

A Brief Introduction to Sociotechnical Systems

Brian Whitworth

Massey University Auckland, New Zealand

INTRODUCTION

The term sociotechnical was introduced by the Tavistock Institute in the 1950's for manufacturing cases where the needs of technology confronted those of local communities, for example, longwall mining in English coalmines (see <http://www.strategosinc.com/socio-technical.htm>). Social needs were opposed to the reductionism of Taylorism, which broke down jobs on say a car assembly line into most efficient elements. Social and technical were seen as separate side-by-side systems which needed to interact positively, for example, a village near a nuclear plant is a social system (with social needs) besides a technical system (with technical needs). The sociotechnical view later developed into a call for ethical computer use by supporters like Mumford (Porra & Hirscheim, 2007).

In the modern holistic view the sociotechnical system (STS) is the whole system, not one of two side-by-side systems. To illustrate the contrast, consider a simple case: A pilot plus a plane are two side-by-side systems with different needs, one mechanical (plane) and one human (pilot). Human Computer Interaction (HCI) suggests these systems should interact positively to succeed. However plane plus pilot can also be seen as a single system, with human and mechanical levels. On the mechanical level, the pilot's body is as physical as the plane, for example, the body of the plane and the body of the pilot both have weight, volume, and so forth. However the pilot adds a human thought level that sits above the plane's mechanical level, allowing the "pilot + plane" system to strategize and analyze. The sociotechnical concept that will be developed changes the priorities, for example, if a social system sits next to a technical one it is usually secondary, and ethics an afterthought to mechanics, but when a social system sits above a technical one it guides the entire system, that is, the primary factor in system performance.

BACKGROUND

General Systems Theory

Sociotechnical theory is based upon general systems theory (Bertalanffy, 1968), which sees systems as composed of autonomous yet interdependent parts that mutually interact

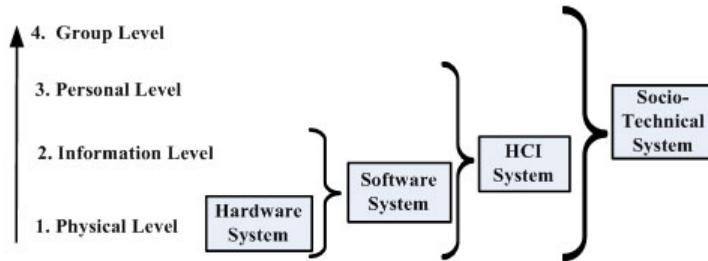
as part of a purposeful whole. Rather than reduce a system to its parts, systems theory explores emergent properties that arise through component interactions via the dynamics of regulation, including feedback and feed-forward loops. While self-reference and circular causality can give the snowball effects of chaos theory, such systems can self-organize and self-maintain (Maturana & Varela, 1998).

System Levels

In the 1950's-1960's computing was mainly about hardware, but the 1970's introduced the software era with business information processing. The 1980's then gave personal computers, adding people into the equation, and email in the 1990's introduced the computer as a social medium. In this decade social computing development continues, with chat rooms, bulletin boards, e-markets, social networks (e.g., UTube, Facebook, MySpace), Wikis and Blogs. Each decade computing has reinvented itself, going from hardware to software, from software to HCI, and now from HCI to social computing. The concept of system levels frames this progression. While Grudin initially postulated three levels of hardware, software and cognitive (Grudin, 1990), Kuutti later added an organizational level (Kuutti, 1996), suggesting an information system (IS) could have four levels: hardware, software, human, and organizational (Alter, 1999). Just as software "emerges" from hardware, so personal cognitions can be seen as arising from neural information exchanges (Whitworth, 2008), and a society can emerge from the interaction of individual people. If the first two levels (hardware/software) are together considered technical, and the last two (human/group) social, then a sociotechnical system is one that involves all four levels (Figure 1).

As computing evolved, the problems it faced changed. Early problems were mainly hardware issues, like over-heating. When these gave way to software problems like infinite loops, then network and database needs began to influence hardware chip development. A similar progression occurred as human factors emerged, and human requirements like usability became part of software engineering (Sanders & McCormick, 1993). Social computing continues this trend, as social problems beyond HCI theory (Whitworth, 2005) are now important in design. Driving this evolution is that each emergent level increases system performance (Whitworth & deMoor, 2003).

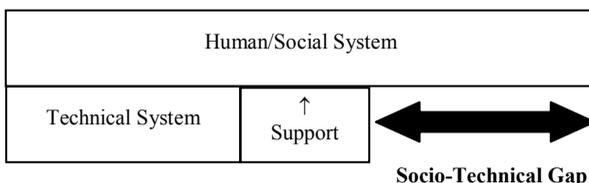
Figure 1. Sociotechnical system levels



The Sociotechnical Gap

The Figure 1 levels are not different systems but overlapping views of the same system, corresponding to engineering, computing, psychological, and sociological perspectives respectively (Whitworth, Fjermestad, & Mahinda, 2006). Higher levels are both more efficient ways to describe a system and also more efficient ways to operate it, for example, social individuals produce more than they would acting independently. Whether a social system is electronically or physically mediated is arbitrary. That the communication medium (a computer network) is “virtual” does not make the people involved less real, for example, one can be as upset by an e-mail as by a letter. Sociotechnical systems are systems of people communicating with people that arise through interactions mediated by technology rather than the natural world. The social system can follow the same principles whether on a physical or electronic base, for example, friendships that cross seamlessly from face-to-face to e-mail interaction. However in physical society, physical architecture supports social norms, for example, you may not legally enter my house and I can also physically lock you out or call the police to restrain you. In contrast, while in cyberspace the “architecture” is the computer code itself, that “... makes cyberspace as it is” (Lessig, 2000), that code is largely designed without any reference to social needs. This sociotechnical gap (Ackerman, 2000), between what computers do and what society wants (Figure 2), is a major software problem today (Cooper, 1999).

Figure 2. Socio-technical gap



STS REQUIREMENTS

System Theory Requirements

A systems approach to performance suggests that systems can adapt the four elements of a boundary, internal structure, effectors, and receptors to either increase gains or reduce losses (Whitworth et al., 2006), where:

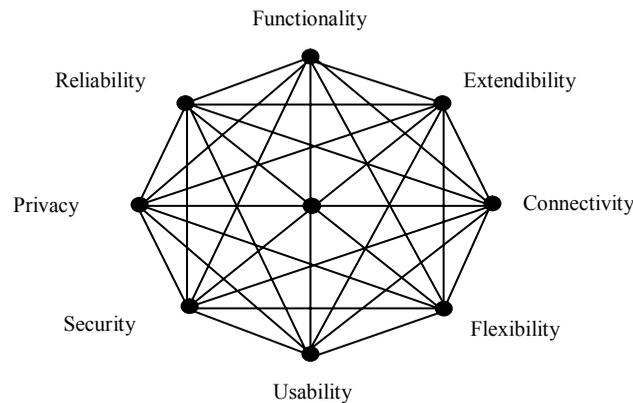
1. The system boundary controls entry, and can be designed to deny unwelcome entry (security), or to use the entity as a “tool” (extendibility).
2. The system internal structure manages system operations, and can be designed to reduce internal changes that cause faults (reliability), or to increase internal changes allowing environment adaptation (flexibility).
3. The system effectors use system resources to act upon the environment, and can be designed to maximize their effects (functionality), or to minimize the relative resource “cost of action” (usability).
4. The system receptors open channels to communicate with other systems, and can enable communication with similar systems (connectivity), or limit communication (privacy).

This gives the eight performance goals of Figure 3, where:

- The Web area shows the overall system’s performance potential.
- The Web shape shows the system’s performance profile, for example, risky environments favor secure profiles.
- The Web lines show tensions between goals, for example, improving flexibility might reduce reliability.

The goals change for different system levels, for example, a system can be hardware reliable but software unreliable,

Figure 3. The Web of system performance



or both hardware and software reliable but operator unreliable (Sommerville, 2004, p. 24). Likewise usability (the relative cost of action) can mean less cognitive “effort” for a person in an HCI system, or for a software system mean less memory/processing (e.g., “light” background utilities), or for a hardware system mean less power use (e.g., mobile phones that last longer). From this perspective, the challenge of socio-technical computing is to design innovative systems that integrate the multiple requirements of system performance at higher and higher levels, where each level builds upon the previous.

Information Exchange Requirements

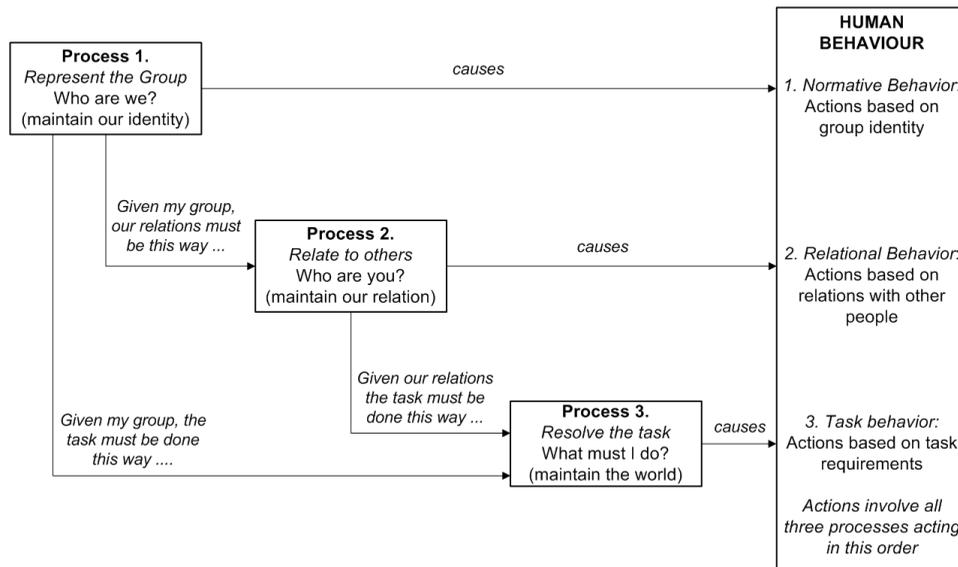
The connectivity-privacy tension line (Figure 3) introduces a social dimension to system design, as information exchanges let the systems we call people combine into larger systems, for example, people can form villages, villages can form states, and states can form nations. In this social evolution not only does “the system” evolve, but also what we define as “the system” evolves. Theories of computer-mediated information exchange postulate the underlying social process. Some consider it a single process of rational analysis (Huber, 1984; Winograd & Flores, 1986), but others suggest process dichotomies, like task vs. socioemotional (Bales, 1950), informational vs. normative (Deutsch & Gerard, 1965), task vs. social (Sproull & Kiesler, 1986), and social vs. interpersonal (Spears & Lea, 1992). A three process model of online communication (Whitworth, Gallupe, & McQueen, 2000) suggests three processes:

1. Factual information exchange: the exchange of factual data or information, that is, message content
2. Personal information exchange: the exchange of personal sender state information, that is, sender context
3. Group information exchange: the exchange of group normative information, that is, group position

The first process involves intellectual data gathering, the second involves building emotional relationships, and the third involves identifying with the group, as proposed by Social Identity theory (Hogg, 1990). The three goals are understanding, intimacy and group agreement, respectively. In this multi-threaded communication model one communication can contain many information threads (McGrath 1984), for example, if one says “I AM NOT UPSET!” in an upset voice, sender state information is analyzed in an emotional channel, while message content is analysed intellectually. A message with a factual content not only lies within a sender state context given by facial expressions etc, but also contains a core of implied action, for example, saying “This is good, lets buy it” gives not only content information (the item is good) and sender information (say tone of voice), but also the sender’s intended action (to buy the item), that is, the sender’s action “position”.

While message content and sender context are generally recognized, action position is often overlooked as a communication channel, perhaps because it typically involves many-to-many information exchange, for example, in a choir singing everyone sends and everyone receives, so if the choir moves off key they do so together. Similarly, in apparently rational group discussions, the group “valence index” predicts the group decision (Hoffman & Maier, 1964). Group members seem to assess where the group is going and change their ideas to stay with the group. This intellectual equivalent of how social flocks/herds cohere seems to work equally well online, so using this process one can generate online agreement from anonymous and lean information exchanges (Whitworth, Gallupe & McQueen, 2001). While factual information suits one-way one-to-many information exchange (e.g., a Website broadcast), and developing personal relations suits two-way one-to-one information exchange (e.g., e-mail), the group normative process needs two-way, many-to-many information exchange (e.g., reputation systems). Figure 4 shows how people prioritize the three cognitive processes, first evaluating group membership

Figure 4. Three information exchange processes



identity, then evaluating the sender’s relationship, and only finally analyzing message content. As each cognitive process favours a different communication architecture, there is no ideal human communication medium. From this perspective, the challenge of sociotechnical computing is to design innovative systems that integrate the multiple channels of human communication.

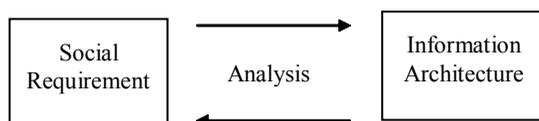
Social Requirements

A startling discovery of game theory was that positive social interaction can be unstable, for example, the “equilibrium point” of the prisoner’s dilemma dyad is that both cheat each other (Poundstone, 1992). Situations like social loafing and the volunteer dilemma are common, including many-to-many cases like the tragedy of the commons which mirrors global conservation problems. In a society one person can gain at another’s expense, for example, theft, yet if everyone steals society collapses into disorder. Human society has evolved various ways to make anti-social acts unprofitable, whether by personal revenge traditions, or by state justice systems. The goal of justice, it has been argued, is to reduce unfairness (Rawls, 2001), where unfairness is not just an unequal

outcome distribution (inequity) but failure to distribute outcomes according to action contributions. In a successful society people are accountable not just for the effects of their acts on themselves but also on others. Without this, people can take the benefits that others create, and harm others with no consequence to themselves. Under these conditions, any society will fail. Fortunately people in general seem to recognize unfairness, and tend to avoid unfair situations (Adams, 1965). They even prefer fairness to personal benefit (Lind & Tyler, 1988). The capacity to perceive “natural justice” seems to underlie our ability to form prosperous societies. The general requirement is legitimate interaction, which is both fair to the parties involved and also benefits the social group (Whitworth & deMoor, 2003). Legitimacy is a complex sociological concept that describes governments that are justified to their people (and not coerced) (Barker, 1990). Fukuyama argues that legitimacy is a requirement for community prosperity, and those that ignore it do so at their peril (Fukuyama, 1992).

It follows that online society should be designed to support legitimate interaction and oppose antisocial acts. Yet defining what is and is not legitimate is a complex issue. Physical society evolved the concept of “rights” expressed in terms of ownership (Freedman, 1991), for example, freedom as the right to own oneself. Likewise analyzing who owns what information online (Rose, 2001), can be used to specify equivalent online rights that designers can support (Whitworth, Aldo de Moor & Liu, 2006) (Figure 5). This does not mechanize online interaction, as rights are choices not obligations, for example, the right to privacy does not force one to be private. From this perspective, the challenge of sociotechnical computing is to design systems that reflect rights in overlapping social groups.

Figure 5. Legitimacy analysis



B

FUTURE TRENDS

The previous discussion suggests the design of future sociotechnical systems will involve:

1. More performance requirements: Simple dichotomies like usefulness and ease of use, while themselves valid, will be insufficient to describe the multiple criteria relevant to STS performance.
2. More communication requirements: Communication models with one or two cognitive processes while themselves valid, will be insufficient to describe the multiple threads of STS communication.
3. More social requirements: Approaches that reference only individual users, while themselves valid, will be insufficient to represent social level requirements, where one social group can contain another.

Performance concepts will expand, as success-creating goals (functionality, flexibility, extendibility and connectivity) and the failure-avoiding goals (security, reliability, privacy and usability) interact in complex ways in a multidimensional performance space that challenges designers. Likewise designing communication media that simultaneously support the flow of intellectual, emotional and positional information along parallel exchange channels is another significant challenge. Finally, in social evolution social systems “stack” one upon another (e.g., states can form into a federation), giving the challenge of groups within groups. In sum, the challenges are great, but then again why should STS design be easy?

The concept of levels (Figure 1) runs through the above trends, suggesting that the World Wide Web will develop cumulatively in three stages. The first stage, a factual information exchange system, seems already largely in place, with the World Wide Web essentially a huge information library accessed by search tools, though it contains disinformation as well as information. The second stage lets people form personal relations to distinguish trusted from not trusted information sources. This stage is now well underway, as social networks combine e-mail and browser systems into protocol independent users environments (Berners-Lee 2000). Finally, in stage three the Web will sustain synergistic and stable online communities, opening to group members the power of the group, for example, the group knowledge sharing of Wikipedia. To do this, online communities must both overcome antisocial forces like Spam and prevent internal take-overs by personal power seekers. In this struggle software cannot be “group blind” (McGrath & Hollingshead, 1993). Online communities cannot democratically elect new leaders to replace old ones unless software provides the tools. Supporting group computing is not just creating a few membership lists. Even a cursory study of Robert’s

Rules of Order will dispel the illusion that technical support for social groups is easy (Robert, 1993).

Human social activity is complex, as people like to belong and relate as well as understand. Each process reflects a practical human concern, namely dealing with world tasks, with other people, and with the society one is within. All are important, because sometimes what you know is critical, sometimes who you know counts, and sometimes, like which side of the road to drive on, all that counts is what everyone else is doing. Yet as we browse the Web software largely ignores social complexity. “Smart” tools like Mr. Clippy have relational “amnesia”, as no matter how many times one tells them to go away they still come back. Essential to relationships is remembering past interactions, yet my Windows file browser cannot remember my last browsed directory and return me there, my Word processor cannot remember where my cursor was last time I opened this document and put me back there, and my browser cannot remember the last Website I browsed and restart from there (Whitworth, 2005). Group level concepts like rights, leadership, roles, and democracy are equally poorly represented in technical design.

These challenges suggest that the Internet is only just beginning its social future. We may be no more able to envisage this than traders in the Middle Ages could conceive today’s global trade system, where people send millions of dollars to foreigners they have never seen for goods they have not touched to arrive at unknown times. To traders in the middle ages, this would have seemed not just technically but also socially impossible (Mandelbaum, 2002). The future of software will be more about social than technical design, as software will support what online societies need for success. If society believes in individual freedom, online avatars should belong to the person concerned. If society gives the right to not communicate (Warren & Brandeis, 1890) so should communication systems like email (Whitworth & Whitworth, 2004). If society supports privacy, people should be able to remove their personal data from online lists. If society gives creators rights to the fruits of their labors (Locke, 1963), one should be able to sign and own one’s electronic work. If society believes in democracy, online communities should be able to elect their leaders. In the sociotechnical approach social principles should drive technical design.

CONCLUSION

The core Internet architecture was designed many years ago at a time when a global electronic society was not even envisaged. It seems long due for an overhaul to meet modern social requirements. In this upgrade technologists cannot stand on the sidelines. Technology is not and cannot be value neutral,

because online code affects online society. Just as physical laws determine what happens in physical worlds, so the “laws” of online interaction are affected by those who write the programs that create it. If computer system developers do not embody social concepts like freedom, privacy and democracy in their code, they will not happen. This means specifying social requirements just as technical ones currently are. While this is a daunting task, the alternative is an antisocial online society which could “collapse” (Diamond, 2005). If the next step of the human social evolution is an electronically enabled world society, computer technology may be contributing to a process of human unification that has been underway for thousands of years.

REFERENCES

- Alter, S. (1999). A general, yet useful theory of information systems. *Communications of the AIS*, 1(March), 13-60.
- Bales, R. F. (1950). A set of categories for the analysis of small group interaction. *American Sociological Review*, 15, 257-263.
- Bertalanffy, L. V. (1968). *General system theory*. New York: George Braziller Inc.
- Cooper, A. (1999). *The inmates are running the asylum-Why high tech products drive us crazy and how to Restore the sanity*. USA.
- Deutsch, M., & Gerard, H. B. (1965). A study of normative and informational influences on social judgement. *Journal of Abnormal and Social Psychology*, 51, 629-636.
- Diamond, J. (2005). *Collapse: How societies choose to fail or succeed*. New York: Viking (Penguin Group).
- Freedman, M. (1991). Rights. *Concepts in social thought*.
- Grudin, J. (1990). *The computer reaches out: The historical continuity of user interface design*. In *Proceedings of the CHI '90, ACM SIGCHI Conference*, Seattle, WA.
- Hoffman, L. R. & Maier, N. R. F. (1964). Valence in the adoption of solutions by problem-solving groups: Concept, method and results. *Journal of Abnormal and Social Psychology*, 69(3), 264-271.
- Hogg, M. A. (1990). *Social identity Theory*. New York: Springer-Verlag.
- Huber, G. P. (1984). Issues in the design of group decision support systems. *Management Information Systems Quarterly*, Sep, 8(3), 195-204.
- Kuutti, K. (1996). Activity theory as a potential framework for human computer interaction research. In B. A. Nardi (Ed.), *Context and consciousness: Activity theory and human-computer interaction*. Cambridge, MA: The MIT Press.
- Locke, J. (1963). An essay concerning the true original extent and end of civil government: Second of “Two Treatises on Government” (1690). In J. Somerville & R. E. Santoni (Eds.), *Social and political philosophy* (pp. 169-204). New York: Anchor.
- Mandelbaum, M. (2002). *The ideas that conquered the world*. New York: Public Affairs.
- Maturana, H. R. & Varela, F. J. (1998). *The tree of knowledge*. Boston: Shambala.
- McGrath, J. E. & Hollingshead, A. B. (1993). Putting the “group” back in group support systems. In L. M. Jessup & J. S. Valacich (Eds.), *Group support systems: New Perspective*. MacMillan.
- Porra, J. & Hirscheim, R. (2007). A lifetime of theory and action on the ethical use of computers. A dialogue with Enid Mumford. *J AIS*, 8(9), 467-478.
- Poundstone, W. (1992). *Prisoner's dilemma*. New York: Doubleday, Anchor.
- Robert, H. M. (Ed.). (1993). *The new Robert's rules of order*. Barnes & Noble.
- Sommerville, I. (2004). *Software engineering* (7th ed.).
- Spears, R. & Lea, M. (1992). Social influence and the influence of the “social” in computer-mediated communication. In M. Lea (Ed.), *Contexts of computer mediated communication* (pp. 30-65). Hemel Hempstead: Harvester Wheatsheaf.
- Sproull, L. & Kiesler, S. (1986). Reducing social context cues: Electronic mail in organizational communication. *Management Science*, 32(11), 1492-1512.
- Warren, S. D. & Brandeis, L. D. (1890). The right to privacy. *Harvard Law Review*, IV(5), 193-220.
- Whitworth, B. (2005). Polite computing. *Behaviour & Information Technology*, 5(September), 353 – 363.
- Whitworth, B. (2008). *Some implications of comparing human and computer processing*. In *Proceedings of the 41st Hawaii International Conference on System Sciences*.
- Whitworth, B. & deMoor, A. (2003). Legitimate by design: Towards trusted virtual community environments. *Behaviour & Information Technology*, 22(1), 31-51.
- Whitworth, B. & Whitworth, E. (2004). Reducing spam by closing the social-technical gap *Computer*, (October), 38-45.
- Whitworth, B., Aldo de Moor, A., & Liu, T. (2006). Towards a theory of online social rights. In Z. T. R. Meersman, P.

Herrero et al. (Eds.), In *Proceedings of the OTM Workshops 2006, LNCS 4277* (pp. 247 - 256). Berlin Heidelberg: Springer-Verlag

Whitworth, B., Fjermestad, J., & Mahinda, E. (2006). The web of system performance: A multi-goal model of information system performance. *Communications of the ACM*, 49, May(5), 93-99.

Whitworth, B., Gallupe, B., & McQueen, R. J. (2000). A cognitive three process model of computer-mediated groups: Theoretical foundations for groupware design. *Group Decision and Negotiation*, 9(5), 431-456.

Whitworth, B., Gallupe, B., & McQueen, R. (2001). Generating agreement in computer-mediated groups. *Small Group Research*, 32(5), 621-661.

Winograd, T. & Flores, F. (1986). *Understanding computers and cognition: A new foundation for design*. Norwood, NJ: Ablex Pub. Corp.

KEY TERMS

Information System: A system that may include hardware, software, people and business or community structures and processes (Alter, 1999), c.f. a social-technical system, which must include all four levels.

Social System: Physical society is not just buildings or information, as without people information has no meaning. Yet it is also more than people. Countries with people of similar nature and abilities, like East and West Germany, perform differently as societies. While people come and go, the “society” continues, for example, we say “the Jews” survived while “the Romans” did not because the people lived on but because their social manner of interaction survived. A social system then is a general form of human interaction that persists despite changes in individuals, communications or architecture (Whitworth & deMoor, 2003).

System: A system must exist within a “world”, as the nature of a system is the nature of the world that contains it, for example, a physical world, a world of ideas, and a social world, may contain physical systems, idea systems and social systems, respectively. A system needs identity to define “system” from “not system”, for example, a crystal of sugar that dissolves in water still has existence as sugar, but is no longer a separate system. Existence and identity seem two basic requirements of any system.

System Elements: Advanced systems have a boundary, an internal structure, environment effectors and receptors (Whitworth et al., 2006). Simple biological systems (cells) formed a cell wall boundary and organelles for internal cell functions. Cells like Giardia developed flagella to effect movement, and protozoa developed light sensitive receptors. People are more complex, but still have a boundary (skin), an internal structure of organs, muscle effectors and sense receptors. Computer systems likewise have a physical case boundary, an internal architecture, printer/screen effectors and keyboard/mouse receptors. Likewise software systems have memory boundaries, a program structure, input analyzers, and output “driver” code.

System Environment: In a changing world, a system’s environment is that part of a world that can affect the system. Darwinian “success” depends on how the environment responds to system performance. Three properties seem relevant: opportunities, threats, and the rate these change. In an opportunistic environment, right action can give great benefit. In a risky environment, wrong action can give great loss. In a dynamic environment, risk and opportunity change quickly, giving turbulence (sudden risk) or luck (sudden opportunity). An environment can be any combination, for example, opportunistic and risky and dynamic.

System Levels: Are physical systems the only possible systems? The term information system suggests otherwise. Philosophers propose idea systems in logical worlds. Sociologists propose social systems. Psychologists propose cognitive mental models. Software designers propose data entity relationship models apart from hardware. Software cannot exist without a hardware system of chips and circuits, but software concepts like data records and files are not hardware. A system can have four levels: mechanical, informational, personal and group, each emerging from the previous as a different framing of the same system, for example, information derives from mechanics, human cognitions from information, and society from a sum of human cognitions (Whitworth et al., 2006).

System Performance: A traditional information system’s performance is its functionality, but a better definition is how successfully a system interacts with its environment. This allows usability and other “non-functional” requirements like security and reliability to be part of system performance. Eight general system goals seem applicable to modern software: functionality, usability, reliability, flexibility, security, extendibility, connectivity and confidentiality (Whitworth et al., 2006).