Looking Back to Tomorrow: Research, Teaching and Practice in IS

Jim Underwood
School of Systems, Management and Leadership
University of Technology, Sydney
Sydney, Australia
Email: Jim.Underwood@uts.edu.au

Abstract

This position paper reflects on experiences in information systems practice, teaching and research, and considers the relevance of the recent trend to design thinking. It argues for the importance of maintaining an eclectic approach to the content, methods and hoped for outcomes of information systems education.

KEYWORDS

design thinking, eclecticism, information systems education

INTRODUCTION - LESSONS FROM AN ACADEMIC LIFE

Reflecting on 45 years of practicing, teaching and researching information systems development, I am left with some uncomfortable feelings. A feeling of failure: while each class has a few students who respond brilliantly, most 'just don't get it'; while theory and research support a number of simple and obvious guidelines, industry continues to ignore these. *Plus c'est la même chose*: despite massive changes in hardware, software and the social penetration of IT, the same issues remain unsolved. An overwhelming garbage can (March and Olsen 1976) of underutilsed theories: particular theories of information, programming, human behaviour, society are used as a basis for teaching, research and practice for a few years, then discarded for some other theory - and if you are old enough you see them recycled.

Why is this so? Perhaps the IS discipline is trying to solve the wrong problems; perhaps the problems of using IT are just too big, too wicked; perhaps we have just not yet found the right theory. There are also issues of impatience (theories take too long to show practical outcomes, too long to be accepted in the research community, too long for students to understand), organisation pressures (for quick profit, for prestigious publication, for good student surveys) and simply a desire to be innovative and fashionable.

When working as a programmer/analyst my search for suitable theory was prompted by immediate issues encountered in practice. Of particular interest in a time of limited (tape based) data storage was the question of what data to collect and keep. Other issues were the use of bad algorithms, understanding end-user needs and behaviour, the ambiguity of requirements specifications, and (this was more personal) the ultimate purpose of the systems we were building. While there was very little interest in theory as such (except perhaps artificial intelligence - *plus c'est la même*) some methods which our company carefully followed were based on implicit 'theories' such as 'design backwards from the required output' and 'modular systems are more robust and maintainable'. My search for suitable theory began with Ashby's (1964) Introduction to Cybernetics - I still have the copy which I failed to return to the ICL library in about 1968.

In the following section I will mention many of the theories that I have found useful in understanding information systems. Following that I will consider possible reference disciplines, in particular the recent trend to 'design thinking'. I will then briefly introduce the possibility considering service professions as reference disciplines, and conclude with a discussion of the implications for IS teaching, research and practice

A JOURNEY THROUGH POTENTIAL FOUNDATION STONES

The following outlines (roughly in time sequence) my discovery of theories and approaches that I considered (and still consider) helpful for understanding and practicing the design, development and use of information systems.

Cybernetics - this covers information, control and feedback, including the Law of Requisite Variety which explains, among other things, information overload; the application to management and information systems was extensively developed by Stafford Beer (1981). Beer helped design systems with minimal data flow, but they assumed defined objectives and hierarchies of control.

Systems Analysis - classic sequential analysis and design methodologies are based on Organisation and Methods (O&M) practices from the early 20th century; these were formalised by US defence researchers (Optner 1973).

Data Modelling - formal models are extensively covered in database courses, but do not help in determining which data to collect; theoretical problems with these models as representation are covered in Kent (1978).

Open systems - this is probably the most complex, most difficult and most useful framework for information systems design, and intersects with most of the theories following; our concern moves from goal seeking to goal setting and system viability; an excellent view of open systems theory is by Ackoff and Emery (1972) while Weinberg (1975) provides an easier entry point; Boulding (1961) provides a classification of different levels of systems complexity (useful for considering whether we are oversimplifying), and an excellent definition of information; prominent applications of open systems theory to IS design are soft systems methodology (Checkland and Scholes 1990) and socio-technical systems design (Emery 1978); Gall (1979) provides a fun but accurate guide for dealing with complex systems.

Failure - one way to understand the complexity of designed systems is to study system failures (Bignell et al. 1977); unfortunately, I suspect that unless they already have the habit of systems thinking, students will concentrate on the final link in the chain of failure, often blaming human error (Pidgeon and O'Leary 2000).

Management of Development - as well as methodology, system success is often attributed to the way development is managed; a most popular topic is project management, although teaching in this area usually ignores the people management aspect (Thomsett 1980); an early and exciting study of how programmers think is provided by Weinberg (1971) while negative effects of some programmer management techniques are explored in Greenbaum (1979); recent work on agile development (Beck 2001) brings together methodology and programmer behaviour, while the soft systems approach (Checkland and Scholes 1990) broadens development management to include more system stakeholders.

Political Concerns - the previous item raises the impact of development on developers and to some extent users; when we consider the wider purposes of an information system we are drawn to contemplate the 'social impact' of information technology; Jacques Ellul (1965) provides a pessimistic but compelling view of technology in general, Nora and Minc (1980) report on likely impacts of communications technology, while Mike Cooley (1982) and Kristen Nygaard (1996) consider union involvement in system design; a good deal of useful information systems research has been based on critical theory (Boland and Hirschheim 1987).

Organisation Theory - until recently most information systems were developed and used within organisations, so organisation theory is clearly relevant; it is also interesting in that it is also an eclectic discipline looking for foundations, making a useful comparison point for information systems; a general text which emphasises this eclectic nature is Bolman and Deal (2003); most useful for understanding information needs in different styles of organisations is Mintzberg (1983), while Thompson (1967) provides a good model of different types of technology in organisations; Ciborra (1996) combines Thompson's model with transaction cost analysis to suggest ways of designing information systems to support different organisation styles.

With the growth of the world-wide web and online discussion further theoretical issues, previously peripheral, have become important.

Ontology and Epistemology - what is, and how can we know it? this was always relevant to IS research, but is now also critical to IS practice - blogs, online news media, knowledge management, how can we reconcile such variety of information sources; the most useful basic theory is Foucault (1972) on discourse; for practical implications much research remains undone; one interesting contribution on cross-cultural knowledge is Ess and Torseth (2006).

Psychology of the Self - another popular question concerning online interaction is what does it do to self-image; this is another question we are only beginning to answer (e.g. Stefanone et al. 2011); Wiley (1994) provides useful theory for the post-modern construction of self, while Allucquère Rosanne Stone (1995) gives a radical view of self and technology.

DESIGN AS A REFERENCE DISCIPLINE

The previous section outlines some theories or concepts that might usefully contribute to IS teaching, research and practice. Another approach has been to attempt to model an information systems discipline on some other 'reference discipline'. Originally these models were implicitly based on the historical origins of information systems studies - in the USA from business schools (which, as we have seen, have their own identity issues), and in the UK from organisation and methods (through the influence of the National Computing Centre and the indigenous IT industry); in both these cases there was some influence of the then dominant 'human relations' approach, but in recent decades this has metamorphosed into 'human resources' which has a quite different ethic.

Engineering also presented an attractive model as a strongly organised discipline that built artifacts based on mathematical models, with highly visible project management practices; this model, however, appealed more to programmers than to information systems designers.

For a time I was attracted to information science (related to libraries). Early work on automated information retrieval (Wikipedia 2012) had some historical connection with early information technology and had useful measures of the value of information. Collaboration with information scientists has, however, proven to be difficult - possibly because their main interest is often archival, possibly as their research tended to move towards the more technical aspects of IT. Their recent work on information behaviour (Wilson 1999) has proven more promising.

A recent trend in academic IS has been to look to design as a reference discipline. This has been motivated by a need to remind ourselves that we are actually interested in the artifacts we are designing as well as our own processes, and is evidenced by a discussion of 'design science' (Gregor 2002), 'design thinking' (to a lesser extent) and perhaps the prevalence of a variety of types of 'architect' in the IT industry. I will concentrate here on the 'design thinking' movement.

Early education in computing and information systems was based, consciously or not, on the model developed for the Bauhaus in Weimar and Dessau (Snider 1996). This was based on three pillars - theory (in this case theory of art and colour), intimate understanding of materials (e.g. metals, paints) and practical creation; the main place of learning was the studio or workshop, and there was a strong emphasis on social interaction among students and faculty. A recent adaptation of this approach in IT education is described by Lynch (2001).

Several writers on design have been popular among IT designers over the last two decades. Christopher Alexander (1971) proposed that design could better fit context and human needs if based on patterns selected from those that evolved in societies over time. Anecdotally this is obvious - designers design by copying and adapting previously successful designs. (A decade ago the majority of my students' interface designs looked like Microsoft Windows, then like Google, and now like Facebook.) Alexander's approach has been adopted by some computer scientists as 'patterns', but has not been used as a basis of IS design methods, except by Ciborra (1996). The human-computer interaction community has drawn quite heavily on Norman's (1988) discussions of industrial design.

Perhaps my favourite writer on design is Victor Papanek (1972), who proposed six dimensions of good design: need, use (it works), method (done properly), association (feels right in the context), teleology (looking to the future), and beauty. Students are sometimes bemused when presented with this list, but I have found that when they are faced with a substantial design project they understand what these criteria mean in practice; 'beauty' needs to be explicitly included in the assessment criteria, but then their colleagues have little trouble in assessing it.

'Design thinking' has been popularised in business consulting circles by Martin (2009) and Brown (2008). Like most business school 'best practice' it is based on case studies from apparently successful companies, and was soon followed by claims that it had been superseded (Nussbaum 2011). The characteristics of design thinking are said to be empathy, integrative thinking, optimism, experimentalism and collaboration (Brown 2008); there is also talk of broadening context, taking a systems view, and rapid prototyping. The failure comes when organisations that do not have a suitable culture attempt to introduce design thinking as a set of policies and procedures, a methodology. None of this seems particularly novel for IS design.

While the discovery of 'design thinking' by business consultants has not been particularly helpful in itself, it has sharpened focus on the behaviour of traditional designers, such as architects, industrial designers and graphic designers (Stewart 2011). Some notable studies have been done previously on how designers work. Lawson (2006) has over many years conducted studies of the way designers work, including comparisons of design students and science students, and observations of the protocols of expert designers. He found that (beyond their first year) science students tend to focus on the problem while design students focus on the solution, that different teams of design students working on the same problem differ widely in their 'prototype' first conceptual models, and that (similarly) all experienced designers have well practiced design strategies, but that these strategies vary greatly from one designer to another. Eventually design teams have to engage with the same issues and constraints, but they differ in their starting points and the sequence of their design processes.

> "I trust the gut feeling, the intuitive hand, the intuitive feel about the project ... you can technically solve accommodation problems, you can solve problems of view and so on but which problem to solve first is a gut feeling ... you can't explain it but you feel that's right and nine times out of ten you are right." - Malaysian architect Ken Yeang, quoted in Lawson (2006, p.203).

Dorst (2011) characterises design as relying heavily on abductive reasoning - that is, we know (more or less) the values we are trying to satisfy, but often know initially neither the situation (needs, constraints) nor the means by

which we can achieve these values. Dorst suggests designers address this conundrum by viewing the situation through a frame, where a frame is a combination of method and intended value; baroque, gothic and modernist designers will work with quite different frames. In information systems design a structured analysis and design approach values efficient programming, a soft systems approach values stakeholder acceptance. Since design situations are complex and wicked (and different stakeholders will embrace different values) good designers might need to revise their frame as their project proceeds.

The ability of the designer to use flexible framing depends on how their role is perceived by their client. Paton and Dorst (2011) interviewed expert designers and found four levels of engagement, which they named 'technician', 'facilitator', 'expert/artist' and 'collaborator'. Designers naturally preferred a deeper level of engagement, with early involvement with the client and projects that were seen as innovative rather than typical. To achieve full collaboration with the client, allowing for possible frame changes, they worked to establish a common 'local language'.

A non-designer's view of design, based on the work of Peter Sloterdijk (2005), is given by Latour (2008). The features that Latour finds attractive in design practice are modesty, attention to detail, the importance of meaning, never beginning from scratch, and having ethical or aesthetic values. From Sloterdijk he derives (as he would) the idea is that design is an 'explication' of the situation, a situation always comprising nature, humans and designed objects in their indistinguishability. Design is about managing concerns rather than producing new objects.

If we attempt to take traditional design as a model for information systems design we are faced with a few problems. Clients, on the whole, are reluctant to see IS designers as collaborators. This is partly because information systems (whatever they are?) have come to be regarded as objects of common knowledge, requiring only skilled technicians to implement them, and partly because, until the advent of aggressive outsourcing, IS designers did not have an image as independent professionals, as do architects or engineers. Additionally, while IS students and practitioners see themselves as rational and clever problem solvers, most do not see themselves as creative in the artistic sense. For example, computer oriented designers often see a user interface as an afterthought, an attractive marketing tool.

Some similarities are not particularly helpful either. Both traditional and information systems designers can tend to individualism, being reluctant to engage in serious teamwork, and both can be disdainful of users or clients (Pressman 1995).

Nevertheless, Stolterman (1992) found that experienced information systems designers are similar to traditional designers in that they value both visionary thinking and abstraction, they use methodologies selectively and adaptively, and they tend to come up with their initial vision of a solution almost immediately, prior to detailed analysis. One question that remains unanswered is exactly how designers envision information systems.

In a recently published collection of articles Fred Brooks (2010) argues for alternatives to the rational design model for information systems.

INFORMATION SYSTEMS AS A SERVICE

A more radical option is to view information systems as a service, using medicine, teaching, librarianship or social work as reference models. These professions cover a range of individual work and work within large organisations, but all are supposed to focus on client needs. A comparison between IT consulting and marriage counselling is tempting (this comparison has already been made by Larry Constantine - pers. comm.).

In one sense this model is realistic; from private contractors to big four consultants and major outsourcers most IS professionals have ongoing relationships with their clients - designing, developing, maintaining, expanding, redesigning. And the information systems themselves usually provide a continuing service to their users, a service that has to be nurtured.

On the other hand this model will not be easily accepted. Both designers and clients desire bounded projects, with manageable budgets and defined end products, which give developers a sense of achievement and clients a sense of security. Some years ago a type of service model was tried with 'partnerships' between major IT firms and major clients, but these arrangements have not continued. Single outsourcing arrangements are being replaced by multi-sourcing, and all outsourcing is plagued by the intricacies of service level agreements (Karten 2001).

Still, cultures can change, and the service model seems in some ways to better reflect the realities of information systems work. This is an area for future investigation.

IMPLICATIONS FOR TEACHING

Despite its limitations the Bauhaus method seems to be the best model for educating information systems designers. Most Australian IT courses are to some extent based on this method, although studio style learning is under threat because of the expense of mentoring and supervision.

To properly implement the model a number of difficulties need to be faced. Project based learning is essential. Ideally this can be achieved through co-operative education, where students spend extended periods working in industry, absorbing the industry culture (Goldberg 2012). Alternatively students can undertake large team projects, preferably extending over several semesters and with real clients. We also need to encourage students who have assumed they are not meant to be creative to come up with visions of their solution. (I usually emphasised the need for a simple theme - as with a root definition - but I shouldn't have kept telling them to do the analysis before coming up with a solution.)

Importantly, students need to be active learners, engaged in what is variously called problem-based learning or discovery learning (Bruner 1966). Unfortunately, this tends to be resisted by students, faculty and university management, because there are no clear content style learning objectives. We don't know beforehand what students will discover, what mental models they will use to understand the problem situation and develop solutions; and we must not assess them on whether we agree with these discoveries or models. The student stress is partly alleviated through supportive feedback, group responsibility and peer assessment; faculty worries are less for those who have a relativistic or constructivist view of knowledge, but this is rare, especially amongst science trained staff; I usually manage to minimise management stress by writing 'working in groups' and 'dealing with uncertainty' into my subject objectives wherever possible.

The Bauhaus model also included study of theory. In a previous section I have suggested a multitude of theories that could provide foundations for understanding information systems. It is obviously not possible to 'cover' all these topics. In a sense it doesn't matter where we start, as long as staff are comfortable with the theories used and students are able to pursue alternatives, deciding which theories suit their needs. Different students will learn different things, but this can be handled through tutorial discussion and flexible assessment. In some subjects I have deliberately avoided introducing methodologies before students commenced their projects, allowing them to learn what they needed as they progressed. Interestingly, most of them learned, but didn't articulate, a version of soft systems methodology.

The third pillar of Bauhaus is understanding materials. From the technical aspect that might include (minimal) practice with programming, web site building, network setup or product evaluation. But the materials of information systems also include information and people. Group projects are a rich (if not always pleasant) field for understanding people, while it is possible to set reflection style exercises on the real world use of information.

If we take seriously the idea of information systems as a service, ethics is an important area for education. Our students are from a wide range of cultural, religious and political backgrounds, so they are unlikely to pay more than lip service to any set of rules we propose. Again, we need to admit a relativist view, allowing students to develop their own ethics as an IS professional. Amongst other things this will involve the ability to question the methods and end products of their work from a broad ethical stance.

To develop their own knowledge and to deal with all the complexity described above, students must be strongly encouraged and assisted to regularly reflect on their experiences, to abstract principles and develop their own narratives (Schön 1987). At UTS almost all subjects include a requirement to 'reflect' as part of the assessment. Students are rather sick of this and the reflection is often rather mechanical. I assume that this is partly due to a feeling that 'right' reflections will be rewarded. Just as we have with group work and ethics, we need to provide theory, assistance and opportunity to support true reflective learning.

IMPLICATIONS FOR RESEARCH AND PRACTICE

Implications for research and practice are less clear. If we see people as critical for information systems (whether as stakeholders or clients) then continuing HCI research, particularly that based on theories of situated action (Suchman 2007), is essential. More generally, there needs to be more and more detailed research on user behaviour and the place of IS developers in organisations and society, especially with reference to social networking systems. This also relates to the need for research into the psychology, culture and politics of systems design, systems development and project management; (Latour 1996) provides the paradigm for this type of research.

These research suggestions fit well with what is needed in IS practice to improve the process of developing information systems. More difficult questions are what type of systems should we design (or promote) and what

values should inform these designs. If we see information systems as a service, then we need systems that support rather than attempt their clients. But who are the clients? The earlier part of my career was spent in a time when concepts such as industrial democracy and participation of the widest range of stakeholders were well accepted. This implies that support should be provided at all levels within organisations: managers and line workers; shareholders, regulators, suppliers and customers (and even competitors?). This requires truly flexible systems, systems which acknowledge a variety of logics - technologies of foolishness (March and Olsen 1976) - rather than attempting to enforce bureaucratic rationalism (Bjørn-Andersen and Eason 1980); systems that promote innovation, creativity and flexible responses in their users (Hedberg and Jönsson 1982). Unfortunately, in these times of managerialism, of excessive emphasis on measurement and governance, the acceptance of this type of system seems very unlikely in formal organisations, although this seems to be exactly what social media systems can do. In some ways the technology is ahead of the culture.

CONCLUSION

Developing, teaching and researching information systems are all complex and difficult activities, requiring knowledge of theories and the materials we work with, constant practice and reflection, and a willingness to change our abductive frames to suit the situation. The observant reader will notice that most of my references are at least two decades old (and I could probably have gone back two millennia if necessary). This reflects the fact that our concerns have been recognised since the beginning of the discipline and, although we have many excellent tools, we have still not been able to use them effectively. Sometimes this leaves me with a sense of hopelessness, but if it had been possible it would not have been worth doing. I conclude my career with the assurance that information systems will remain gloriously undisciplined for a long time to come.

REFERENCES

Ackoff, R.L., and Emery, F. 1972. *On Purposeful Systems*. Chicago: Aldine.

Alexander, C. 1971. *Notes on the Synthesis of Form*. Cambridge MA: Harvard UP.

Ashby, W.R. 1964. *An Introduction to Cybernetics*. London: Methuen.

Beck, K. 2001. *Planning Extreme Programming*. Boston: Addison-Wesley.

Beer, S. 1981. *Brain of the Firm*. New York: Wiley.

Bignell, V., Peters, G., and Pym, C. 1977. *Catastrophic Failures*. Milton Keynes: Open University.

Bjørn-Andersen, N., and Eason, K.D. 1980. "Myths and Realities of Information Systems Contributing to Organisational Reality," in: *Human Choice and Computers, 2,* A. Mowshowitz (ed.). Amsterdam: North-Holland.

Boland, R.J., and Hirschheim, R. (eds.). 1987. *Critical Issues in Information Systems Research*. Chichester: Wiley.

Bolman, L.G., and Deal, T.E. 2003. *Reframing Organizations*, (3 ed.). San Francisco: Jossey-Bass.

Boulding, K.E. 1961. *The Image*. Ann Arbor: University of Michigan.

Brooks, F.P. 2010. *The Design of Design*. Upper Saddle River NJ: Addison-Wesley.

Brown, T. 2008. "Design Thinking," *Harvard Business Review*), June.

Bruner, J. 1966. *Toward a Theory of Instruction*. Cambridge MA: Harvard UP

Checkland, P.B., and Scholes, J. 1990. *Soft Systems Methodology in Action*. Chichester: Wiley.

Ciborra, C.U. 1996. *Teams, Markets, and Systems*. Cambridge: Cambridge UP.

Cooley, M. 1982. *Architect or Bee?* Boston: South End Press.

Dorst, K. 2011. "The Core of 'Design Thinking' and Its Application," *Design Studies* (32:6), November, pp 521-532.

Ellul, J. 1965. *The Technological Society*. London: Jonathan Cape.

Emery, F. 1978. *The Emergence of a New Paradigm of Work*. Canberra: Australian National University.

Ess, C., and Torseth, M. 2006. "Neither Relativism nor Imperialism: Theories and Practices for Global Information Ethics," *Ethics and Information Technology* (8:3), pp 91-95.

Gall, J. 1979. *Systemantics*. London: Fontana.

Goldberg, J. 2012. "Active Learning in Capstone Design Courses," *IEEE Pulse* (3:3), May, pp 54-57.

Greenbaum, J.M. 1979. *In the Name of Efficiency*. Philadelphia: Temple University.

Gregor, S. 2002. "Design Theory in Information Systems," *Australasian Journal of Information Systems* (special issue), December 2002, pp 14-22.

Hedberg, B., and Jönsson, S. 1982. "Designing Semi-Confusing Information Systems for Organizations in Changing Environments," *ACM SIGMIS Database* (13:2-3), Winter-Spring 1982, pp 12-25.

Karten, N. 2001. "How to Establish Service Level Agreements." Retrieved 25 July, 2012, from http://gatsby.tafeitltn.com.au/~rileyc/certIV/support/class/sla.pdf

Kent, W. 1978. *Data and Reality*. Amsterdam: North-Holland.

Latour, B. 1996. *Aramis or the Love of Technology*. Cambridge MA: Harvard UP.

Latour, B. 2008. "A Cautious Prometheus? A Few Steps toward a Philosophy of Design," *Networks of Design - Design History Society meeting*, Falmouth.

Lawson, B. 2006. *How Designers Think*, (4 ed.). Amsterdam: Elsevier.

Lynch, K. 2001. "It Infrastructure for an Undergraduate Studio-Based It Degree," *12th Australasian Conference on Information Systems*, Coffs Harbour NSW.

March, J.G., and Olsen, J. 1976. *Ambiguity and Choice in Organizations*. Bergen: Universitetvorlaget.

Martin, R.L. 2009. *The Design of Business*. Boston: Harvard Business School.

Mintzberg, H. 1983. *Structure in Fives*. Englewood Cliffs NJ: Prentice-Hall.

Nora, S., and Minc, A. 1980. *The Computerization of Society*. Cambridge MA: MIT Press.

Norman, D.A. 1988. *The Psychology of Everyday Things*. New York: Basic Books.

Nussbaum, B. 2011. "Design Thinking Is a Failed Experiment. What Next?" Retrieved 25 July, 2012, from http://www.fastcodesign.com/1663558/design-thinking-is-a-failed-experiment-so-whats-next

Nygaard, K. 1996. "Those Were the Days? Or Heroic Times Are Here Again?," *Scandinavian Journal of Information Systems* (8:2), pp 91-108.

Optner, S.L. (ed.) 1973. *Systems Analysis*. Harmondsworth: Penguin.

Papanek, V. 1972. *Design for the Real World*. London: Thames and Hudson.

Paton, B., and Dorst, K. 2011. "Briefing and Reframing: A Situated Practice," *Design Studies* (32:6), November, pp 573-587.

Pidgeon, N., and O'Leary, M. 2000. "Man-Made Disasters: Why Technology and Organizations (Sometimes) Fail," *Safety Science* (34:1-3), February 2000, pp 15-30.

Pressman, A. 1995. *The Fountainhead-Ache*. New York: Wiley.

Schön, D.A. 1987. *Educating the Reflective Practitioner*. San francisco: Jossey-Bass.

Sloterdijk, P. 2005. "Forward to the Theory of Spheres," in: *Cosmograms*, M. Ohanian and J.C. Royoux (eds.). New York: Lukas and Sternberg, pp. 223-241.

Snider, C. 1996. "Bauhaus Philosophy." Retrieved 25 July, 2012, from http://academic.chrissnider.com/bauhaus/pages/philosophy.html

Stefanone, M.A., Lackaff, D., and Rosen, D. 2011. "Contingencies of Self-Worth on Social-Networking Behavior," *Cyberpsychology, Behavior and Social Networking* (14:1-2), pp 41-49.

Stewart, S.C. 2011. "Interpreting Design Thinking," *Design Studies* (32:6), November, pp 515-520.

Stolterman, E. 1992. "How System Designers Think About Design and Methods," *Scandinavian Journal of Information Systems* (4), pp 137-150.

Stone, A.R. 1995. *Desire and Technology*. Cambridge MA: MIT Press.

Suchman, L. 2007. *Human-Machine Reconfigurations : Plans and Situated Actions*, (2 ed.). Cambridge: Cambridge UP.

Thompson, J.D. 1967. *Organizations in Action*. New York: McGraw-Hill.

Thomsett, R. 1980. *People and Project Management*. New York: Yourdon Press.

Weinberg, G.M. 1971. *The Psychology of Computer Programming*. New York: van Nostrand Reinhold.

Weinberg, G.M. 1975. *An Introduction to General Systems Thinking*. New York: Wiley.

Wikipedia. 2012. "Information Retrieval." Retrieved 25 July, 2012, from http://en.wikipedia.org/wiki/Information_retrieval

Wiley, N. 1994. *The Semiotic Self*. Cambridge: Polity Press.

Wilson, T. 1999. "Models in Information Behaviour Research," *Journal of Documentation* (55:3), pp 249-270.

COPYRIGHT