to encourage both systems designers and researchers to take visibility seriously as a system property. I hope that other minds will be stimulated to take these ideas further.

## References

Christopher **Alexander**, Notes on the Synthesis of Form (Harvard University Press, Cambridge MA, 1964).

Christopher **Alexander**, The Timeless Way of Building (Oxford University Press, New York, 1979).

Richard J. **Boland**, Control, causality and information system requirements. Accounting, Organizations and Society 4 (4) (1979) pp 259-272.

Colin **Eden** et al, Messing about in problems (Pergamon Press, Oxford, 1983).

W.A. **Freyenfeld**, Decision-Support Systems (NCC, Manchester, 1984).

Anton **Gramsci**, Selections from the Prison Notebooks (Lawrence & Wishart, London 1971).

J.R. **Lucas**, Democracy and Participation, (Penguin, Harmondsworth, 1976).

Lewis **Mumford**, Art and technics (Columbia University Press, New York, 1952).

Hirokazu **Negishi**, Tentative classification of global software. Behaviour and Information Technology 4 (2) (1985) pp 163-170.

H-E. **Nissen**, When people design an information system, then the information system designs people, in John **Hawgood** (ed) Evolutionary information systems (North-Holland, Amsterdam, 1982).

Victor **Papanek**, Design for human scale (Van Nostrand Reinhold, New York, 1983).

John **Rawls**, A theory of justice (Clarendon Press, Oxford, 1972).

Roger **Scruton**, Sexual desire (Weidenfeld & Nicholson, London, 1986).

Herbert **Simon**, The sciences of the artificial (MIT Press, Cambridge MA, 1981).

David **Stodolsky**, Information systems for self-management. Human Systems Management 5, (1985) pp 39-45.

Sherry **Turkle**, The second self: Computers and the human spirit (Grenada Publishing, London, 1984).

Richard **Veryard**, Pragmatic data analysis (Blackwell Scientific Publications, Oxford, 1984).

Richard **Veryard**, What are methodologies good for? Data Processing (July/Aug 1985) pp 9-12.

# Retrospective Notes (1999)

In the mid 1980s, I wrote a batch of articles, covering a range of related topics. Some were published in newspapers and magazines (The Times, The Financial Times, New Society), some in computer journals and magazines (Computer Weekly, Data Processing, Information and Software Technology), and some in system journals (Journal of Applied Systems Analysis, Human Systems Management). I also wrote a number of book reviews, mostly for Information and Software Technology.

That body of work did two things. It established a set of themes that have remained important for me, and it also hinted at some further themes whose form and significance have only emerged more recently.

These articles are the work of a young man, with the brash and sometimes careless optimism of youth. Although I now find some of the analysis simplistic and naïve, and I would certainly try to express the positions and opinions of these articles with greater precision and care, I hope I have retained the spirit of them.

❋❋❋

If I were to rewrite this article today, I should certainly want to refine the notion of technology, the notion of system, and the notion of model.

Since writing the article, my understanding of technology and visibility has improved greatly from reading Heidegger, Borgmann and Latour. I believe that Heidegger's notion of Unconcealment (Unverborgenheit) is a fair reflection of the notion of visibility I was groping towards, and I should want to reword my definitions and analysis to align more closely with Heidegger.

In the article, I express some reservations about distinguishing 'machine' and 'system' in such simple terms. I now doubt whether the distinction is necessary at all here, given that apparently tangible machines are already loaded with so many intangible associations. In the article, I use the term "system" to refer to a particular kind of technological or social artefact. Visibility is then primarily regarded as a property of such a system. Although I state that visibility is actually to be regarded as a context-sensitive property, in relation to an observer, this point is made in passing and is not carried through the article. I am no longer sure that such systems are significantly different from any other kind of artefact, and I should now put try to analyse visibility more consistently as a property of an artefact-in-use, which demands attention to the relationship between the artefact and a user, or community of users. This also demands attention to the purposes and values of the user.

Some of the examples of metaphor provided in the article might more properly be called metonymy. Reading Lacan has alerted me to the potential importance of this distinction, but I haven't yet explored its relevance to technology.

I would also want to explore the apparent contradiction between visibility and concealment. In an extreme case, a conjuror makes something visible ("nothing up my sleeve") in order to make something else invisible. Perhaps the same phenomenon occurs always, although not always under such conscious human control. If total visibility is impossible, striving towards more visibility needs to be qualified by an appreciation of the potential loss of visibility elsewhere. This qualification is attributed to Scruton in this article, but I am now aware of many other sources for this line of enquiry.

As for the notion of model expressed in the paper, I now find this almost embarrassingly simplistic. I should now need to make explicit the fact that every model itself has a purpose and perspective, and is not merely an objective representation of “reality”. To argue for the social value of modelling, I should need to call upon a body of social theory, probably starting with Habermas.

This article does not explore the political and ethical issues about the legitimate stakeholders for visibility - who has the right to demand visibility over a particular system. In regulated industries, corporate systems should be (at least to some extent) visible to the industry regulator. Journalists, on behalf of a supposedly information-hungry public, demand visibility of many systems whose owners would prefer to remain private. In international politics, justice and democracy must be visible, not only to the people within a country, but also to powerful foreigners.

In information systems, the relationship between the design process and the design end-product has been studied extensively during the past 15 years, and ought therefore to be better understood. Quality assurance, software metrics and software process improvement all rely on the visibility of a process.

Technological notions of encapsulation (black-box) and openness have developed greatly in the software industry in the past 15 years, as manifested by Component-Based Development (CBD) and Open Distributed Processing (ODP). These apparently contradictory developments have generated a wealth of potential examples, whose significance can be interpreted in multiple ways. There is a recent trend towards visible program code, as for example with the operating system Linux, which is supposedly more reliable and robust as a result of its visibility.

Finally, the references to my own work could be greatly expanded and updated, as I have explored some of these themes in greater detail in the intervening period.

 