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

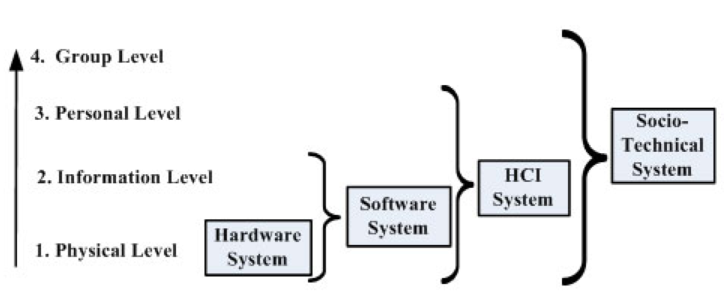
Sociotechnical theory is based upon general systems theory (Bertalanffy, 1968), which sees systems as composed of autonomous yet interdependent parts that mutually interact as part of a purposeful whole. Rather than reduce a system to its parts, systems theory explores emergent properties that arise 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).

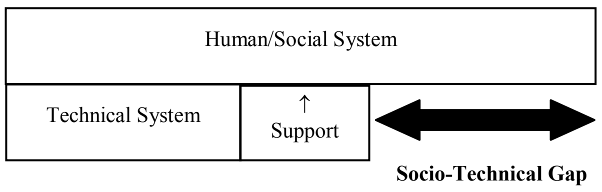
Figure 1. Sociotechnical system levels



The Sociotechnical Gap

The Figure1 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,