Special Seminar: Assessing Emergent Business IT Using the Web of System Performance

Brian Whitworth[1], *Cheikna* Sylla[2], *Elizabeth Whitworth*[3]

[1]Institute of Information and Mathematical Sciences, Massey University (Albany), New Zealand

[2]School of Management, New Jersey Institute of Technology, USA

[3]Psychology Department, Carleton University, Canada

Abstract: Businesses must often decide whether to purchase emergent technology in various states of maturity. Purchasing immature technology can have serious consequences for a business, but equally not purchasing new technology can invoke intangible opportunity losses that are equally costly in the long term. Businesses that don't upgrade their IT can go out of business, but upgrading every time can be equally disastrous. How does one find the "sweet spot" with respect to new technology? Traditional evaluation models, like the technology acceptance model (TAM), predict poorly, yet were designed over 15 years ago for a different type of software. Complex modern software is not just technical but socio-technical, i.e. it has a social dimension. The Web of System Performance (WOSP) model combines traditional and modern criteria. It proposes eight intrinsic system factors affect performance, which when combined with external factors like marketing and distribution, give a general framework for assessing emergent IT. This tutorial covers definitions and examples of the eight dimensions, their relation to other theories and practices, cross-cutting and integration issues, how technology can "bite back", and environment factors outside the model. Recent research will also be reported.

1. INTRODUCTION

The use of information technology (IT) has become a primary survival factor for business organizations in a global competitive environment. However just as IT can make money for business, it can also lose money, as IT has become a major corporate expenditure. Some estimate that in the last two decades, about 50% of all new capital investments in organizations have been in information technology (Westland & Clark, 2000), and the total worldwide expenditure on IT exceeded one trillion US dollars per annum in 2001, with a 10% annual compounded growth rate (Seddon, Graeser, & Willcocks, 2002). Even so, IS organizations today have less money available for information technology than before (Rivard, Poirier, Raymond, & Bergeron, 1997), so the pressure on the IT dollar is increasing. In general terms this means that businesses must "buy smarter" to justify huge IT investments (Hitt & Brynjolfsson, 1996). This paper reports a tutorial on assessing emergent business information technology. A brief introduction discusses *why assess technology*, then considers eight important socio-technical dimensions of information technology assessment in today's complex global world. It explains how a web of system performance can be constructed from these eight dimensions that can be used to assess human-machine systems in general, and business information systems in particular.

2. WHY A NEW THEORY OFINFORMATION SYSTEMPERFORMANCE?

In the infancy of software development, designers held functionality (what the system does to the world) as the primary goal of software development. This is because at that time, software was just a tool, as say a hammer is a tool. As information systems developed however, they not only became more complex, but also less passive and more active systems in their own right. IS today works with the user not just *for* the user, and now enables a virtual online society that could span the globe. Hence functionality has become an insufficient indicator of information system performance. The main battle against functionality as the prime directive of system designers was carried out by the proponents of usability, human-factors and human-computer interaction, supported by theoretical frameworks such as the Technology Acceptance Model (TAM). These views presented ease of use as equal to usefulness in determining user acceptance of a system (Davis, 1989), e.g. if a web site performs well functionally, but users don't like it and click on to other sites, then it is a failure. Functional failure and usability failure it was noted have the same effect – the system does not run!

The Web of System Performance (WOSP) model goes a step further. It proposes not just two system performance criteria, but eight distinct dimensions, including usability and functionality. It is a multi-attribute model of system performance that links two different literatures:

1. System design requirements (e.g. security and reliability)
2. System evaluation and acceptance (e.g. usability and social requirements).

3. WHY DOES A THEORY OF PERFORMANCE MATTER?

One could say that a major goal of progress is to increase performance overall. However, many predictions of computer technology performance increases have not happened as predicted. For example, the paperless office (Toffler, 1980) was predicted but today we use more paper than ever before. If you had invested in a paper company when Toffler made his prediction, you would have made money. Likewise James Martin predicted the demise of programmers, but since then the demand for programmers has steadily increased. Programming is alive and well. Perhaps the oldest prediction of all was the "leisure society", where we would all be working a three-day week now, as computers and technology took over people's work. This did not happen as expected, because modern workers are busier and more stressed than ever before (Schor, 1991).

The WOSP model suggests why the directions of IS progress have often confounded expectations: namely that these, and other similar predictions, failed due to a too narrow view of system performance. It can also explain the success of advances such as the World Wide Web, HTML, cell phones, and chat, advances that slipped in under the theoretical radar, and can offer guidance in the development of successful future systems.

4. WHAT IS A SYSTEMS APPROACH?

Nearly forty years ago Bertalanffy noted that certain mathematical formulas repeated across many disciplines like chemistry, physics and biology (Bertalanffy, 1968), which used the same formulae to describe completely different things. Hence was borne the idea of studying a "system" without referencing what type of system it was.

Computer systems seem systems in a general sense (Churchman, 1979), so a hardware computer system of chips and circuits is also a software system of information exchanges, and today also the human-computer combination (Alter, 1999), e.g. a plane is mechanical, its computer controls are informational, but the plane plus pilot is also a system – a human-computer system. Human-computer interaction (HCI) sees computers as more than just technology (hardware and software). Computing began as hardware in the 1950s and 1960's, progressed to commercial information processors in the 1970's, to personal computers in the 1980's, to computers as social communication tools in the 1990's. This decade seems to be that of social computing, where software serves not just people but social groups, with systems like email, chat rooms and bulletin boards. Table 1 summarizes the four computer system levels, matching the idea of an information system (IS) as hardware, software, people, and business processes (Alter, 1999). The levels are different views of the same system not different systems, and match disciplines of Engineering, Computing, Psychology and Sociology, respectively.

Table 1 Information system levels

Level	Examples	Discipline
Social	Norms, culture, laws, zeitgeist, sanctions, roles	Sociology
Cognitive	Semantics, attitudes, beliefs, opinions, ideas, morals	Psychology
Information	Software programs, data, bandwidth, memory, processing	Computing
Mechanical	Hardware, computer, telephone, FAX, physical space	Engineering

5. WHAT IS THE WEB OF SYSTEM PERFORMANCE?

The Web of System Performance model decomposes system performance into many goals as suggested by Chung (Chung, Nixon, Yu, & Mylopoulos, 1999). A full derivation is given elsewhere (Whitworth, Fjermestad, & Mahinda, 2005; Whitworth & Zaic, 2003). This analysis suggests any advanced system has four elements: a boundary, a supporting internal structure, output effectors, and input receptors (Whitworth & Zaic, 2003). For example: cells have a membrane boundary, internal support (nucleus), flagella to move (effectors), and photo-receptors; people have a skin boundary, internal brain and organs, acting muscles, and sensory input; computers have a physical case, motherboard architecture, printer/screen effectors, and keyboard/mouse "receptors"; and finally software has a memory boundary, an internal program structure and specialized input/output modules. If performance is defined as how well a system interacts with its environment to gain value and avoid loss, each element can be designed to maximize opportunity or minimize risk. Hence one can derive eight performance goals:

1) *Boundary: (defines system entry)*
a) To enable useful entry (*extendibility*).
b) To deny harmful entry (*security)*.
2) *Internal structure: (controls and sustains)*
a) To accommodate external change (*flexibility*).
b) To accommodate internal change (*reliability*).
3) *Effector: (changes the environment)*
a) To maximize external effects (*functionality*).
b) To minimize internal effort (*usability*).
4) *Receptor: (senses the environment)*
a) To enable meaning exchange (*connectivity*).
b) To limit meaning exchange (*privacy*).

6. WHAT DOES THE FIGURE REPRESENT?

In the web of system performance (Figure1) the:

- Web area represents system performance in general, so a bigger the area means a greater system performance potential.
- Web shape represents the goal criterion weights, which vary with the environment, e.g. a threat environment may mean security has more weight.
- Web lines represent goal tensions, imagined as connecting rubber bands that can pull back one performance dimension as another increases.

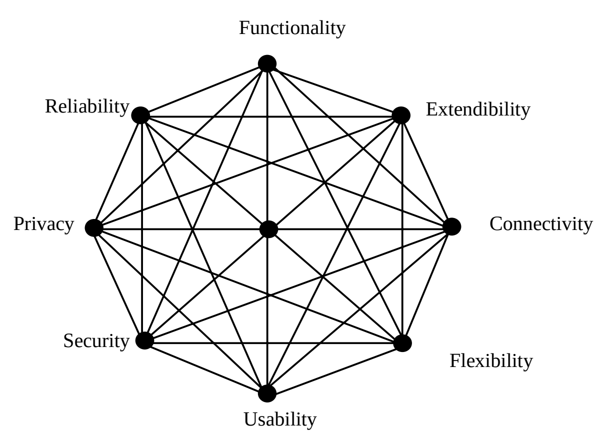


Figure1 The Web of System Performance.

7. MANY OF THE DIMENSIONS SEEM FAMILIAR, WHAT IS NEW ABOUT THE WOSP MODEL?

None of the WOSP dimensions are new. System goals like privacy and security have been quoted in the literature for a long time. While the WOSP dimensions are not new (see Table 2 for similar terms), their integration into a common framework under a general performance concept is new. Also new is that it derives its concepts from the general nature of systems. Some see performance as just functionality, but if your car has poor locks and is stolen, how well does it perform? The WOSP model says it does not perform very well. If a "high performance" hotrod won't start, does it perform well? Again the WOSP model says no. Note that most modern systems devote more lines of code to interface and error routines than to the functional mainline. There are four opportunity increasing dimensions (functionality, flexibility, extendibility, connectivity), and four failure avoiding dimensions (security, reliability, privacy, usability). The risk terms can be linked to the European General Security model, which calls all four terms together "security," so privacy in this view is a part of security (as is reliability).

Table 2 WOSP Dimensions

Dimension	Similar Terms
Extendibility	Openness, interoperability, permeability, compatibility, scalability.
Security	Defense, protection, safety, threat resistance.
Flexibility	Adaptability, portability, customizability, plasticity, agility, modifiability.
Reliability	Stability, dependability, robustness, ruggedness, durability, availability, maintainability.
Functionality	Capability, effectualness, usefulness, effectiveness, power, utility.
Usability	Ease of use, simplicity, user friendliness, efficiency, accessibility.
Connectivity	Networkability, communicativeness, interactivity, sociability.
Privacy	Confidentiality, secrecy, camouflage, stealth, social rights, ownership.

8. HOW ARE THE DIMENSIONS EDFINED?

Table 3 defines the performance goals in systems terms, but any analysis must also specify the system level as each WOSP goal is different at each level, e.g., while usability usually refers to less personal cognitive "effort," on a software level it means less memory/processing (e.g., "light" background utilities), and on a hardware level, less power usage (e.g., mobile phones that last longer). Similarly, engineers talk about system "confidentiality" while sociologists talk about "privacy", but in general system terms, it is the same concept.

Table 3 System Performance Goals

System Element	Dimension	Definition
Boundary	Security	To protect against unauthorized entry, misuse or takeover.
	Extendibility	To make use of outside elements
Internal structure	Flexibility	To still work in new environments
	Reliability	To continue operating despite internal changes like part failure
Effector	Functionality	To act directly on the environment to produce a desired change
	Usability	To minimize the relative resource costs of action
Receptor	Connectivity	To communicate with similar systems
	Privacy	To control the release of information about itself

8.1 Ideas seem similar to Alexander's synthesis of form, which forms the basis of pattern theory today.

Yes, this model merely applies Alexander's theory to IS. Over forty years ago Alexander noted the "tension" problems of physical world system design (Alexander, 1964). Since then, his architectural pattern theory has been applied to information systems (IS) and object orientated (OO) design. Design tensions arise when physical systems composed of parts have multiple contextual demands. For example, in a simple machine such as a vacuum cleaner, each part, like the engine, can be designed for its specific function by using the best materials. Specialized materials allow a powerful engine, with more suction, but this may also create more noise, heat and weight, making the vacuum harder to use. Part specialization may also mean more complex joints that fail easier, reducing reliability. Finally, customizing parts can increase manufacturing material diversity, raising costs. Figure 2 shows the system design problem space to create a vacuum with powerful suction that is also cheap, lightweight and reliable.

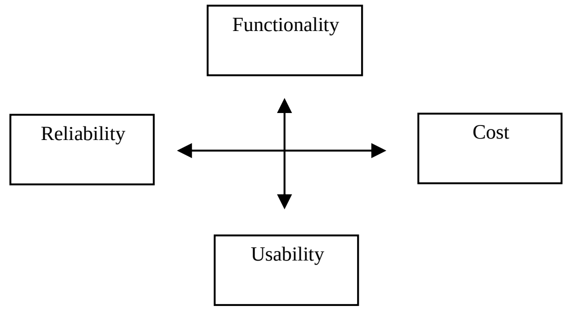


Figure 2 Design Tensions.

Design then, is the art of synthesizing "forms" to reconcile contradictory contextual demands, e.g. vacuums that are both lightweight and powerful. "Patterns" are generic solutions to design conflicts that repeat: "*Each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem.*" (Alexander, 2005). If problems repeat, it makes sense to re-use successful solutions. The logic applies as well to IS design as it does to physical design.

8.2 Is WOSP useful for system evaluation as well as system design?

Yes, it can be used as a process-oriented design framework for system developers, or a product-oriented evaluation framework for system users/buyers. The common concept of system performance connects the two fields: generally designers want to produce high performance systems, and likewise users want to buy them.

8.3 How is WOSP useful for system evaluation?

Unlike theories that focus on a few criteria, like security or usability, the WOSP model combines *many* IS criteria under an overall goal of system performance, i.e. successful environment interaction. This gives a broader view. For example, Microsoft Windows superseded DOS by adding usability, flexibility, connectivity and extendibility, but in making these gains lost reliability. One of the main goals of Windows XP was to improve Windows reliability by improving the kernel. This approach raises an issue designers can easily forget: *performance integration,* combining what are called "cross-cutting requirements". For example, if one buys a system with excellent security but poor usability, staff by-pass it. It is this integration issue that makes system design as much an art as a science, and why the assessment of software is more complex than many suppose.

8.4 Can you give an example of how new products need integration?

The WOSP measure of performance is the total *area* of the web. Early developments may be easy, as any improvement in a dimension simply takes up "slack". As systems evolve however, "tensions" occur, and then focusing on specialties may distort the web without increasing the overall area of system performance. If increasing one dimension of performance makes others decrease, the result may even be an overall decrease in performance. It is only through an area increase that real system development occurs.

This may require designers to develop 'combination breakthroughs', which genuinely expand the web by "pulling" on two or more sides at once. For example, in 1992, Apple CEO John Scully introduced the hand-held Newton, saying that portability (flexibility) was the wave of the future. Unfortunately the usability of the device was lacking, as data entry proved to be more difficult on such a small device, and Apple eventually dropped the line in 1998. It was Palm's breakthrough Graffiti language, which greatly improved handwriting recognition, that eventually allowed users to take advantage of the portability performance gain and revived the PDA market. One can *expand the web* by "pulling" two or more sides at once, e.g. logon sub-systems (security) can welcome users by name and recall their preferences (increase usability).

8.5 Does the WOSP model connect to new developments like agile programming?

If performance has many dimensions not one, specialists in the different dimensions (like security and usability) need to talk to each other. Communication both avoids goal conflict and enables goal synergies. The idea of balanced design or non-linear progress leads one to advocate multi-disciplinary teams, who communicate to resolve design conflicts and develop synergies**.** The WOSP model offers theoretical justification for the recent "agile" and "extreme" software development methodologies.

8.6 Can the WOSP model explain why IS progress is so often unpredicted?

The WOSP model rejects the view that progress occurs on a single linear dimension. Progress in one direction soon gives diminishing returns, as it creates tensions that pull back other areas. As Edward Tenner says, progress can "bite back" (Tenner, 1997). The WOSP model suggests, counter-intuitively, that while a system's strongest aspect(s) may create its current success, its weakest aspect(s) may offer the greatest gain potential. The WOSP area of performance is most increased by extending its shortest extent, while pushing the same old performance dimension can give diminishing returns, e.g. multi-media gaming advances led to predictions of virtual reality games, with users wearing headsets and goggles. Yet the game industry ended up branching out in quite different directions. Games became *connected*, allowing virtual social worlds like the SIM's and massively multi-player online role-playing games (MMORPGs), and game editors that let users *extend* games with new maps and units became popular, e.g. DOOMs Wad files.

8.7 Why is it a *web* of system performance?

Because developing one aspect of performance can degrade another, e.g. more functionality gives more menu options, which means more things to learn, which reduces usability, so functionality and usability seem to conflict. As Tenner notes:

"The more powerful systems have become, the more human time it takes to maintain them, to develop the software, to resolve bugs and conflicts*, to learn new versions, to fiddle with options,"* (Tenner, 1997, p266).

However with *innovation* such conflicts can be reconciled, e.g. clip and paste a graphic is a different function from clip and paste of text, however adding a new graphic clip and paste function does not decrease usability as it uses an existing interface. In general, using existing interfaces (plus context cues) allows more functionality without reducing usability. Consistent user interfaces reduce user cognitive costs, so adding a new function has less effect on usability.

8.8 Why is innovation so important in IT progress?

The need for performance integration explains a strange IT paradox: that later versions of *successful* products can, after "development", actually perform *worse* than the original! Likewise some upgrades, customers feel by common sense, may not increase performance enough to be worth the upgrade cost. Yet in most cases developers work on a key performance dimension, like security. The forgotten requirement is to integrate the advance within the system. The user does not have to accept simple trade-offs. Flexibility need not deny reliability, nor functionality reduce usability, nor Internet connectivity abuse privacy [Whitworth, 2003 #1684]. In the WOSP model apparent opposites, like security and open extendibility, can be *reconciled by innovation*.

8.9 What is the role of environment in the WOSP model?

If performance is the interaction of a system with its environment, then *system performance is not absolute* but relative to the environment the system is in. Hence the web of performance has no "perfect" form. Its best shape will depend on which "right" actions the environment rewards and which "wrong" actions it punishes. This is an important attribute in a model that aims to describe diverse software systems in a range of problem spaces. Each IS purchaser must assess the WOSP performance criteria for their business, as shown in Table 4, before using the model. Else one may purchase software that is recommended, but has high performance in dimensions of little value to one's business.

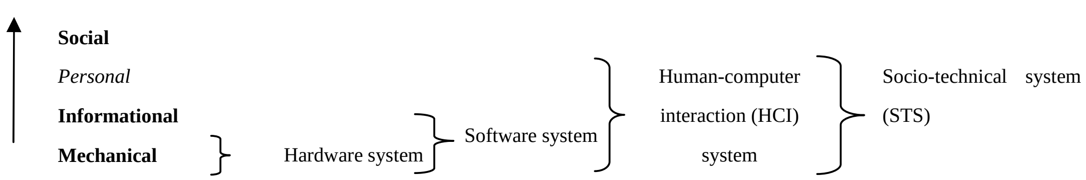
Table 4 WOSP performance criteria evaluation

Dimension	Detail	Value%
Functionality	To act effectively upon the environment.	
Usability	To operate efficiently or easily.	
Security	To resist or avoid outside attack or take-over.	
Extendibility	To use outside components or data.	
Reliability	To avoid or recover from internal failure.	
Flexibility	To change to fit outer circumstances.	
Connectivity	To communicate with other systems.	
Privacy	To control internal information release.	
Performance	To interact successfully with environment.	100%

8.10 Which level should the WOSP model apply to?

The WOSP model can apply to any IS level, but not all at once, as each level is a different problem, e.g., a system may be hardware reliable but software unreliable, or both hardware and software reliable, yet operator unreliable (Sommerville, 2004, p24). To assess reliability, one must first specify the system level, as each level represents a different type of "world". At the lowest level is the physical or mechanical world, which as hardware is the foundation of any information system. From hardware we build software, yet software design is quite different from hardware design. The one involves voltages and circuits while the other involves data flows and entity relationships, or objects and method calls, depending on your programming paradigm. Being derived from a general systems perspective, the WOSP model applies to either level, but not to both at once. Adding people creates another HCI level, which adds cognitive processes.

The fourth level in the hierarchy is an interesting one, as it is the future. It suggests *socio-technical systems* can arise with all four levels together, when cognitive/social interaction is mediated by information technology, rather than the natural world. These levels are not different systems but different views of the same system (though not all systems have all levels). In general, higher levels are higher "views", yet each level creates the next, as mechanical actions create information, information creates personal meaning, and personal meanings create social norms, structures and cultures: "*Each level "emerges" from the previous, and adds a new level of causality. Information derives from mechanics, cognitions from information, and society from individual cognitions.*" (Whitworth, 2003).



9.WHY DISTINGUISH DIFFERENT SYSTEM LIVELS?

Some examples of WOSP dimensions by level are given in Table 5. Each level is not only a more powerful way to *describe* a system, but also a more powerful way to *operate* it, i.e. the potential gains increase with each level. However so do the system requirements. The system levels reflect the progression of IS development. Computing began with hardware, then created commercial software, then developed the "personal" computer (PC). Today's online groups, e-voting, reputation systems, "blogs", and groupware represent the fourth level of social computing. Initially the benefits of powerful hardware seemed immense; however over time we realized that software could give even greater benefits. Modern IS performance successes are more about software than hardware. Right now we are in a period where 'how software works with people' is an issue. In the human-computer interaction period, cognitive resources like attention have become more important than CPU resources. Now to create successful computer systems they must meet usability requirements. It follows that *social* systems offer the greatest productivity gains of all. Generic, non-zero-sum benefits (like sharing scientific research) far outweigh individual personal performance gains (Wright, 2001). *If humanity's greatest "invention" is cooperative society*, technology should take note.

Table 5 WOSP Dimension Examples

Dimension	A Car	IT Hardware	IT Software	HCI/CHI
Functionality	Speed, ability to turn	Chip capacity, memory	Output change rate, frames/sec	Task ability eg to change documents
Usability	Miles per gal, comfort	Heat, power consumption,	"Lite" s/w, less cpu resources to run background	Intuitive s/w, need no manual/training
Security	Locks, keys, door codes	Sealed, secure, insulated	Firewall, virus checks	UserID/Password, bio-id
Extendibility	Tow-bar, roof rack	Standard plugs & connections,	S/w compatible with other s/w	User plug –ins, extensions
Reliability	Starts, maintenance	Uptime, easy to repair	Error recovery	Reduce operator errors
Flexibility	4 wheel drive	Switchable, e.g. 110-240 volts	Platform independence	Control panel, set language
Connectivity	GPS, crash sensors	Network card, comms outputs	Bandwidth, no of connections	Can exchange meaning (email)
Privacy	Detect radar tint windows	Shielded, tempest proof	Encrypt PINS in online buying	Anonymous web surfing

10. CAN YOU GIVE EXAMPLES OF THE SOCIAL LEVEL?

The trend above is that as lower systems-level problems are solved, the performance-level focus rises. Hence software issues began to attract more attention than hardware issues, and human-computer interaction issues such as usability have more recently been coming to the forefront. The level of focus is rising, however, so it is predictable that research and practice will evolve over the next ten years to the socio-technical level. In current terminology, the new "user" of technology will be social organizations, and issues like spam will exemplify "social errors" (Whitworth & Whitworth, 2004). If social organizations become the new information system "user," then social needs like privacy, justice and legitimacy may become IS requirements. For example, when a business purchases an internal messaging system it may require that system to satisfy its social needs, like privacy, or its social structure, which determines who can talk to who. The WOSP model exposes new concerns for the future of technology and society.

10.1 Should only engineers assess technology?

No, because technology now has more than just technical affects. Technology today affects the whole of society, as the Internet illustrates. This is why information systems are at the crossroads of other disciplines, such as engineering, biology, and sociology. Knowledge from many fields can fertilize computing, making it the quintessential, cross-disciplinary discipline. Most of the software advances of the last decade, liked email, chat, "blogs" and "wikis" have been quite simple in engineering terms, but very powerful socially. Businesses that recognize this social dimension, and respect users, will tend to succeed (Whitworth, 2005). The general change is to look for integration as well as excellence. Perhaps we, humans, are a good example of the system that performs well by being balanced. We are not the most powerful animal (i.e., an elephant or lion), nor the most economic (i.e., a plant), nor the most flexible (i.e., a bacterium), nor the most extendible (i.e., a virus), and so on, yet overall we dominate the earth. For only engineers to assess technology is a one sided view, as technology now affects everything.

10.2 What factors beyond the WOSP model apply to assessment?

When investing in an information technology a company has the "make-or-buy" decision to make. To make such decisions, business managers deal with the cost benefit analysis of system acquisition as well as the operational issues involving IT. WOSP measure of performance is primarily concerned with the total area of the web defined by the eight dimensions. Thus in its current form, WOSP does not directly either take system acquisition costs, nor does it take system operation costs into consideration. Similarly, many business strategic goals are also not outside the WOSP model. Examples include factors like competitive advance, alignment and customer relations, informal information access benefits, information quality, transactional benefits to communication efficiency, systems development and business efficiency, to list a few. WOSP can compare two systems and find both adequate by its eight dimensions, but this technical assessment is only part of a business assessment, i.e. which is superior for business goals. These and other important factors of management IT assessment (especially IT investment issues), can be integrated with the WOSP model.

10.3 What methods can be used to assess such factors?

A number of methodologies consider cost along with other tangible and intangible benefits, in addition to information system risk factors (Sylla and Wen, 2002). The evaluation methodologies of tangible benefits critical to business operations include: (1) return on investment, (2) cost-benefit analyses, (3) return on management and (4) information economics. These and similar well established procedures for business IT assessment can be combined with the WOSP technical level assessment.

Intangible benefits and risk factors require more elaborate assessment procedures since they deal with complex subjective factors, but tools based on multi-objective and multi-criteria value analysis and critical success factors can be designed to integrate intangible benefits with the WOSP criteria. Sylla and Wen (2002) present a list of all such methodologies showing their advantages and limitations for business IT assessment. Examples include procedures such as real option and the Delphi approach. Finally, most IT selection is a multi-criteria problem involving a group of decision makers, so theories of negotiation help in making tradeoffs among the multiple decision makers. Research is needed to integrate the WOSP model with other methodologies to help management make a comprehensive evaluation new information systems, with criteria beyond the traditional functionality, usability and cost factors. The authors are currently integrating procedures for comparing alternative system choices, such as the Analytic Process Hierarchy (AHP) with WOSP (Whitworth et al., 2006).

11. HAVE YOU VALIDATED THE MODEL?

Preliminary research shows that users recognize the WOSP attributes when evaluating software (Mahinda & Whitworth, 2005). A recent follow-up study using AHP found much the same results (see http://brianwhitworth.com/wospahp.doc). In particular, when evaluating Internet browsers, users put security and privacy above functionality and usability when making an assessment. The old view that performance is just functionality plus usability cannot be sustained. We are currently developing a user checklist for the WOSP dimensions, but one issue is that the terms used vary with the type of software.

11.1 Can you give an example of software that is unbalanced?

Yes, "Mr Clippy", Office's automatic help, should have been by current theories, like TAM, a smash success. He followed all the principles of the current theories, being a cute and friendly little paper clip, who was there to help. His most famous line: "*It looks like your writing a letter…*" came whenever you typed "Dear …". At that point, Mr Clippy took charge, and was hard to get rid of. Despite 25,000 hours of user testing (Horvitz, 2004), and a Bayesian logic basis (which Microsoft still sees as the future of smart help), Mr. Clippy was so notable a failure that his *removal* was a Windows XP sales pitch (Levitt, 2001). According to a 2001 PC

Magazine survey, Mr. Clippy was the third biggest software flop of the year, with same concept Microsoft Bob as the first (PCMagazine, 2001). He was replaced by the more polite smart tags and task panes, which recognized users' rights to control their own private virtual work-space. Privacy is probably the "sleeper" dimension of the WOSP model, though it is often misunderstood. The equivalent of human social privacy in the animal kingdom is camouflage, and in the military setting, it is stealth weapon systems.

11.2 Can you give an example of successful multi-dimensional development?

Perhaps the best example is the Word Wide Web. Performance integration explains why the web was so successful. Berners-Lee chose HTML as the WWW language because it was simple enough to work on any computer (Berners-Lee, 2000). The Hypertext academic community considered HTML a "simple tag language", much less powerful than Standard Generalized Markup Language (SGML). Yet HTML's simplicity may have been a critical factor in its success. Berners-Lee designed the WWW to be a communication system that was also both scalable (extendible) and usable, while still retaining the functionality it had had for the 20 years before it became popular. The result was a system that connected people as never before. The greatest information system achievement of the last decade only "grew up" when Berner's Lee made it scalable, usable, and somewhat standard. That Berners-Lee's World Wide Web project was rejected by CERN, the Hypertext community, and Microsoft, before becoming the way of the future says something about our current one-track view of progress and performance (Berners-Lee, 2000). In assessing software performance, it is time to recognize integration as well as specialization, and balance as well as excellence.

REFERENCES

[1]Alexander, C. (1964). *Notes on the Synthesis of Form*. Cambridge, Ma: Harvard University Press.

[2]Alexander, C. (2005). Design Patterns. 2005, from http://www.patternlanguage.com/bios/designpatterns.htm

[3]Alter, S. (1999). A general, yet useful theory of information systems. *Communications of the AIS, 1*(March), 13-60.

[4]Berners-Lee, T. (2000). *Weaving The Web: The original design and ultimate destiny of the world wide web*. New York: Harper-Collins.

[5]Bertalanffy, L. v. (1968). *General System Theory*. New York: George Braziller Inc.

[6]Chung, L., Nixon, B. A., Yu, E., & Mylopoulos, J. (1999). *Non-functional requirements in Software Engineering*. Boston: Kluwer Academic.

[7]Churchman, C. W. (1979). *The Systems Approach*. New York: Dell Publishing.

[8]Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *Management Information Systems Quarterly, Sep*.

[9]Hitt, L. M., & Brynjolfsson, E. (1996). Productivity, business profitability, and consumer surplus: three different measures of information technology value. *MIS Quarterly, 20*(2), 121-142.

[10]Horvitz, E. (2004). Lumiere Project: Bayesian Reasoning for Automated Assistance. 2004, from http://reserach.microsoft.com/~horvitz/lum.htm

[11]Levitt, J. (2001). Internet Zone: Good help is hard to find. 2004, from http://www.informatonweek.com/835/35uwjl.htm

[12]Mahinda, E., & Whitworth, B. (2005, August 11-14). *The Web of System Performance: Extending the TAM Model*. Paper presented at the AMCIS (Americas Conference on Information Systems) August 11-14, Omaha, Nebraska, USA.

[13]PCMagazine. (2001). 20th Anniversary of the PC Survey Results. 2004, from http://www.pcmag.com/article2/0,1759,57454,00.asp

[14]Rivard, S., Poirier, G., Raymond, L., & Bergeron, F. (1997). Development of a measure to assess the quality of user-developed applications. *The DATA BASE for Advances in Information Systems, 28*(3), 44-58.

[15]Schor, J. B. (1991). *The overworked American: The unexpected decline of leisure*. New York: Basic Books.

[16]Seddon, P. B., Graeser, V., & Willcocks, L. P. (2002). Measuring organizational IS effectiveness an overview and update

of senior management perspectives. *ACM SIGMIS Database, 33*.

[17]Sommerville, I. (2004). *Software Engineering* (7th ed.).

[18]Sylla, C. & Wen, J., (2002). A conceptual framework for evaluation of information technology investments. *International Journal of Technology Management*, 24, 2/3, 236-261.

[19]Tenner, E. (1997). *Why Things Bite Back*. New York: Vintage Books, Random House.

[20]Toffler, A. (1980). *The Third Wave*.

[21]Westland, J. C., & Clark, T. H. K. (2000). *Global Electronic Commerce*. Cambridge, MA: MIT Press.

[22]Whitworth, B., Banuls, V., Sylla, C., Mahinda. E.. Expanding the Criteria for Evaluating Human-Machine Systems using the Web of System Performance, submitted for publication to IEEE Transactions on Systems, Man and Cybernetics, 2006.

[23]Whitworth, B. (2005). Polite Computing. *Behaviour & Information Technology,* (5, September, http://brianwhitworth.com/polite.rtf), 353 – 363.

[24]Whitworth, B., Fjermestad, J., & Mahinda, E. (2005). The Web of System Performance: A multi-goal model of information system performance. *Communications of the ACM, in press*(http://brianwhitworth.com/WOSP05.doc).

[25]Whitworth, B., & Whitworth, E. (2004). Reducing spam by closing the social-technical gap. *Computer*(October), 38-45.

[26]Whitworth, B., & Zaic, M. (2003). The WOSP Model: Balanced Information System Design and Evaluation. *Communications of the Association for Information Systems, 12*, 258-282.

[27]Wright, R. (2001). *Nonzero: The logic of human destiny*. New York: Vintage Books.