# A THEORETICAL BASIS FOR IS? THE CONTRIBUTION OF ANT

Jim Underwood
University of Technology
Sydney, Australia
jim@it.uts.edu.au

## ABSTRACT

Representation is a key issue of IS design and operation that is often ignored. Actor-network theory (ANT), a semiotic theory of stakeholders, provides a way of dealing with representation. Combining aspects of ANT and Foucault's discourse theory allows us to include concepts as actors and promises a flexible and durable foundation for IS practice, but ANT itself indicates that the search for a purely theoretical foundation for IS is misguided.

## INTRODUCTION

An information system is a system – an interaction of people, devices and procedures; a technology – for providing information: news, facts which are useful and previously unknown, experiences that change our view of the world (Avison and Fitzgerald, 1995; Boulding, 1961). Such systems usually operate by acquiring information form other sources such as people, documents, sensing devices, traces of events, and representing it as data (possibly highly structured) and making these data available to potential users (interpreters) by either sending messages or storing the data for later retrieval. Many introductory texts (e.g. Oz, 2000) deal with the storage and transmission of data, the uses to which they might be put and sometimes the way in which data may be collected. Representation and interpretation are seldom treated comprehensively in information systems (IS) texts, except technically as coding and decoding. In this paper I will argue that representation is a key issue in IS design and use, and that semiotic theories provide an important foundation for the study of IS. I will further argue that actor-network theory (ANT) is a semiotic approach that is particularly suited to understanding information systems.

## REPRESENTATION IN INFORMATION SYSTEMS DEVELOPMENT

In traditional life-cycle approaches systems development takes place through a number of fairly clearly differentiated stages. Transitions between successive stages are marked by "sign-offs" where all stakeholders agree that the work so far has been completed satisfactorily. What is being agreed upon is some sort of document, usually a mixture of text and diagrams, which is supposed to represent in a certain level of detail (depending on the particular development phase) the system that will eventually be installed. From a semiotic point of view these text elements are signs that represent parts or features of the proposed system. There is considerable anecdotal evidence that different stakeholders have widely varying understandings of what they have agreed to in signing-off. We have also noticed, at least in student projects, that developers see the documentation as representing existing business rules and development work already completed, and that their idea, if any, of the final system, is imagined privately and separately from the agreed documents. Prototyping has come to be used as a supplement, or in some cases a replacement, for life-cycle approaches. With prototyping, representation of the future system is in the form of hardware and software artefacts as well as documents. Agreement on the design is negotiated continuously rather than at a number of discrete sign-off points. Concerns have been raised over the extent to which prototypes can represent issues of system management, maintenance and performance under normal operating conditions (Davis, 1984). From our point of view prototypes are also signs, and raise similar issues of understanding to documents.

Beyond prototyping there is evolutionary development. Here requirements construction, implementation and operation occur simultaneously. An embryonic system is built and put into use. With use some features are found to be helpful and are enhanced, others are ignored and gradually disappear from the system (although they often remain buried in unused code). As with prototyping, the speed of evolutionary change is facilitated by rapid software development environments, particularly where changes can be easily made by end users. In a sense, evolution of information systems is what always happens over time anyway. Adopting evolutionary development as a deliberate policy is a matter of realising that we will never be able to faithfully represent a future system, of removing controls that attempt to make the future system represent the original design, and allowing the evolutionary process to speed up.

We see design activities as social activities, exercises in communication. Where there is a designated "designer" their role involves at least two communication relationships - assuring the users that the system-to-be will make their lives better and guiding the system builders (often programmers) in their construction work. When prototyping is used extensively the designer might take a less dominant role, mainly facilitating communication between the users and the builders. Successful information systems design would seem to depend on establishing mutual understanding either between users and designers and designers and builders or between users and builders - and preferably among all three groups. An appreciation of how information systems are constructed through representation and shared understanding is critical for anyone attempting to manage the systems development process. This is so even with an evolutionary approach: design, building and use occur simultaneously, but even within one user there are conceptual shifts as they move among these three modes of system construction.

## Peirce's Semiotic Model

For these reasons, an important theoretical foundation for research and teaching in IS is semiotics (Nöth, 1990), the theory of signs and meaning. Fundamentally, semiotics is simply about representation, about how signs stand for things in the world of objects. One popular semiotic theory is Peirce's triadic model.

> A sign, or representamen, is something, which stands to somebody for something in some respect or capacity. It addresses somebody, that is, creates in the mind of that person an equivalent sign, or perhaps a more developed sign. That sign which it creates I call the interpretant of the first sign. The sign stands for something, its object. It stands for that object, not in all respects, but in reference to a sort of idea, which I have sometimes called the ground of representation. (Peirce Collected Papers of C.S. Peirce (2.228) quoted in Spinks, 1991)

In the case of IS development some of the things represented, systems-to-be, do not yet exist and we might expect to find that the same sign could have different meanings for different stakeholders. Hence IS development is about networks of meaning - the existing network of the organisation, the project development network where the information system becomes an object of shared understanding, and the changed organisation network which will include the completed information system. Documents, models and discussions are the links, which carry, or fail to carry, meaning in these networks.

Traditional descriptions of information systems rely not only on natural language but also on diagrams (structured or unstructured) and restricted or formal languages (such as pseudocode). Some authors attempt to use semiotics as a basis for yet another specialised IS specification language (Stamper, 1992), or use semiotic concepts to classify user communications in existing systems (Clarke, 1998). Our image of information systems development is strongly constructivist. This means that the world, in this case the supposed reality underlying the system design - the business rules and the technical possibilities, is created by the participation of the participants (users, designers, programmers, managers) (Watzlawick, 1984). Although some methodologies give the impression that system design follows logically from predetermined requirements, it is fairly obvious that information systems are in fact designed and built by people, almost always working with others, and sometimes with computers, in a social setting. But we also believe that the requirements themselves are socially constructed during the design process. In this situation aspiring IS developers do not need a semi-formal semiotics based methodology, nor a theoretical model of "best practice" organisational communication, but a knowledge, sensitivity and attitude which will keep them continuously aware of representation issues during the development process.

One representation issue, which (as we have mentioned) is of particular significance in design and development, is that of signs referring to future objects. Peirce provides some light on this issue when he distinguishes objects as immediate, dynamic and real. The immediate object is that which takes part directly in the sign occurrence, and is in a sense created by the act of interpretation (semiosis). The dynamic object has an existence independently of the particular sign occurrence and to some extent can be thought of as co-producing the sign. If there is anything real that “sufficiently corresponds” to the immediate object then this can be called the real object (Oehler, 1987, p10). The introduction of an immediate object avoids or at least defers a difficult semantic question: can a sign refer to an object that does not exist (such as a unicorn) or which is nonsensical (such as a square circle)? For Peirce anything referred to in a sign is meaningful, because by definition it means in some respect something to somebody (perhaps as an example of impossibility); the sign itself validates the immediate object. But can these unreal objects also be dynamic objects? We can regard them as such if the same object seems to take part in a variety of signs. A unicorn can represent good fortune, we can say “a unicorn is a mammal” and many children can draw one. It is reasonable to believe that these cases all refer to the same (dynamic) object.

The idea of a dynamic object is clarified when signs are used in a design process. Suppose we have a data-flow diagram of part of an existing payroll system. The data-flow diagram and the phrase "add new employee" are two sign vehicles referring to the same dynamic, and presumably real, object. In another context the data-flow diagram may signify a quite different object: if someone is photocopying the system documentation this diagram could indicate the middle page of the documentation. With an existing system we assume that the objects are to some extent already established; when designing a new system the objects are gradually brought into being. Suppose now that we begin our design by drawing boxes representing subsystems of the payroll system. We label one of these "add new employee". By this action we create an immediate object, which we imagine as a process, to which the sign refers. As we think about this object, mention it in reports, discuss it with others and perhaps ask someone else to draw a data-flow diagram for it, the immediate object gradually achieves the status of a dynamic object, existing independently of any particular sign occurrence. If we spend a large amount of money on hardware, software and training and persuade several people to change their payroll processing behaviour we might eventually feel confident in calling the object real.

The interpretant is what the sign means, or how the sign works. It is the “respect in which” the sign vehicle stands for the object. But what exactly is it? For Peirce the interpretant is that which the sign produces in the interpreter; it may be a feeling, an action or another sign. Since Peirce was an empiricist and pragmatist the ultimate meaning of the sign must be its observable effect, so the feeling must be eventually expressed by speech, action or symptom; in the end all interpretants are available as vehicles for further signs. As with objects, there are various types of interpretants. The immediate interpretant is the actual understanding of the sign (perhaps a process), the dynamic interpretant is the reaction provoked and the final interpretant is the “proper” reaction to which all interpreters converge (Oehler, 1987, p6). The combination of sign vehicle, final interpretant and dynamic object gives a general sign that is understood through objective consensus, not individual thought. This is important for Peirce because it removes signs from the unverifiable mental space of the interpreter. The disadvantage is that signs can only be understood in terms of other signs and in theory this leads to an infinite regress; but in practice we always manage to stop.

Until the object becomes real the assumed existence of a dynamic sign shared by a number of sign occurrences depends on the existence of a stable mental image. Agreement among sign users depends on two links: the object in question must invoke the same mental image in all users, and they must use the same sign to indicate that image. Because the common element, the mental image, is private and cannot be accessed except through other signs, it is impossible to tell whether agreement has been reached or how it has broken down. Management and workers may agree that workers should be "more productive" but management may have an image of more output per dollar while the workers imagine more output per person (and more pay). If, referring to the same events, managers say "we have down-sized by 10%" and workers say "500 people were sacked" management may be using a euphemism (different sign, same object and image) or may have a quite different mental image of the events.

In practice, although our mental images related to a sign differ and although any clarification must be based on equally ambiguous signs, we eventually build a network of connected signs that we are fairly confident others understand. That this confidence could be grossly misplaced is apparent when we discuss "red" with a colour-blind person, or are told by a teenager that certain behaviour is "sick". Nevertheless, such confidence is exactly what keeps communication going.

### Different Ways of Using Signs (Morris)

The pragmatic semiotics of Peirce and James, which believes "that, to develop a thought’s meaning, we need only determine what conduct it is fitted to produce: that conduct is for us its sole significance" (James, 1991, p23), is taken to its logical conclusion by Charles Morris. Morris has been credited with defining the terms syntax, semantics and pragmatics with reference to signs (Posner, 1987, p25). His own theory of signs is based on behavioural psychology. For Morris all behaviour is goal-seeking: the organism selects a desirable object (orientation), performs appropriate actions (manipulation) and fulfils its desire (consummation). Of course while this is going on many other organisms are doing the same thing so it is easy to become distracted or completely diverted; and organisms can have very obscure strategies where even they find it difficult to distinguish ends from means. The goal-seeking model can, however, still provide the basis for a definition of sign.

> Thus a sign is a preparatory stimulus that, in the absence of an impulse-satisfying object, causes in an organism a sequence of responses of the same type that would be caused by the object itself. (Posner, 1987, p28)

Here Morris was probably thinking of something like Pavlov’s bell and he would most likely expect his definition to expand to include more complex situations. If we imagine a specification as a sign for an information system then this definition could work in some cases. A customer might, under strict conditions, pay for the specification as if it were the system and a court of law might (again only in some circumstances) accept the specification as signifying the system. But an end user could not actually use the specification to do their work and a software engineer faced with the specification might be tempted to build the system, something they wouldn’t do if the system already existed. In the same place, however, Posner compares the sign to the behaviour sequence itself. The sign vehicle alerts the organism to the desired object, which will satisfy the impulse, while the interpretant is the impulse itself. In this case we can say that the meaning of the specification to the engineer (or the client) is the satisfaction they anticipate when using the completed system. Thus we might respond to an advertisement (the sign vehicle) for a new operating system by imagining that obtaining it will lead to an experience (the object) that will give us greater reliability or will make our friends think we are top computer scientists (the interpretant or imagined consummation).

Corresponding to the behavioural sequence, Morris defined three dimensions of signification. In the orientation phase it is important to assess the facts of the situation and designative signs are most helpful; in the manipulative phase we need rules or guidelines about what to do through prescriptive signs; and when considering consummation we need to evaluate the desired object using appraisive signs. We would normally, for instance, use a prescriptive sign if we wished someone else to do something for us ("please back up the disks regularly"); but in some circumstances it may be more effective to use designative ("did you hear how Fred lost all his data last week?") or appraisive ("our database is our most valuable resource") signs for manipulative purposes (Posner, 1987, p41).

## REPRESENTATION IN ACTOR-NETWORK THEORY

From the above we can see that meaningful signs, particularly for yet to be constructed objects, depend on the development of shared meaning, but that ambiguity of meaning can be used for a form of manipulation. Systems development methodologies often assume that all parties to the development share goals or, in the case of Soft Systems methodology, that a common direction is established at an early stage (Checkland, 1972). If various participants continue to have different interests then the likelihood of shared meaning developing is greatly reduced. Stakeholder theory is a widely accepted model for describing conflicting interests, but gives us little insight into the expression, communication and manipulation of these interests. Actor-Network Theory (ANT) provides a semiotic approach to stakeholder analysis.

### Actors, Networks and Meaning

Actor-network theory (ANT) was developed as a method in science and technology studies. It attempts to anchor meaning to the desires of people and the constraints of nature (Latour, 1993, p64) by granting status to hybrids or quasi-objects, which are heterogeneous networks of phenomena. The hole in the ozone layer, for instance, "links the most esoteric sciences and the most sordid politics, the most distant sky and some factory in the Lyon suburbs, dangers on a global scale and the impending local elections or the next board meeting" (Latour, 1993, p1). Similarly, a management information system is "an integrated user-machine system for providing information to support operations, management, and decision-making functions in an organization. The system utilises computer hardware and software; manual procedures; models for analysis, planning, control and decision-making; and a database" (Davis and Olsen, 1984, p6). By tracing the activity of these hybrid networks we can see how the world is interpreted, reinterpreted and changed. These networks of actors are the hybrids ("network", "hybrid" and "quasi-object" are interchangeable terms).

From the point of view of much traditional philosophy, confusing machines, people, ideas and organisations would be regarded as a category error. ANT makes category error its major virtue. Latour (1993, p11) explains this in the following diagram.

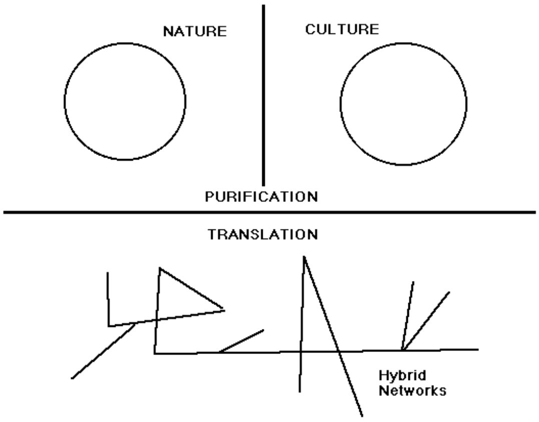

**Figure 1 - Purification and Translation**

He says that the project of modernism attempted to divide the world of thought into the natural and the social, a world of objects, forces and velocities, and a world of feelings, beauty and community. In one world statements may be true or false, in the other right or wrong. Before we can discuss or even think of some phenomenon we must decide on which side of the vertical divide it lies. This decision may vary over time, and may even divide different aspects of what appears to be one being - at a particular time in history, human anatomy may have been considered a natural phenomenon, while human thought may have been cultural. Purification ensures that we talk about the right things with the right word, that we avoid such "errors" as anthropomorphism.

But below is the hidden world of translation. Practical people are building factories, musical instruments, information systems and cloned sheep; artefacts in which the natural and the cultural are hopelessly mixed. The modern project requires that we don't think of the two levels, the worlds of purification and translation, at the same time. Otherwise the belief that thought can be clear, that ontology is possible, that our life can be controlled by the right ideas, would dissolve. And because we cannot use our much refined tools from the world of thought, the world of hybrids does run out of control and is filled with all sorts of monsters on which we depend but which we don't understand. How can we bring meaning to these networks of translation?

### Scripts and Translation

ANT is a type of stakeholder analysis where the stakeholders may be other than the usual people and organisations and the stakes may be reinterpreted as negotiations progress. Typical stakeholder phenomena such as the formation of alliances are dealt with in detail by Callon (1986) and will not be covered here. We will concentrate on the key concepts of ANT: actor, script and translation.

In IS development the obvious candidates for actors are the stakeholders such as owners, clients, operators and regulators; and the potential components of the future system are hardware, software, people and processes. As Callon comments, "... first ... the list of actors is not restricted to social entities; but second, and most important, their identities and their wishes are all constantly negotiated during the process of translation" (Callon, 1986, p227, p21). Scripts represent the interests of the actors. Notice that we have just uncovered another level of ambiguity in "representation". A script may be as simple as "close the door" or a million line computer program. The design of technical artefacts (quasi-objects) involves delegation of actions to various actors by inscribing. This process is complicated with a collection of script-based activities such as conscription, prescription, proscription and subscription, all of which may be ascribed to a particular situation by researchers (or project managers) (Akrich and Latour, 1992, p259).

Technical artefacts such as database servers or application programs are judged by their actions; they make performative statements. In systems development they will be conscripted to the purposes of other actors by being inscribed with scripts in a language, which they (pragmatically) understand. This conversion of scripts (perhaps into machine code) as they are inscribed from one actor to another is one of the several meanings of translation in ANT. Translation can model the operation of IS components as well as their design, for example when client requests are translated into keystrokes. It is widely accepted that failures in translation, both during design and operation, are a common source of system "malfunction". ANT allows us to model this mistranslation as possibly intentional (betrayal) and as an essential part of system design and operation, rather than as an exogenous disturbance, which can be removed. In the pure world of modern thought, where humanity and nature occupy separate domains, such a removal might be imaginable. In the messy world of modern practice, the world of technology in general and information systems in particular, such a removal would render translation impossible (Latour, 1993, p11).

### Representing Concepts in ANT

Latour gives us a vivid picture of the way scripts move and change among the network of actors, but does not address how the meaning of the scripts might change with the interests of various classes of actors. We have proposed (Underwood, 2001) that, using Foucault's (1972) concept of discursive formation, we can see how particular actors may translate scripts with reference to a "home discourse" such as accounting or computer science. This translation takes place in the "world of thought" and is facilitated by the use of learned concepts such as "efficiency" or "attribute". But in the "world of practice", where actors from different discursive backgrounds must engage in at least minimal collaboration if any progress is to be made at all, these concepts must be reinterpreted or betrayed to develop some level of shared meaning. As this process continues, some type of local language or "hybrid discourse" will develop and actors may gradually detach from their home discourse ("go native"). The ability to straddle these two worlds, shown in Figure 2, with relative comfort is a critical skill for any system developer.

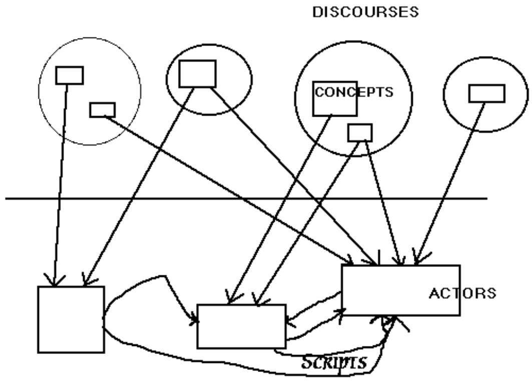

**Figure 2 - Concepts as Actors**

From a theoretical point of view, the fate of the concepts as they are betrayed and transformed can be followed in ANT by regarding the concepts themselves as actors (Underwood, 2001). This move was anticipated in stakeholder theory by Mitroff (1989). For practical purposes, however, we still need to quarantine the concepts from the rest of the network. This recognises that although concepts can inscribe other actors they are not, as long as they remain in their home discourse, themselves inscribable. In the long run the construction of multitudes of computer-based accounting information systems may change the discourses of both accounting and computing, but for any given development project these discourses have to be taken as given.

### Dealing with Key Concepts in IS Development

Over the years a number of concepts have been proposed as, or assumed to be, important to IS development. Most technically oriented approaches either explicitly or implicitly advocate rationality and objectivity. More socially oriented values are often relegated to treatment as "social impacts" or "professional responsibility" but a number of methodologies have attempted to build in social concepts from the beginning. Stafford Beer's (1985) Viable Systems Model focuses on control; Enid Mumford's (1979) ETHICS methodology emphasises freedom; Checkland's (1972) SSM is based on consensus; and current approaches to IS design emphasise knowledge. With ANT as a basis for our understanding of IS development we have neither to leave these desirable values for an "impact assessment" at the end nor to choose one and use it as the (somewhat disguised) basis of a complete methodology. These and other concepts can be seen as additional actors, represented in various guises during system design, development and operation, with a shifting band of allies and enemies. We can follow their fortunes and assess their influence on the "final" system, while all the time remembering that we are also actors in the network. If we can inspire this attitude in our students we may contribute to the growth of IS developers who are ethical, effective, and sometimes even both.

## CONCLUSION

Actor-network theory and Foucault's discourse theory have already been used by a number of authors (eg Bloomfield et al, 1997; Cooper and Bowers, 1995; Hanseth and Monteiro, 1997; Hocking, 1996; Poster, 1990; Remenyi et al, 1999; Woolgar, 1996) to better understand IS development and use. A combination of these theories, emphasising representation, translation and concept may provide a durable foundation for he study of information systems. But actor-network theory is not, according to Latour (1999), really a theory at all. It is either a framework, providing a series of tricks to aid our understanding, or a meta-theory, allowing us to compare other theories or even eras. Many have searched for a foundation theory for information systems. Davis (2000, p78) found that at least ten different major theoretical areas have been used recently as a basis for IS research and these are divided more or less equally between "hard" (if not natural) and social sciences. If Latour's analysis of modernism is correct, theories from these two "sides" will have been more or less deliberately constructed to be incompatible. To simplify, we may imagine "computer science" represents nature in Figure 1, and "sociology" represents Culture. Then the Hybrid networks will be represented by "information systems". The only way these networks can exist is by not referring back to "foundations"; otherwise they will be mired in contradiction and controversy. It follows that the only way the putative foundational disciplines can be represented is by some of their concepts taking part as actors, on an equal basis with all others (computers, bank tellers, financial regulations, etc). The practical and very useful role of ANT in any particular systems development or operation is to promote understanding, and to facilitate collaboration and betrayal. Its equally useful theoretical role in the “discipline” of information systems is to show us that there can never be any foundations, except in practice. Even though speaking of social research, the comment of Schratz and Walker (1995, p105) is apposite:

> Our aim is to resist generalising, to scrutinise every statement for values, to search for blind spots in our own perceptions in order to understand both the espoused theories and theories-in-use of all those involved and implicated in a particular setting, event or process. We see theory as implicit in every social action, not simply the concern of academic theorists, and rather than seeking to replace these everyday theories with those we consider better, we seek to explicate and understand them.

## REFERENCES

Akrich, M., Latour, B. (1992) "A Summary of a Convenient Vocabulary for the Semiotics of Human and Nonhuman Assemblies" in W. E. Bijker, J. Law (eds) **Shaping Technology / Building Society** MIT Press

Avison, D. E, Fitzgerald, G. (1995) **Information Systems Development**, McGraw-Hill

Beer, S. (1985) **Diagnosing the Syst**em, Wiley

Bloomfield, B. et al (eds). (1997) **Information Technology and Organizations**, Oxford University Press

Boulding, K. E. (1961) **The Image**, University of Michigan

Callon, M. (1986) "Some Elements of a Sociology of Translation: Domestication of the Scallops and the Fishermen of Saint Brieuc Bay" in J. Law (ed) **Power, Action and Belief: a new Sociology of Knowledge? Sociological Review Monograph**, Routledge and Kegan Paul

Checkland, P. B. (1972) "Towards a System-Based Methodology for Real-world Problem Solving" **Journal of Systems Engineering** 3(2)

Clarke, R. J. (1998) "Systemic Semiotic Development using SFX: Prototyping Genre-based Representations of IS Workpractices" **Proceedings of 1998 IEEE International Symposium on Intelligent Control (ISIC)** Sept 14th-19th 1998, pp846-851

Cooper, G. & Bowers, J. (1995) "Representing the User: notes on the disciplinary rhetoric of human-computer interaction" in P. J. Thomas. (ed) The Social and Interactional **Dimensions of Human-Computer Interfaces** Cambridge University Press

Davis, G. B. (1984) "Caution: User Developed Systems Can Be Dangerous to Your Organization" **Joint International Symposium on Information Systems ACS/IFIP**

Davis, G. B. & Olson, M. H. (1984) **Management Information Systems** McGraw-Hill

Davis, G.B. (2000) "Information Systems Conceptual Foundations" in R. Baskervilleet al (eds) **Organizational and Social Perspectives on Informational Technology** Kluwer, Boston

Foucault, M. (1972) The Archaeology of Knowledge Tavistock

Hanseth, Ø. & Monteiro, E. (1997) "Inscribing Behaviour in Information Infrastructure Standards**" Accounting, Management and Information Technology** 7(4), 183-211

Hocking, L. (1996) "Systems Analysis and Early Design as the Negotiation of Meaning and Interest" in C. D. Keen et al (eds) **Proceedings of 7th Australasian Conference on Information Systems ACIS’96**

James, W. (1991) **Pragmatism** Prometheus Books

Latour, B. (1993) **We Have Never Been Modern**, Harvester

Latour, B. (1999) "On recalling ANT" in J. Law, J. Hassard (eds) **Actor Network Theory and After**, Blackwell, Oxford

Mitroff, I. I. (1989) **Stakeholders of the Organizational Mind**, Jossey-Bass

Mumford, E. & Weir, M. (1979) **Computer Systems in Work Design**, Associated Business Press

Nöth, W. (1990) **Handbook of Semiotics**, Indiana University

Oehler, K. (1987) "An Outline of Peirce’s Semiotics" in M. Krampen et al (eds) **Classics of Semiotics**, Plenum Press

Oz, E. (2000) **Management Information Systems**, Thomson Learning

Posner, R. (1987) "Charles Morris and the Behavioral Foundations of Semiotics" in M. Krampen et al (eds) **Classics of Semiotics**, Plenum Press

Poster, M. (1990) **The Mode of Information**, Polity Press

Remenyi, D., White, Terry & Sherwood-Smith, M. (1999) "Language and a post-modern approach to information systems" **International Journal of Information Management** 19(1), February 1999, 17-32

Schratz, M. & Walker, R. (1995) **Research as Social Change** Routledge

Spinks, C.W. (1991) **Peirce and Triadomania**, Mouton de Gruyter

Stamper, R. (1992) "Signs, Organisations, Norms and Information Systems" in R. MacGregor (ed) **Proceedings of Third Australian Conference on Information Systems**, University of Wollongong

Underwood, J. (2001) **Meaning in IS Development**, unpublished PhD Thesis, University of Wollongong

Watzlawick, P. (ed) (1984) **The Invented Reality** W.W. Norton

Woolgar, S. (1996) "The User Talks Back: Directed Reading at the Front Line" **CRICT Discussion Paper No. 57**, Brunel University