BRIAN WHITWORTH WITH ADNAN AHMAD

# THE SOCIAL DESIGN OF TECHNICAL SYSTEMS

## BUILDING TECHNOLOGIES FOR COMMUNITIES

INTERACTION DESIGN FOUNDATION

BRIAN WHITWORTH WITH ADNAN AHMAD

# THE SOCIAL DESIGN OF TECHNICAL SYSTEMS

INTERACTION DESIGN FOUNDATION
Est. 2002

TITLE: The Social Design of Technical Systems: Building technologies for communities. 2nd Edition

AUTHORS: Brian Whitworth and Adnan Ahmad

PUBLISHER: The Interaction Design Foundation

EDITION NUMBER: 2

ISBN PAPERBACK EDITION: 87-92964-09-5

ISBN ONLINE EDITION: 87-92964-10-9

ISBN EBOOK EDITION: 87-92964-11-7

COPY EDITOR: Ailsa Campbell

REFERENCE VALIDATION: Armin Walinsky

GRAPHICS, INTERIOR/EDITORIAL DESIGN, AND COVER: Ramesh Kumar and Mads Soegaard

COPYRIGHT CLEARANCE: Michael Thorman

PROOFREADER: Ailsa Campbell

SUPERVISORY EDITORS: Mads Soegaard and Rikke Friis Dam

EDITORIAL ASSISTANT: Soeren Dam

TYPOGRAPHY: This text is set in Georgia

COPYRIGHT: See section on Copyright

PEER REVIEW: This book has undergone double-blinded peer-review based on the reviewing guidelines

# Table of Contents

# About this book

"*For Audrey, Alex, Elaine, Elizabeth and the next generations*"

A socio-technical system (STS) is a social system operating on a technical base. Email, chat, bulletin boards, blogs, Wikipedia, eBay, Twitter, Facebook and YouTube are all socio-technical systems. Hundreds of millions of people use them every day, but how do they work? More importantly, can we build them better?

This book is for courses on socio-technical system design. The field changes every year, so each chapter has questions for students to discuss, try out online and report back to the class. The general issues raised may interest users, designers and managers of social software, or anyone interested in social computing.

The book questions the view that computing is smart and people are dumb. It says that people in a community are ethical not because they foolishly ignore their self-interest but because they intuitively see community gains beyond themselves. As social standards like privacy, freedom, democracy, ownership and transparency can guide online communities, so online successes like Wikipedia suggest new social forms for physical society.

Information technology, like fire, is a good servant but a bad master. People and computers are better than people or computers only if people are the senior

partner. Social computing takes human centered computing a step further, by introducing the community level. The aim of socio-technical design is no less than to civilize the Internet because the future is socio-technology not technology.

We, the authors, hope you, the reader, find our book interesting and useful. Don't be daunted by the breadth of ideas covered. If both computing and society are complex, how could they together be any less? As people are today building socio-technical systems, there is a need for socio-technical design. The ideas presented in this book have been developed over the last decade. They work. They are proven perennials in the ever changing garden of computing.

*Brian Whitworth,*

*January, 2014*

# CHAPTER 1

# The Evolution of Computing

"*Computing is evolving to higher levels*"

This chapter reviews how computing has evolved since it began, and what this means for "us all" who are building and using it.

## 1.1 A (VERY) SHORT HISTORY OF COMPUTING

The first computer was conceived as a machine of cogs and gears (Figure 1.1) but only became practical in the 1950s and 60s with the invention of semi-conductors. In the 1970s, a *hardware* company called IBM[1] emerged as the computing leader. In the 1980s, however, software became increasingly important, and by the 1990s

1. IBM stands for International Business Machines

a *software* company called Microsoft[2] had become the computing frontline leader by giving ordinary people tools like word-processing. During the 1990s computing became more *personal,* until the World-Wide-Web turned Internet URLs into web site names that people could read[3]. A company called Google[4] then offered the ultimate personal service, free access to the vast public library we call the Internet, and soon everyone's gateway to the web was the new computing leader. In the 2000s computing evolved yet again, to become a *social* medium as well as a personal tool. So now Facebook challenges Google, as Google challenged Microsoft, and as Microsoft challenged IBM.

Yet to design a computer system one must define it, so what *is* computing? This deceptively simple question requires many answers because computing has re-invented itself every decade or so (Figure 1.2). What began as hardware became about software, then about users and is now about online communities. This chapter analyzes the evolution of computing as it impacts computing design.

2. Microsoft stands for microcomputer software

3. IP addresses like 208.80.154.225 became Uniform Resource Locator (URL) names like http://en.wikipedia.org/

4. Google stands for a 1 followed by 100 zeros, i.e. a very large number

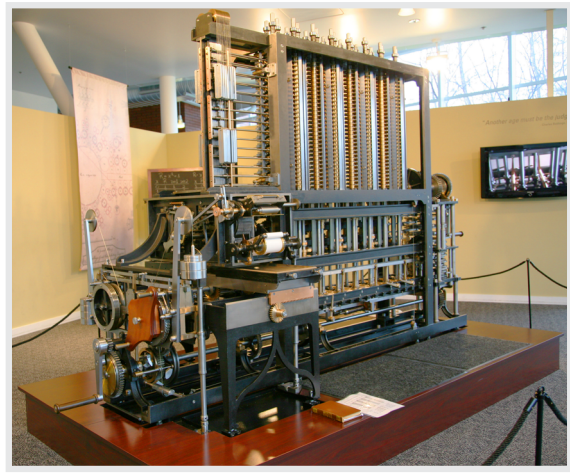
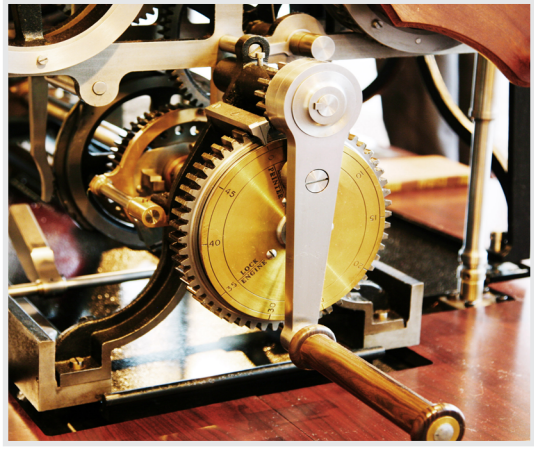


Figure 1.1: Charles Babbage (1791-1871) designed the first automatic computing engine. He invented computers but failed to build them. The first complete Babbage Engine was completed in London in 2002, 153 years after it was designed. Difference Engine No. 2, built faithfully to the original drawings, consists of 8,000 parts, weighs five tons, and measures 11 feet. Shown above is Serial Number 2, located in Silicon Valley at the Computer History Museum in Mountain View, California.

*Courtesy of Jitze Couperus. Copyright: CC-Att-SA-2 (Creative Commons Attribution-ShareAlike 2.0 Unported).*



*Courtesy of Jitze Couperus. Copyright: CC-Att-SA-2 (Creative Commons Attribution-ShareAlike 2.0 Unported).*

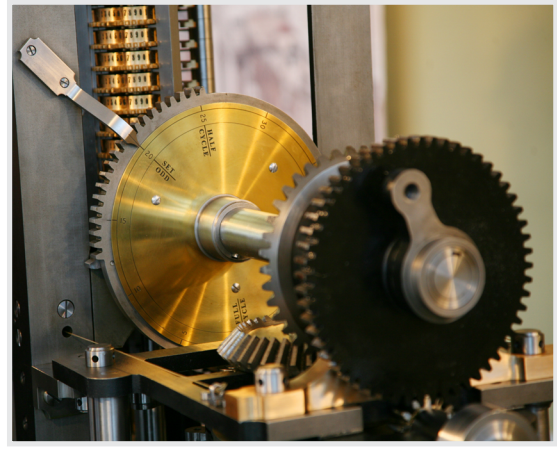


Figure: Details from Babbage's difference engine.

*Courtesy of Jitze Couperus. Copyright: CC-Att-SA-2 (Creative Commons Attribution-ShareAlike 2.0 Unported).*

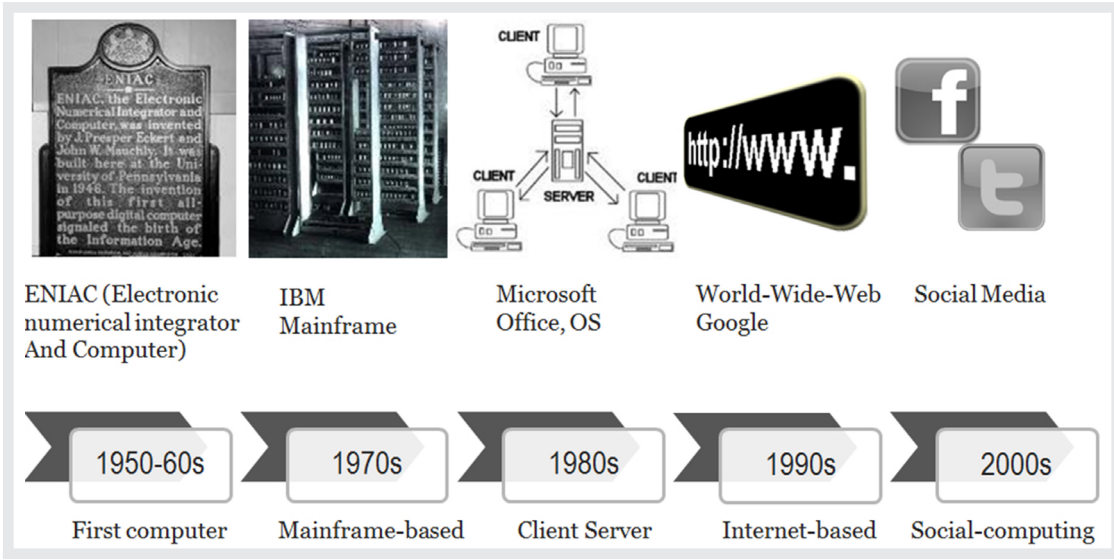



FIGURE 1.2: The computing evolution.

*Courtesy of Brian Whitworth and Adnan Ahmad. Copyright: CC-Att-SA-3 (Creative Commons Attribution-ShareAlike 3.0).*

## 1.2 COMPUTING LEVELS

The evolution of computing is approached here using Bertalanffy's *general systems theory* (Bertalanffy, 1968). This theory is based on the observation of discipline isomorphisms, when different specialist fields discover the same abstract equation or law in different contexts, e.g. a social agreement measure that matches a biological diversity measure (Whitworth, 2006). Bertalanffy proposed a "science of sciences", namely the study of systems in general, since sociologists study social systems, psychologists cognitive systems, computer scientists information systems, and engineers hardware systems. The isomorphisms of science are then general system rules that apply across disciplines.

Applying general systems theory to the evolution of computing gives the computing levels shown in Figure 1.3, where a computing system can be studied

as a mechanical system, a software system, a human system or a social system, by engineers, computer scientists, psychologists and sociologists respectively. Computing began at the mechanical level, added an information level (software), then a human level and finally a community level; it is an example of general system evolution. Stamper's (1993) semiotic onion model depicts similar levels within an organization, of technical, formal (informational rules) and informal (human meaning).

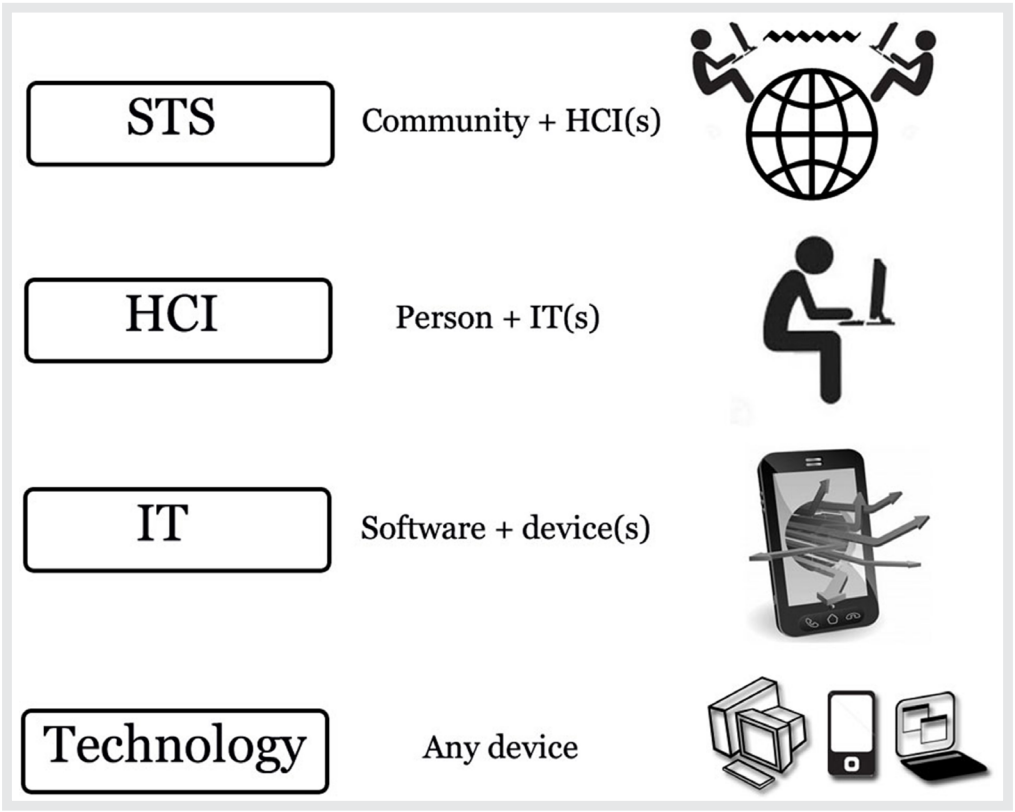



Figure 1.3: Computer system levels.

*Courtesy of Brian Whitworth and Adnan Ahmad. Copyright: CC-Att-SA-3 (Creative Commons Attribution-ShareAlike 3.0).*

Table 1.1 shows how different levels change not only the system type but also what is exchanged. So a physical system exchanges energy, a software system exchanges information, a human system exchanges meaning, and a community system exchanges memes[5]. Each level of a system evolution is built on the previous, so that social computing emerges from personal computing, personal computing emerges from software, and software emerges from hardware. As computing evolves to higher system levels, so its design also changes, from technical to socio-technical design.

Levels can clarify the often confusing terms of computing. In Figure 1.3, a *technology* is any tool that people build to use[6], e.g. a spear is a technology. Technology is designed and built by engineers. In contrast *information technology* (IT) is the application of hardware and software, with a user implied. *Computer science* (CS)[7] is then just about the software level, regardless of the hardware implementation. So information technology is not a sub-set of technology, nor is computer science a sub-set of engineering (because software is not part of hardware).

*Human computer interaction* (HCI) is a person in front of a computer, a human plus IT system, with physical, informational and psychological levels. Just as IT is more than hardware, so HCI is more than IT because it includes a human level. HCI systems exchange meaning, while IT systems exchange information. The semantic web vision of Tim Berners-Lee, the founder of the World Wide Web — his "dream for the web" — was a recognition of the human level of computing.

Today, computing supports online communities that, by the same logic, have hardware, software, personal and community levels. If the first two levels are technical and the last two social, the result is *a socio-technical system* (STS). If information technology design is computing built to hardware and software require-

5. A meme is a common idea, behaviour or style communicated within a culture.

6. Anything we build and use physically is technology, e.g. a table is technology.

7. The study of information processing.

ments, then socio-technical design is computing built to personal and community requirements *as well*. In socio-technical design, the new "user" of computing is the community (Whitworth, 2009a).

Unfortunately, different disciplines use different terms for the same levels, e.g. the study of software can be called computer science or software engineering. The human level of computing is even more confusing: engineers use the term IT to refer to user applications; business prefers the term information systems (IS); education uses information communication technology (ICT); and health professionals invented the term informatics to meet their needs. Each defines itself apart, but in this pan-discipline view, all are just the human level of computing. This book uses the term HCI for consistency[8].

| Level | Exchange | Examples | Design |
|---|---|---|---|
| *Community* (sociology) | Memes | Norms, culture, laws, zeitgeist, sanctions, roles | STS |
| *Personal* (psychology) | Meaning | Semantics, attitudes, beliefs, feelings, ideas | HCI |
| *Information* (computer science) | Information | Programs, data, bandwidth, memory | IT |
| *Mechanical* (engineering) | Energy | Hardware, motherboard, telephone, FAX | Technology |

Table 1.1: The levels of computing.

8. An alternate term is CHI, or computer-human interaction.

If *all* the Figure 1.3 levels are computing we must design computer products as both social *and* technical systems. Limiting computing to hardware (engineering) or software (computer science) denies its obvious evolution.

Levels in computing are not system parts. To draw an analogy, a pilot flying an aircraft is *one* system with different levels, *not* a mechanical part (the aircraft) with a human part (the pilot). The physical level includes not just the aircraft body but also the pilot's body, as both have mass, volume, inertia etc. Likewise, the information level is not just the onboard computer, but also the neuronal processing of the pilot's brain.

The human level is the pilot, who from the sensations and perceptions of his or her brain generates meaning. To the pilot, the aircraft is an extension of his or her body, like extra hands or feet, and computer data is like extra eyes or ears. On a human level, the pilot is the actor, with the aircraft just a mechanical tool of the pilot's will, so in an aerial conflict, the tactics of a piloted aircraft are different from a computer drone.

To repeat, the mechanical level is not just the physical aircraft but also the pilot's body, and the information level is all the processing, of both the brain and of onboard computers. Finally, an aircraft in a squadron may do things it would not do alone, e.g. expose itself as a decoy so that others can attack the enemy.

## 1.3 LEVELS AS WORLD VIEWS

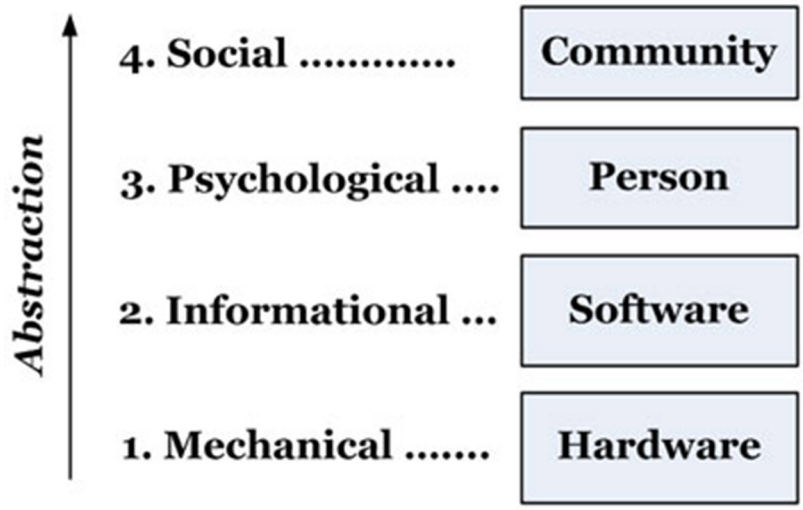



FIGURE 1.4: Levels as higher abstractions.

*Courtesy of Brian Whitworth and Adnan Ahmad. Copyright: CC-Att-SA-3 (Creative Commons Attribution-ShareAlike 3.0).*

A level is now formally defined as a world view, a way of *seeing* reality that is complete and consistent in itself. So a computer on the mechanical level is *all* hardware but on an informational level it is *all* software. We cannot observe a program on a motherboard nor a hardware device in a data structure. Your mobile phone does not have separate hardware and software parts, but is hardware or software *in toto,* depending on how you view it.

Hardware and software are therefore ways to view a computer system, not ways to divide it up. Hardware becomes software when we see computing in a different way. The switch is like swapping glasses to see the same thing differ-

ently. Hardware and software are the different types of glasses we can use to view computing.

Following this theory, all the disciplines of science are world views, like walking around an object to see it from different perspectives. We can then choose which view is most appropriate. For example, to describe World War II as a scientific history of atomic events would be ridiculously ineffective, as a political summary is the more appropriate view.

As views, levels *emerge* from each other, as lower abstractions give higher ones (Figure 1.4). Information emerges from hardware options, meaning emerges from information flows, and communities emerge from common citizen meanings. Conversely, without physical choices there is no information, without information there is no meaning, and without meaning there is no community[9].

A world view is:

1. *Essential*. To view a world one needs a view perspective.
2. *Empirical*. It arises from interaction with the world.
3. *Complete*. It consistently describes a whole world.
4. *Subjective*. We choose a view perspective, explicitly or not.
5. *Exclusive*. You cannot view the world in two different ways at the same time, as you cannot sit in two places at once[10].
6. *Emergent*. One world view can emerge from another.

Note that a level as a view must be chosen before viewing, i.e. *first pick a level, and then view*.

Levels affect design because how we see the world affects how we act in it; e.g. if we saw ultra-violet light, as bees do, previously dull flowers would become bright and so every flower shop would want to change its stock. *Levels as higher*

9. A community is defined as a set of people who see themselves as a social unit.

10. In the same way as you cannot lever on two fulcrums at once. We can, of course, view from one perspective and then another.

*ways to view a system are also new ways to operate and design the system.* Hence new software protocols like Ethernet can improve network performance as much as new cables.

Levels have also changed how business works in computing. When the software era arrived, hardware continued to evolve but hardware leaders like IBM no longer dominated computing unilaterally, as they had before. The software level changed business fortunes by changing what computing *is*. Selling software makes as much money as selling hardware, as the software changes more rapidly and needs to be replaced or updated more often. Web queries are even more volatile, so Google gave its service away for free and then sold advertising around it — it sold its services to those who sold theirs.

As computing levels changed, so did the business model, as selling knowledge is not like selling software. Facebook is still working out its business model, because you cannot "sell" friendships as you do hardware and software. Yet Facebook now challenges Google because we *relate* to family and friends even more than we *query* knowledge — social exchange has as much trade potential as knowledge exchange.

New ways to view computing thus affect how we design and build computing systems. Each level emerges to add to rather than replace earlier levels. As design requirements cumulate, socio-technical design includes hardware, software and human requirements, as well as community needs (Figure 1.5). Computing that *appears* to us as just hardware now has a social level; e.g. smart-phones are a communication medium as well as a hardware device. Computer design is *inclusively* evolving from engineering design to socio-technical design, to keep pace with computer evolution.

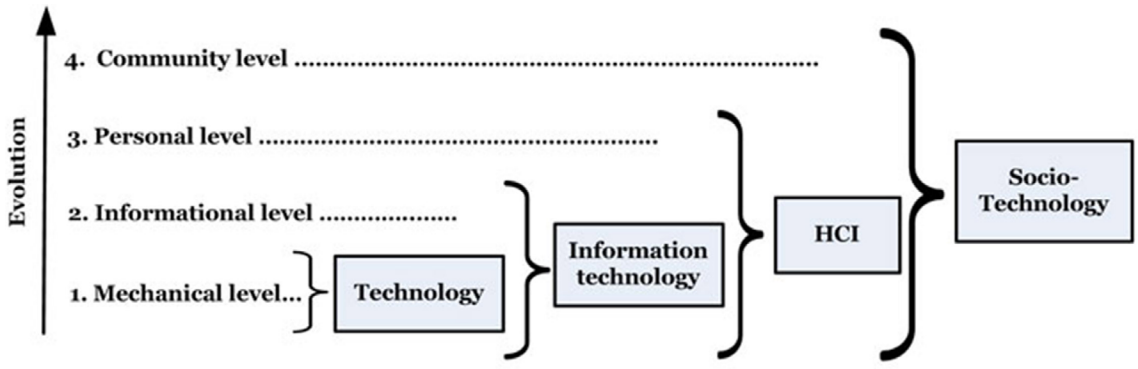



FIGURE 1.5: Computing levels cumulate.

*Courtesy of Brian Whitworth and Adnan Ahmad. Copyright: CC-Att-SA-3 (Creative Commons Attribution-ShareAlike 3.0).*

It is worth reflecting on how the role of technology has changed in a few centuries. During the industrial revolution, technology was isolated from the needs of society; e.g. a village beside a factory belching smoke found its need for clean air threatened. The idea of socio-technology arose when technology was implemented with ethics as an after-thought.

However, in the information revolution the social and technical merge into one. If social needs are not met online, there will be no online community, which means the technology fails. In socio-technical design, social needs as the higher level always come first. In the evolution of design, higher level requirements have a natural priority, as is now discussed.

## 1.4 FROM HARDWARE TO SOFTWARE

Hardware is any physical computer part, e.g. mouse, screen or case. It does not "cause" software, and nor is software a hardware output, in the way that physical systems have physical outputs. *We* create software by seeing information choices in physical events. Software needs hardware but the same code can run on a PC, Mac or mobile phone. An entity relationship diagram can work for any physical storage, whether disk, CD or USB, as data entities are not disk sectors. Software assumes some hardware but does not specify which hardware.

If any part of a device acquires software, the whole system gets an information level; e.g. a computer is information technology even though its case is just hardware. We describe a system by its highest level, so if the operating system "hangs"[11] we say "the computer" crashed, even though the *computer hardware is working fine*. Rebooting fixes the software problem with no hardware change, so a software system can fail while the hardware still works perfectly.

Conversely, a computer can fail as hardware but not software, if a chip overheats. Replace the hardware part and the computer works with no software change needed. Software can fail without hardware failing, and hardware can fail without software failing. A hardware update need not change the software, and a software update need not change the hardware. Equally, each level has its own knowledge base: if software fails we call a programmer, but if hardware fails we call an engineer.

With this approach, design at one level is subject to higher level requirements. So software *requirements* can be met by hardware *operations*. For example, reading the information of a file takes longer if it is fragmented, as the drive head must jump between physically distant disk sectors. Defragmenting a disk improves software access by putting file data in adjacent physical sectors.

11. If the software gets in an infinite loop, we say it 'hangs'.

Information access improves even though the physical drive read rate has not changed. Hardware actions can meet software goals, as database and network requirements are met by new hardware chip commands. The software goal, of better information throughput, also becomes the hardware goal, so physical chip design today is as much about caching and co-processing as it is about cycle rate. Hardware design is increasingly about meeting software requirements.

As software emerged from hardware, it generated the new academic discipline of computer science. Computer scientists are not engineers, because they design algorithms not objects, but neither are they mathematicians, because their code runs on physical machines. Computer science is the "child" of mathematics and engineering, with the features of both but combined into a new discipline.

## 1.5 FROM SOFTWARE TO USERS

HCI began in the personal computing or PC era. Adding people to the computing equation meant that getting the technology to work was only half the problem — the other half was getting people to use it. Web users who did not like a site just clicked on, and only web sites that got hits succeeded. Given equal functionality, users prefer a more usable product (Davis, 1989); e.g. Word replaced Word Perfect because users took a week to learn Word Perfect but picked up Word in a day. Just as computing had previously gained a software level, so it now gained a human level.

*Human computer interaction* (*HCI*) is a person using IT, just as IT is software using hardware. As computer science emerged from a combination of mathematics and engineering, so HCI is emerging from psychology and computer science.

If psychology is the study of people and IT the study of software and hardware, then HCI is the study of psychology as it applies to IT. HCI is the discipline child of IT and psychology. It links CS to psychology as CS linked engineering to mathematics. This new meta-discipline applies psychological principles to computing design; e.g. Miller's paper on cognitive span suggests limiting computer menu choices to seven (Miller, 1956). The nature of people now defines the nature of computing; e.g. our many senses led to the multi-media trend.

## 1.6 FROM USERS TO COMMUNITIES

Even as HCI develops into a traditional academic discipline, computing has already moved on to add sociology to its list of paramours. Socio-technical systems use the social sciences in their design as HCI interfaces use psychology. STS is not part of HCI, nor is sociology part of psychology, because a society is more than the people in it; e.g. East and West Germany, with similar people, performed differently as communities, as do North and South Korea today. A society is not just a set of people. People who gather to view an event or customers shopping for bargains are an aggregate, not a community. They only become a community if they see themselves as one, i.e. the community level arises directly from personal level cognitions.

Social systems can have a physical base or a technical base, so a *socio-physical system* is people socializing by physical means. Face-to-face friendships cross seamlessly to Facebook because the social level persists across physical and electronic architecture bases. Whether electronically or physically mediated, a social system is *always* people interacting with people. Electronic communication may be "virtual" but the people involved are real.

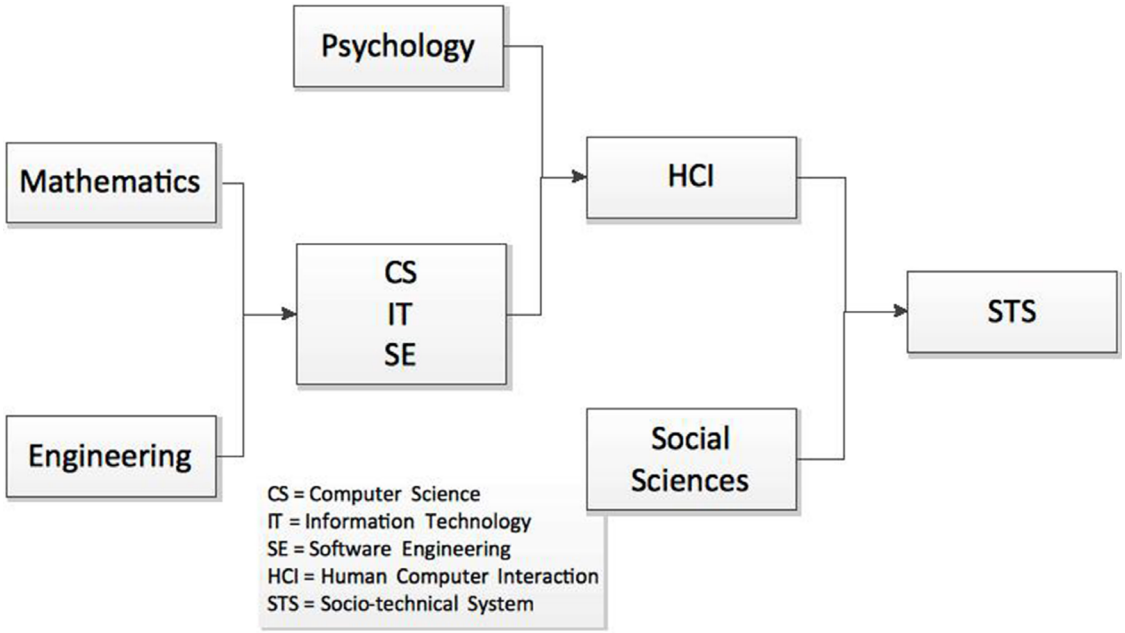



FIGURE 1.6: How computing disciplines arise.

*Copyright status: Unknown (pending investigation). See section "Exceptions" in the copyright terms below.*

Online communities work through people, who work through software that works through hardware. While sociology studies the social level alone, socio-technical design studies how social, human, information and hardware levels interact. A sociologist can no more design socio-technologies than a psychologist can design human-computer interfaces. STS and HCI need computer-sociologists and computer-psychologists but most universities don't train across disciplines. Yet both the complexity and the success of modern computing arises from its discipline promiscuity (Figure 1.6).

## 1.7 THE REDUCTIONIST DREAM

Before going on, we review the opposing theory of reductionism, which states that there is only one level, the physical level, and everything can reduce to it. Readers of a less philosphical bent may want to skip this section.

The reductionist dream is based on logical positivism[12], the idea that only the physical exists so all science must be expressed in physical terms. Yet when Shannon and Weaver defined information as a choice between physical options, the options were physical but the choosing was not (Shannon & Weaver, 1949). A message physically fixed in *one* way has by this definition *zero* information because the other ways it *could* have been fixed do not exist physically[13]. It is logically true that hieroglyphics that cannot be read contain in themselves no information at all.

If reader choices generate information, the data in a physical signal is unknown until it is deciphered. Data compression fits the same data in a physically smaller signal by encoding it more efficiently. It could not do this if information was fully defined by the physical message. The physical level is *necessary* for the information level but it is not *sufficient*. Conversely, information does not exist physically, as it cannot be touched or seen.

So if the encoding is unknown, the information is undefined; e.g. an electronic pulse sent down a wire could be a bit, or a byte (an ASCII "1"), or, as the first word of a dictionary, say "aardvark", be many bytes. The information a message conveys depends on the decoding process; e.g. every 10th letter of this text gives an entirely new (and nonsensical) message.

12. Logical positivism is a nineteenth century meta-physical position stating that all science involves only physical observables. In psychology, it led to Behaviorism (Skinner, 1948) which is now largely discredited (Chomsky, 2006). Science is not a way to prove facts, but a way to use world feedback to make best guesses. See http://researchroadmap.org/content/ for more details.

13. An on/off voltage choice is one bit, but a physical signal alone is no information.

One response to reductionism is *mathematical realism,* that mathematical laws are real even if they are not concrete (Penrose, 2005). Mathematics is a science because its constructs are logically correct, not because they are physical. That an equation is later physically useful is not the cause of its reality. Reality is now a consensual construct, with physicality just one option. Likewise in psychology Skinner's attempt to reduce all cognitions to physical behaviour did not work and has been replaced by *cognitive realism,* that cognitions are also real.

The acceptance of mathematical and cognitive constructs does not deny science, because science only requires that theory constructs be *validated* empirically, i.e. by a physical measure[14], not that they *be* physical. For example, fear as a cognitive construct can be measured by heart rate, pupil dilation, blood pressure, a questionnaire, etc.

Even physics cannot reduce its theories to pure physicality, as quantum theory implies a primordial non-physical[15] *quantum level* of reality below the physical (Whitworth, 2011). Physical reductionism gave a clockwork universe where each state perfectly defined the next as in a computer, but quantum physics flatly denied this, as random quantum events by definition have no preceding physical cause. The quantum world cannot be reduced to physical events[16]. Either quantum theory is wrong, or reductionism does not work. If all science were physical, all science would be physics, which it is not.

A reductionist philosophy that has failed in science in general is hardly a good base for a computing model. If the physical level were sufficient alone, there would be no choices and so no information, i.e. *reductionism denies information science*. As the great 18th century German philosopher Kant argued

14. Empirical means derived from the physical world. Mental constructs with no physical referent, like love, are outside it.

15. For example, quantum collapse ignores the speed of light limit and quantum waves travel many paths at once.

16. Rather the reverse, with a non-physical quantum reality below the physical.

long ago, we see an object, or phenomenon, as a view, but don't see *the thing in itself* (Kant, 1781/2002)[17]. Following Kant's model, the different disciplines of science are just different ways to view the same unspecified reality. Levels return the observer to science, as quantum theory's paradoxes demand.

Currently, sociology sees individuals as conduits of meaning that reflect external social structures, and so psychological, biological, and physical views are the faulty reductionism of social realities. In this *social determinism*, society writes social agendas, such as communism or capitalism, upon individual *tabulae rasae* (blank slates). Yet this just replaces the determinism of fields like biology (Wilson, 1975) and psychology (Skinner, 1948) by another form of determinism.

By contrast, in the general system model of computing shown in Figure 1.5, each level emerges from the previous. So if all individual thoughts were erased, society would also cease to exist as surely as if all its citizens had vanished physically. Sociology assumes psychology, which has led to attempts to re-attach it to its psychological roots, e.g. Bourdieu's *habitus* references individual cognitions of the social environment and Gidden's mental frames underlie social life (Bone, 2005). The top-down return of sociology to its source matches an equally vibrant bottom-up movement in computing, which has long seen itself as more than hardware and software (Boulding, 1956).

## 1.8 THE REQUIREMENTS HIERARCHY

The evolution of computing implies a requirements hierarchy (Figure 1.7). If the hardware works, then software becomes the priority; if the software works, then user needs become important; and if user needs are fulfilled, then social requirements arise. As one level's issues are met, those of the next appear, just as climbing one hill reveals another. As hardware over-heating problems are solved, software

17. He called the "thing in itself" the *noumenon*, as opposed to the *phenomenon*, or view we see. A bat or a bee see the world differently from us. It is egocentrism to assume the world is only as we see it.

data locking problems arise. As software response times improve, user response times become the issue. Companies like Google and E-bay still seek customer satisfaction, but customers in crowds also have community needs like fairness, i.e. higher system levels invoke higher requirements.

In general, *the highest system level defines its success*; e.g. social networks need a community to succeed. If no community forms, it does not matter how easy to use, fast or reliable the software is. Lower levels become necessary to avoid failure but not sufficient to define success.

| Level | Requirements | Errors |
|---|---|---|
| *Community* | Reduce community overload, clashes. Increase productivity, synergy, fairness, freedom, privacy, transparency. | Unfairness, slavery, selfishness, apathy, corruption, lack of privacy, loss of synergy. |
| *Personal* | Reduce cognitive overload, clashes. Increase meaning transfer efficiency. | User misunderstands, gives up, is distracted, or enters wrong data. |
| *Information* | Reduce information overload, clashes. Increase data processing, storage, or transfer efficiency | Processing hangs, data storage full, network overload, data conflicts. |
| *Mechanical* | Reduce physical heat or force overload. Increase heat or force efficiency. | Overheating, mechanical fractures or breaks, heat leakage, jams |

TABLE 1.2: Computing errors by system level.

Conversely, *any level can cause failure;* it does not matter how strong the community is if the hardware fails, the software crashes or the interface is unusable. An STS fails if its hardware fails, if its program crashes or if users cannot figure it out. Hardware, software, personal and community failures are all computing errors (Table 1.2). The common feature is that *the system fails to perform* and in evolution what does not perform, does not survive[18].
Each level emerges from the previous but fails differently:

1. *Hardware systems* based on physicalenergy exchange fail from problems like overheating.
2. *Software systems* based on information exchange fail from problems like infinite loops.
3. *HCI systems* based on meaning exchange fail from problems like misunderstanding or information overload.
4. *Socio-technical systems* based on normative meme exchange fail from problems like mistrust, unfairness and injustice.

Computing as technology fails for technical reasons but, as socio-technology, also fails for social reasons.

18. For example, traffic congestion, a problem in large cities, can be addressed at the:

    **1.** *Physical level*, e.g. by building more roads;
    **2.** *Informational level*, e.g. by better road rules or signs;
    **3.** *Human level*, e.g. by driver training or phrases like *merge like a zip* at high way on-ramps;
    **4.** *Community level*, e.g. by changing norm that everyone must work from 9am to 5pm, to reduce rush hour overloads.

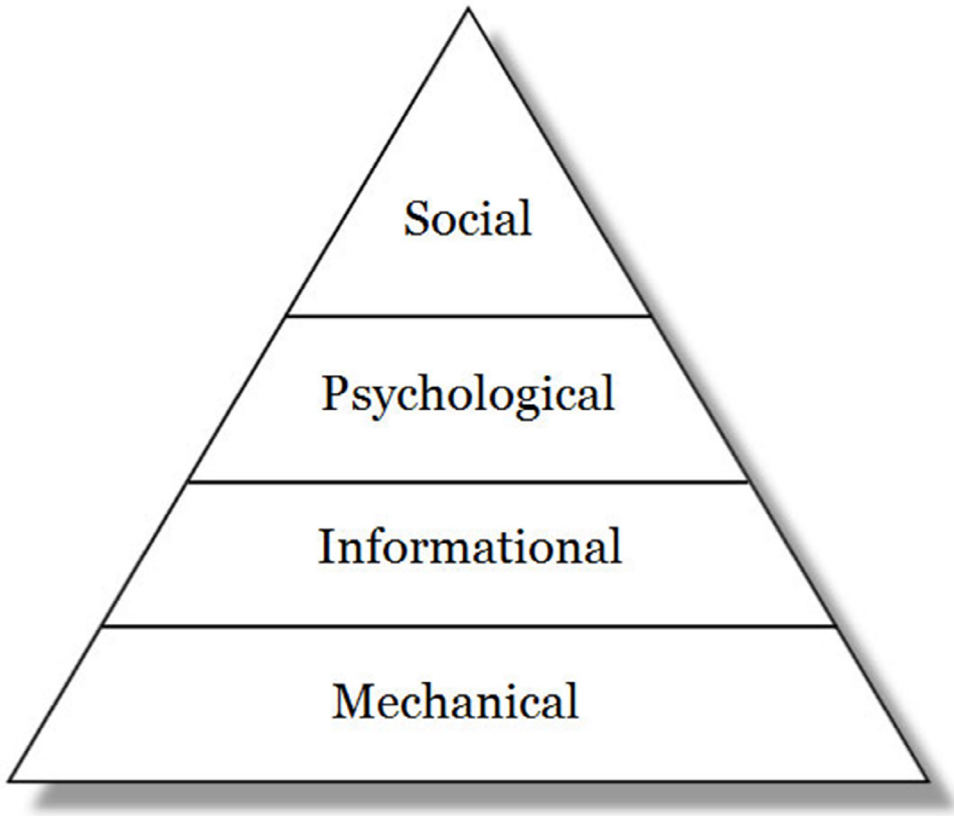



FIGURE 1.7: The computing requirements hierarchy.

*Courtesy of Brian Whitworth and Adnan Ahmad. Copyright: CC-Att-SA-3 (Creative Commons Attribution-ShareAlike 3.0).*

Technology is hard, but society is soft. That the soft should direct the hard seems counter-intuitive, but trees grow at their soft tips more than at their hard base. As a tree trunk does not direct its expanding canopy, so today's social computing advances were undreamt of by its engineering base. Today's technology designers will find the future of design in level combinations.

## 1.9 DESIGN LEVEL COMBINATIONS

The general system concept of levels makes system design complex, as the design requirements of one level can "flow down" to those below it. This gives us a variety of design fields, as seen below.


FIGURE 1.8.A: Apple controls meet human requirements.

*Courtesy of Ocrho. Copyright: pd (Public Domain (information that is common property and contains no original authorship)).*


FIGURE 1.8.B: TV controls meet engineering requirements.

*Courtesy of Unknown author. Copyright: pd (Public Domain (information that is common property and contains no original authorship))*[19].

1. *Ergonomics* designs safe and comfortable machines for people. Applying biological needs, such as avoiding posture and eye-strain, to technology design merges biology and engineering.
2. *Object design* applies psychological needs to technology in the same way (Norman, 1990): e.g. a door's design affects whether it is pushed or pulled. An *affordance* is a physical object feature that cues its human use, as but-

19. Courtesy of Ocrho.

tons cue pressing. Physical systems designed with affordances based on human requirements perform better. In World War II, aircraft crashed until engineers designed cockpit controls with the cognitive needs of pilots in mind, as follows (with computing examples):

a. Put the control by the thing controlled, e.g. a handle on a door (context menus).
b. Let the control "cue" the required action, e.g. a joystick (a 3D screen button).
c. Make the action/result link intuitive, e.g. press a joystick forward to go down, (press a button down to turn on).
d. Provide continuous feedback, e.g. an altimeter, (a web site breadcrumbs line).
e. Reduce mode channels, e.g. altimeter readings, (avoid edit and zoom mode confusions).
f. Use alternate sensory channels, e.g. warning sounds, (error beeps).
g. Let pilots "play", e.g. flight simulators, (a system sandbox).

8. *Human computer interaction* applies psychological requirements to screen design. Usable interfaces respect cognitive principles, e.g. by the nature of human attention, users do not usually read the entire screen. HCI turns psychological needs into IT designs as architecture turns buyer needs into house designs. Compare Steve Jobs' IPod to a television remote (Figure 1.8). Both are controls[20] but one is a cool tool and the other a mass of buttons. If one was designed to engineering requirements and the other to HCI requirements, which performs better?
9. *Fashion* is the social requirement to look good applied to wearable object design. In computing, a mobile phone can be a fashion accessory, just like a hat or handbag. Its role is to impress, not just to function. Aesthetic criteria

20. In fact the IPod does more

apply when people buy mobile phones to be trendy or fashionable, so colour can be as important as battery life in mobile phone design.

10. *Socio-technology* is information technology meeting social requirements. Anyone online can see its power, but most academics see it as an aspect of their specialty, rather than a new multi-discipline in its own right. As computing evolved a social level, social requirements became part of computing design (Sanders & McCormick, 1993).

Multi-disciplinary fields cannot, by their nature, be reduced to component discipline specialties; e.g. sociologists study society not technology, and technologists study technology not society, so neither can address socio-technical design — how social needs impact technical design. Table 1.3 summarizes design fields by level combination.

| Design | Requirements | Target | Examples |
|---|---|---|---|
| *STS* | Social | IT | Wikipedia, YouTube, E-bay |
| *Fashion* | Social | Physical Accessory | Mobile phone as an accessory |
| *HCI* | Psychological | IT | Framing, border contrast, richness |
| *Design* | Psychological | Technology | Keyboard, mouse |
| *Ergonomics* | Biological | Technology | Adjustable height screen |

TABLE 1.3: Design fields by target and requirement levels.

In Figure 1.9, higher level requirements flow down to lower level design, giving a *higher affects lower* design principle. Higher levels direct lower ones to improve system performance. Levels cumulate, so the requirements of each level flow down to those below, e.g. community agreement gives normative influence at the citizen level, laws at the informational level, and cultural events at the physical level. The same applies online, as online communities make demands of Netizens[21] as well as software. STS design therefore is about having it all: reliable devices, efficient code, intuitive interfaces and sustainable communities.

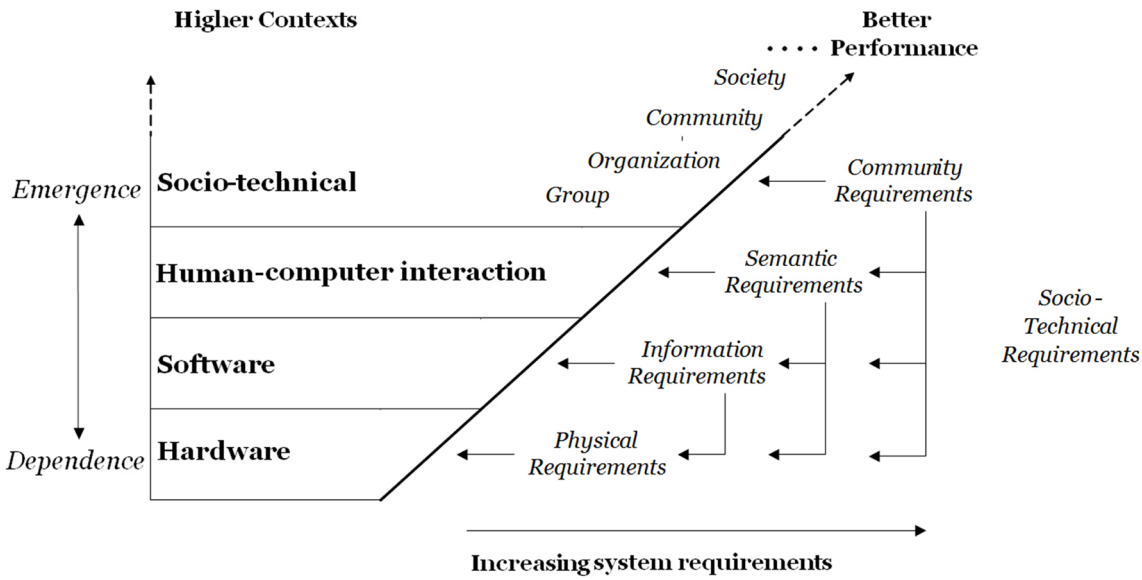



FIGURE 1.9: Computing requirements cumulate.

*Copyright status: Unknown (pending investigation). See section "Exceptions" in the copyright terms below.*

21. For 'netiquette' see: http://www.kent.edu/dl/technology/etiquette.cfm

Note that the social level is open ended, as social groups form higher social groups, e.g. in physical society, over thousands of years, families formed tribes, tribes formed city states, city-states formed nations and nations formed nations of nations, each with more complex social structures (Diamond, 1998). How social units combine into higher social units[22] with new requirements is discussed further in Chapter 5.

It is naive to think that friend systems like Facebook are the last step, that social computing will stop at a social unit size of two. Beyond friends are tribes, cities, city-states, nations and meta-nations like the USA. Since we *have* a friend but *belong* to a community, the rules also change. With the world population at seven billion and growing, Facebook's over 900 million active accounts are just the beginning. The future is computer support not just for friends, but also for families, tribes, nations and even global humanity.

For example, imagine a *group browser,* designed for many not just one, so that people can browse the Internet in groups, discussing as they go. Instead of a physical tour bus there is an informational tour browser. It can have a "driver" who comments along the way: "*This site shows how the Internet began …*" . Or members could take turns to host the next site, showing what they like. The possibilities of social computing are just beginning.

22. The social unit of analysis can be a person, a friend dyad, a group, a tribe, etc.

## 1.10 THE FLOWER OF COMPUTING

Figure 1.10 shows how computing evolved through the four stages of hardware, software, people and community. At each stage, a new specialty joined computing, but pure engineers still see only mechanics, pure computer scientists only information, pure psychologists only human constructs, and pure sociologists only social structures. Yet the multi-discipline of computing as a whole is not pure, because purity is not the future. It is more akin to a bazaar than a cathedral, as computer practitioners understand (Raymond, 1999).

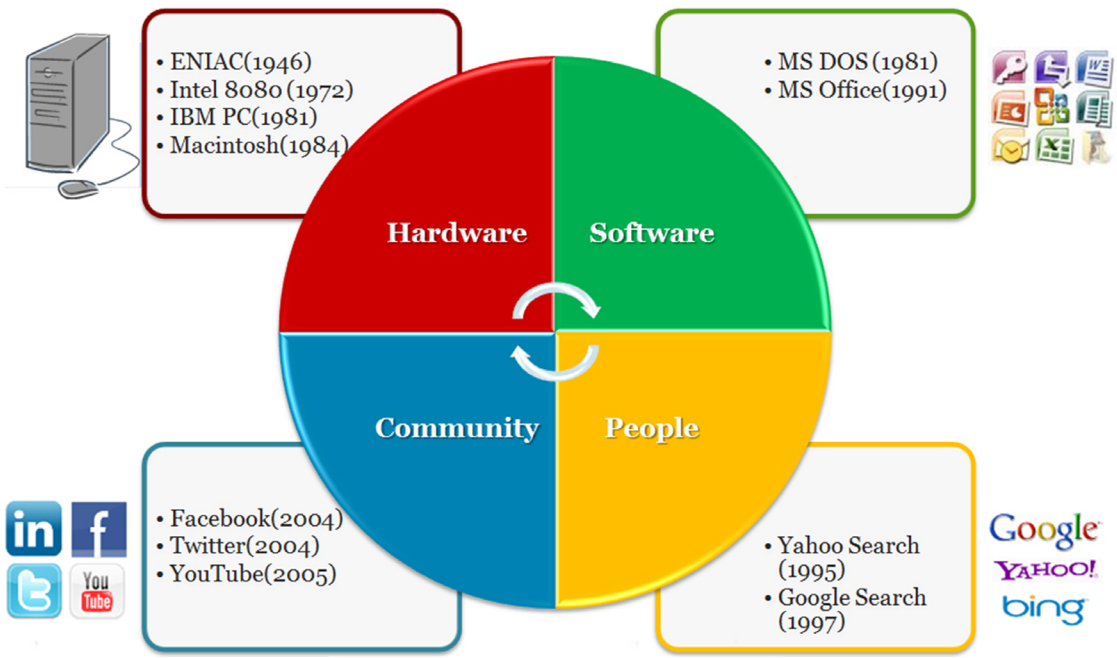



FIGURE 1.10: The four stages of computing.

*Courtesy of Brian Whitworth and Adnan Ahmad. Copyright: CC-Att-SA-3 (Creative Commons Attribution-ShareAlike 3.0)*[23].

23. Design by Yijing Qian

In academia, computing struggles because academics must specialize to get publications, grants and promotions (Whitworth & Friedman, 2009), so discipline specialties guard their knowledge in journal castles with jargon walls. Like medieval fiefdoms, they hold hostage knowledge that by its nature should be free. The divide and conquer approach of reductionism does not allow computing to prosper as an academic multi-discipline.

In practice, however, computing is thriving. Every day more people use computers to do more things in more ways, so engineering, computer science, health[24], business, psychology, mathematics and education compete for the computing crown[25]. The kingdom of computingresearch is weak because it is a realm divided. It will get weaker if music, art, journalism, architecture etc. also add outposts. Computing researchers are scattered over the academic landscape like the tribes of Israel, some in engineering, some in computer science, some in health, etc. *Yet we are one.*

The flower of computing is the fruit of many disciplines but it belongs to none. It is a new multi-discipline in itself (Figure 1.11). For it to bear research fruit, its discipline parents must release it. Using different terms, models and theories for the same subject just invites confusion. Universities that compartmentalize computing research into isolated discipline groups deny its multi-disciplinary future. As modern societies federate states and nations, so the future of computing is as a federation of disciplines. Until computing research unifies, it will remain as it is now — a decade behind computing practice.

24. Health created its own field of **informatics**, with separate journals, conferences and courses, to meet its non-engineering/non-business computing needs

25. Computing is the Afghanistan of academia, often invaded but never conquered. It should be the Singapore, a knowledge trade centre.

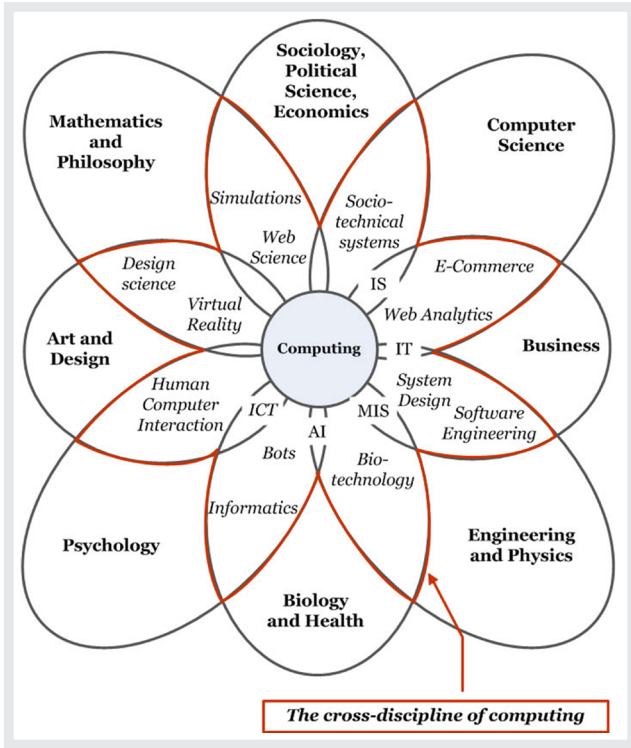



FIGURE 1.11: The flower of computing.

*Courtesy of Brian Whitworth and Adnan Ahmad. Copyright: CC-Att-SA-3 (Creative Commons Attribution-ShareAlike 3.0).*

## 1.11 DISCUSSION QUESTIONS

The following questions are designed to encourage thinking on the chapter and exploring socio-technical cases from the Internet. If you are reading this chapter in a class - either at university or commercial – the questions might be discussed in class first, and then students can choose questions to research in pairs and report back to the next class.

1. How has computing evolved since it began? Is it just faster machines and better software? What is the role of hardware companies like IBM and Intel in modern computing?

2. How has the computing business model changed as it evolved? Does selling software make more money than selling hardware? Can selling knowledge make money? What about selling friendships? Can one sell communities?
3. Is a kitchen table a technology? Is a law a technology? Is an equation a technology? Is a computer program a technology? Is an information technology (IT) system a technology? Is a person an information technology? Is an HCI system (person plus computer) an information technology? What, exactly, *is not* a technology?
4. Is any set of people a community? How do people form a community? Is a socio-technical system (an online community) any set of HCI systems? How do HCI systems form an online community?
5. How does computer science relate to engineering and mathematics? What about human computer interaction (HCI) and engineering, computer science and psychology? Or socio-technology and engineering, computer science, psychology and the various social sciences?[26]
6. In an aircraft, is the pilot a person, a processor, or a physical object? Can one consistently divide the aircraft system into human, computer and mechanical parts? If not, what is the alternative?
7. What is the reductionist dream? How did it work out in physics? Does it recognize computer science? How did it challenge psychology? Has it worked out in any discipline?
8. How much information does a physical book, that is fixed in *one* way, by definition, have? If we say a book "contains" information, what is assumed? How is a book's information generated? Can *the same* physical book "contain" different information for different people? Give an example.
9. If information is physical, how can data compression put the same information in a physically smaller signal? If information is not physical, how does data compression work? Can we encode more than one semantic stream into one physical message? Give an example.

26. Like, sociology, history, political science, anthropology, ancient history, etc.

10. Is a bit a physical "thing"? Can you see or touch a bit? If a signal wire sends a physical "on" value, is that always a bit? If a bit is not physical, can it exist without physicality? How can a bit *require* physicality but not itself *be* physical? What creates information, if it is not the mechanical signal?
11. Is information concrete? If we cannot see information physically, is the study of information a science? Explain. Are cognitions concrete? If we cannot see cognitions physically, is the study of cognitions (psychology) a science? Explain. What separates science from imagination if it can use non-physical constructs in its theories?
12. Give three examples of other animal species who sense the world differently from us. If we saw the world as they do, how would it change what we *do?* Explain how *seeing* a system differently can change how it is *designed*. Give examples from computing.
13. If a $1 CD with a $1,000 software application on it is insured, what do you get if it is destroyed? Can you insure something that is not physical? Give current examples.
14. Is a "mouse error" a hardware, software or HCI problem? Can a mouse's hardware affect its software performance? Can it affect its HCI performance? Can mouse software affect HCI performance? Give examples in each case. If a wireless mouse costs more and is less reliable, how is it better?
15. Give three examples of a human requirement giving an IT design heuristic. This is HCI. Give three examples of a community requirement giving an IT design heuristic. This is STS.
16. Explain the difference between a hardware error, a software error, a user error and a community error, with examples. What is the common factor here?
17. What is an application user sandbox? What human requirement does it satisfy? Illustrate with an online example of a user sandbox.
18. Distinguish between a personal requirement and community requirement in computing. Relate to how STS and HCI differ and how socio-technology

and sociology differ. Are sociologists qualified to design socio-technical systems? What about HCI experts?

19. In general, what do people do when their needs are not met in a physical situation? Relate to what users do if their needs are not met online. Is there a difference? Explain. What do citizens of a physical community do if it does not meet their needs? What about an online community? Again, is there a difference? Give specific examples to illustrate.
20. According to Norman, what is ergonomics? What is the difference between ergonomics and HCI? What is the difference between HCI and STS?
21. Give examples of the following: Hardware meeting engineering requirements. Hardware meeting Computer Science requirements. Software meeting CS requirements. Hardware meeting psychology requirements. Software meeting psychology requirements. People meeting psychology requirements. Hardware meeting community requirements. Software meeting community requirements. People meeting community requirements. Communities meeting their own requirements. Which of these are computing design?
22. Why is an IPod so different from TV or video controls? Which is better and why? Why has TV remote design changed so little in decades? If scheduled television competes with Internet videos for the hearts and minds of viewers, which one will win? Give advantages and disadvantages of both sides.
23. How does an online friend differ from a physical friend? Can friendships transcend physical and electronic interaction architectures? Give examples. How is this possible?
24. How available are academic papers? Pick 10 important journal papers and using non-university browsing, try to access them for free. How many author home pages offer their own papers for free download? Should journals be able to copyright papers they neither wrote nor paid for?
25. Why do universities divide computing research across many disciplines? What is a cross-discipline? What past cross-disciplines became disciplines. Why is computing a cross-discipline?

## Your notes and thoughts on chapter 1

Record your notes and thoughts on this chapter. If you want to share these thoughts with others online, go to the bottom of the page at: http://www.interaction-design.org/books/the_social_design_of_technical_systems_2nd_ed/the_evolution_of_computing.html

**NOTES:**

# CHAPTER 2

# Design Spaces

*"Customer: 'Which of your cuts are the best?'*
*Butcher: 'All of my cuts are the best.'"*

While the previous chapter described computing system levels, this chapter describes how the many dimensions of system performance interact to create a design requirements space.

## 2.1 THE ELEPHANT IN THE ROOM

The beast of computing has regularly defied pundit predictions. Key advances like the cell-phone (Smith et al., 2002) and open-source development (Campbell-Kelly, 2008) were not predicted by the experts of the day, though the signs were there for all to see. Experts were pushing media-rich systems even as lean text chat, blogs, texting and wikis took off. Even today, people with smart-phones still

send text messages. Google's simple white screen scooped the search engine field, not Yahoo's multi-media graphics. The gaming innovation was social gaming, not virtual reality helmets as the experts predicted. Investors in Internet bandwidth lost money when users did not convert to a 3G video future. Cell phone companies are still trying to get users to sign up to 4G networks.

In computing, the idea that practice leads but theory bleeds has a long history. Over thirty years ago, paper was declared "dead", to be replaced by the electronic paperless office (Toffler, 1980). Yet today, paper is used more than ever before. James Martin saw program generators replacing programmers, but today, we still have a programmer shortage. A "leisure society" was supposed to arise as machines took over our work, but today we are less leisured than ever before (Golden & Figart, 2000). The list goes on: email was supposed to be for routine tasks only, the Internet was supposed to collapse without central control, video was supposed to replace text, teleconferencing was supposed to replace air travel, AI smart-help was supposed to replace help-desks, and so on.

We get it wrong time and again, because computing is the elephant in our living room. We cannot see it because it is too big. In the story of the blind men and the elephant, one grabbed its tail and found the elephant like a rope and bendy, another took a leg and declared the elephant was fixed like a pillar, a third felt an ear and thought it like a rug and floppy, while the last seized the trunk, and found it like a pipe but very strong (Sanai, 1968). Each saw a part but none saw the whole. This chapter outlines the multi-dimensional nature of computing.

## 2.2 DESIGN REQUIREMENTS

To design a system is to find problems early, e.g. a misplaced wall on an architect's plan can be moved by the stroke of a pen, but once the wall is built, changing it is not so easy. Yet to design a thing, its performance requirements must be known. Doing this is the job of *requirements engineering,* which analyzes stakeholder

needs to specify what a system must do in order that the stakeholders will sign off on the final product. It is basic to system design:

> "*The primary measure of success of a software system is the degree to which it meets the purpose for which it was intended. Broadly speaking, software systems requirements engineering (RE) is the process of discovering that purpose...*"
>
> *-- Nuseibeh & Easterbrook, 2000: p. 1*

A requirement can be a particular value (e.g. uses SSL), a range of values (e.g. less than $100), or a criterion scale (e.g. is secure). Given a system's requirements designers can build it, but the computing literature cannot agree on what the requirements are. One text has usability, maintainability, security and reliability (Sommerville, 2004, p. 24) but the ISO 9126-1 quality model has functionality, usability, reliability, efficiency, maintainability and portability (Losavio et al., 2004).

Berners-Lee made scalability a World Wide Web criterion (Berners-Lee, 2000) but others advocate open standards between systems (Gargaro et al., 1993). Business prefers cost, quality, reliability, responsiveness and conformance to standards (Alter, 1999), but software architects like portability, modifiability and extendibility (de Simone & Kazman, 1995). Others espouse flexibility (Knoll & Jarvenpaa, 1994) or privacy (Regan, 1995). On the issue of what computer systems need to succeed, the literature is at best confused, giving what developers call *the requirements mess* (Lindquist, 2005). It has ruined many a software project. It is the problem that agile methods address in practice[1] and that this chapter now addresses in theory.

In current theories, each specialty sees only itself. Security specialists see *security* as availability, confidentiality and integrity (OECD, 1996), so to them, reliability is part of security. Reliability specialists see *dependability* as reliability,

1. See the agile page at the open directory project: http://www.dmoz.org/Computers/Programming/Methodologies/Agile/

safety, security and availability (Laprie & Costes, 1982), so to them security is part of a general reliability concept. Yet both cannot be true. Similarly, a usability review finds functionality and error tolerance part of usability (Gediga et al., 1999) while a flexibility review finds scalability, robustness and connectivity aspects of flexibility (Knoll & Jarvenpaa, 1994). Academic specialties usually expand to fill the available theory space, but some specialties recognize their limits:

> "*The face of security is changing. In the past, systems were often grouped into two broad categories: those that placed security above all other requirements, and those for which security was not a significant concern. But ... pressures ... have forced even the builders of the most security-critical systems to consider security as only one of the many goals that they must achieve*"
>
> *-- Kienzle & Wulf, 1998: p5*

Analyzing performance goals in isolation gives diminishing returns.

## 2.3 DESIGN SPACES

Architect Christopher Alexander observed that vacuum cleaners with powerful engines and more suction were also heavier, noisier and cost more (Alexander, 1964). One performance criterion has a best point, but two criteria, like power and cost, give a best line. The efficient frontier of two performance criteria is a line, of the maxima of one for each value of the other (Keeney & Raiffa, 1976). System design is choosing a point in a multi-dimensional design criterion space with many "best" points, e.g. a cheap, heavy but powerful vacuum cleaner, or a light, expensive and powerful one (Figure 2.1). The efficient frontier is a set of "best" combinations in a design space[2]. Advanced system performance is not a one dimensional ladder to excellence, but a station with many trains to many destinations.

2. Not all criterion combinations may be achievable, e.g. a light, cheap and powerful vacuum cleaner

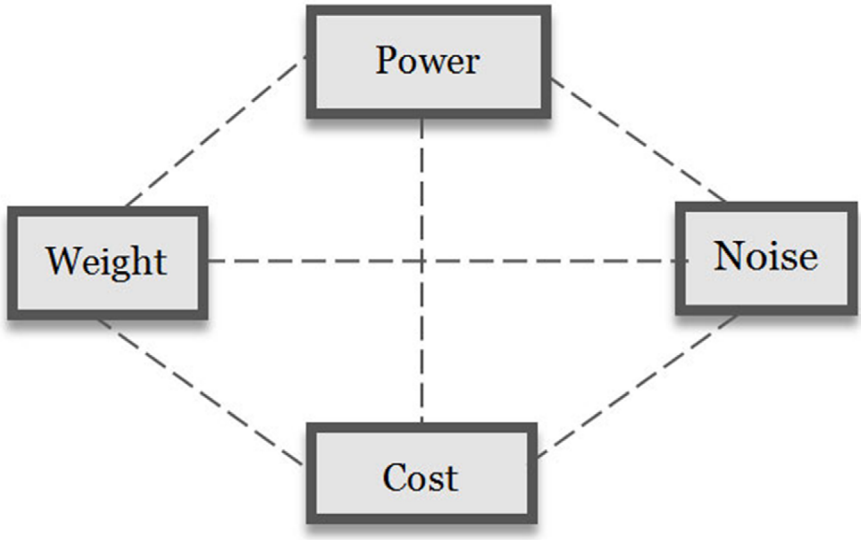



Figure 2.1: A vacuum cleaner design space.

*Courtesy of Brian Whitworth and Adnan Ahmad. Copyright: CC-Att-SA-3 (Creative Commons Attribution-ShareAlike 3.0).*

Successful life includes flexible viruses, reliable plants, social insects and powerful tigers, with the latter the endangered species. There is no "best" animal because in nature performance is multi-dimensional, and a multi-dimensional design space gives many best points. In evolution, not just the strong are fit, and too much of a good thing can be fatal, as over specialization can lead to extinction.

Likewise, computing has no "best". If computer performance was just more processing we would all want supercomputers, but we buy laptops with less power over desktops with more (David et al., 2003). On the information level, blindly adding software functions gives bloatware[3], applications with many features no-one needs.

Design is the art of reconciling many requirements in a particular system form, e.g. a quiet and powerful vacuum cleaner. It is the innovative synthesis of a system in a design requirements space (Alexander, 1964). The system then *performs* according to its requirement criteria.

3. Also called featuritis or scope creep

Most design spaces are not one dimensional, e.g. Berners-Lee chose HTML for the World Wide Web for its flexibility (across platforms), reliability *and* usability (easy to learn). An academic conference rejected his WWW proposal because HTML was inferior to SGML (Standard Generalized Markup Language). Specialists saw their specialty criterion, not system performance in general. Even after the World Wide Web's phenomenal success, the blindness of specialists to a general system view remained:

"*Despite the Web's rise, the SGML community was still criticising HTML as an inferior subset ... of SGML*"

*-- Berners-Lee, 2000: p96*

What has changed since academia found the World Wide Web inferior? Not a lot. If it is any consolation, an equally myopic Microsoft also found Berners-Lee's proposal unprofitable from a business perspective. In the light of the benefits both now freely take from the web, academia and business should re-evaluate their criteria.

## 2.4 NON-FUNCTIONAL REQUIREMENTS

In traditional engineering, criteria like usability are *quality* requirements that affect functional goals but cannot stand alone (Chung et al., 1999). For decades, these non-functional requirements (NFRs), or "-ilities", were considered second class requirements. They defied categorization, except to be non-functional. How exactly they differed from functional goals was never made clear (Rosa et al., 2001), yet most modern systems have more lines of interface, error and network code than functional code, and increasingly fail for "unexpected" non-functional reasons[4] (Cysneiros & Leite, 2002, p. 699).

4. Hardly surprising if we define NFRs to be less important.

The logic is that because NFRs such as reliability cannot exist without functionality, they are subordinate to it. Yet by the same logic, functionality is subordinate to reliability as it cannot exist without it, e.g. an unreliable car that will not start has no speed function, nor does a car that is stolen (low security), nor one that is too hard to drive (low usability).

NFRs not only modify performance but define it. In nature, functionality is not the only key to success, e.g. viruses hijack other systems' functionality. Functionality differs from the other system requirements only in being more obvious *to us*. It is really just one of many requirements. The distinction between functional and non-functional requirements is a bias, like seeing the sun going round the earth because we live on the earth.

## 2.5 HOLISM AND SPECIALIZATION

In general systems theory, any system consists of:

a. Parts, and
b. Interactions.

The performance of a system of parts that interact is not defined by decomposition alone. Even simple parts, like air molecules, can interact strongly to form a chaotic system like the weather (Lorenz, 1963). Gestalt psychologists called the concept of the whole being more than its parts holism, as a curve is just a curve but in a face becomes a "smile". Holism is how system parts change by interacting with others. Holistic systems are individualistic, because changing one part, by its interactions, can cascade to change the whole system drastically. People rarely look the same because one gene change can change everything. The brain is also holistic — one thought can change everything you know.

Yet a system's parts need not be simple. The body began as one cell, a zygote, that divided into all the cells of the body, including liver, skin, bone and

brain cells[5]. Likewise in early societies most people did most things, but today we have millions of specialist jobs. A system's specialization[6] is the degree to which its parts differ in form and action, especially its constituent parts.

Holism (complex interactions) and specialization (complex parts) are the hallmarks of evolved systems. The former gives the levels of the last chapter and the latter gives the constituent part specializations discussed now.

## 2.6 CONSTITUENT PARTS

In general terms, what are the parts of a system? Are software parts lines of code, variables or sub-programs? A system's *elemental parts* are those not formed of other parts, e.g. a mechanic stripping a car stops at the bolt as an elemental part of that level. To decompose it further gives atoms which are physical not mechanical elements. Each level has a different elemental part: physics has quantum strings, information has bits, psychology has qualia[7] and society has citizens (Table 2.1). Elemental parts then form complex parts as bits form bytes.

| Level | Element | Other parts |
|---|---|---|
| *Community* | Citizen | Friendships, groups, organizations, societies. |
| *Personal* | Qualia | Cognitions, attitudes, beliefs, feelings, theories. |
| *Informational* | Bit | Bytes, records, files, commands, databases. |
| *Physical* | Strings? | Quarks, electrons, nucleons, atoms, molecules. |

TABLE 2.1: System elements by level.

5. Deciphering the human genome gave the pieces of the genetic puzzle, not how they connect
6. Specialization is also called differentiation
7. A qualia is a basic subjective experience, e.g. the pain of a headache

A system's *constituent parts* are those that interact to form the system but are *not* part of other parts (Esfeld, 1998), e.g. the body frame of a car is a constituent part because it is part of the car but not part of another car part. So, dismantling a car entirely gives elemental parts, not constituent parts, e.g. a bolt on a wheel is not a constituent part if it is part of a wheel.

How elemental parts give constituent parts is the *system structure*, e.g. to say a body is composed of cells ignores its structure: how parts combine into higher parts or *sub-systems*. Only in system *heaps*, like a pile of sand, are elemental parts also constituent parts. An advanced system like the body is not a heap because the cell elements combine to form sub-systems just as the elemental parts of a car do, e.g. a wheel as a car constituent has many sub-parts. Just sticking together arbitrary constituents in design without regard to their interaction has been called the Frankenstein effect[8] (Tenner, 1997). The body's constituent parts are, for example, the digestive system, the respiratory system, etc., not the head, torso and limbs. The specialties of medicine often describe body constituents.

In order to develop a general model of system design it is necessary to specify the constituent parts of systems in general.

8. Dr. Frankenstein made a human being by putting together the best of each individual body part he could find in the graveyard. The result was a monster.

## 2.7 GENERAL SYSTEM REQUIREMENTS

Requirement engineering aims to define a system's purposes. If levels and constituent specializations change those purposes, how can requirements engineering succeed? The approach used here is to try to view *the system from the perspective of the system itself*. This means specifying requirements for level constituents without regard to any particular environment. How general requirements can be reconciled in a specific environment is then the art of system design.

In general, a system *performs* by interacting with its environment to gain value and avoid loss in order to survive. In Darwinian terms, what does not survive dies out and what does lives on. Any system in this situation needs a *boundary,* to exist apart from the world, and an internal *structure,* to support and manage its existence. It needs *effectors* to act upon the environment around it, and *receptors* to monitor the world for risks and opportunities. The requirement to reproduce is here ignored as it is not very relevant to computing. This would however add a time dimension to the model.

| Constituent | Requirement | Definition |
|---|---|---|
| **Boundary** | **Security** | *To deny unauthorized entry, misuse or takeover by other entities.* |
| | **Extendibility** | *To attach to or use outside elements as system extensions.* |
| **Structure** | **Flexibility** | *To adapt system operation to new environments* |
| | **Reliability** | *To continue operating despite system part failure* |
| **Effector** | **Functionality** | *To produce a desired change on the environment* |
| | **Usability** | *To minimize the resource costs of action* |
| **Receptor** | **Connectivity** | *To open and use communication channels* |
| | **Privacy** | *To limit the release of self-information by any channel* |

Table 2.2: General system requirement definitions.

For example, cells first evolved a boundary membrane, then organelle and nuclear structures for support and control; then eukaryotic cells evolved flagella to move, and then protozoa got photo-receptors (Alberts et al., 1994). We also have a skin boundary, metabolic and brain structures, muscle effectors and sense receptors. Computers also have a case boundary, a motherboard internal structure, printer or screen effectors and keyboard or mouse receptors.

Four general system constituents, each with risk and opportunity goals, gives eight general *performance requirements* (Table 2.2) as follows:

a. *Boundary* constituents manage the system boundary. They can be designed to deny outside things entry (security) or to use them (extendibility). In computing, virus protection is security and system add-ons are extendibility (Figure 2.2). In people, the immune system gives biological security and tool-use illustrates extendibility.
b. *Structure* constituents manage internal operations. They can be designed to limit internal change to reduce faults (reliability), or to allow internal change to adapt to outside changes (flexibility). In computing, reliability reduces and recovers from error, and flexibility is the system preferences that allow customization. In people, reliability is the body fixing a cell "error" that might cause cancer, while the brain learning illustrates flexibility.
c. *Effector* constituents manage environment actions, so can be designed to maximize effects (functionality) or minimize resource use (usability). In computing, functionality is the menu functions, while usability is how easy they are to use. In people, functionality gives muscle effectiveness, and usability is metabolic efficiency.
d. *Receptor* constituents manage signals to and from the environment, so can be designed to open communication channels (connectivity) or close them (privacy). Connected computing can download updates or chat online, while privacy is the power to disconnect or log off. In people, connectivity is conversing, and privacy is the legal right to be left alone. In nature, privacy is camouflage, and the military calls it stealth. Note that privacy is the ownership of self-data, not secrecy. It includes the right to make personal data public.

These general system requirements map well to current terms (Table 2.3), but note that as what is exchanged changes by level, so do the names preferred:

a. *Hardware systems exchange energy*. So functionality is power, e.g. hardware with a high CPU cycle rate. Usable hardware uses less power for the same result, e.g. mobile phones that last longer. Reliable hardware is rugged enough to work if you drop it, and flexible hardware is mobile enough to work if you move around, i.e. change environments. Secure hardware blocks physical theft, e.g. by laptop cable locks, and extendible hardware has ports for peripherals to be attached. Connected hardware has wired or wireless links and private hardware is tempest proof i.e. it does not physically leak energy.
b. *Software systems exchange information*. Functional software has many ways to process information, while usable software uses less CPU processing ("lite" apps). Reliable software avoids errors or recovers from them quickly. Flexible software is operating system platform independent. Secure software cannot be corrupted or overwritten. Extendible software can access OS program library calls. Connected software has protocol "handshakes" to open read/write channels. Private software can encrypt information so that others cannot see it.
c. *HCI systems exchange* meaning, including ideas, feelings and intents. In functional HCI the human computer pair is effectual, i.e. meets the user task goal. Usable HCI requires less intellectual, affective or conative[9] effort, i.e. is intuitive. Reliable HCI avoids or recovers from unintended user errors by checks or undo choices — the web Back button is an HCI invention. Flexible HCI lets users change language, font size or privacy preferences, as each person is a new environment to the software. Secure HCI avoids identity theft by user password. Extendible HCI lets users use what

9. Conative refers to the will; affective refers to the emotions; while intellectual refers to thoughts. All are cognitions that form from perceptions.

others create, e.g. mash-ups and third party add-ons. Connected HCI communicates with others, while privacy includes not getting spammed or being located on a mobile device.

Each level applies the same ideas to a different system view. The community level is discussed in Chapter 3.

| Requirement | Synonyms |
|---|---|
| *Functionality* | Effectualness, capability, usefulness, effectiveness, power, utility. |
| *Usability* | Ease of use, simplicity, user friendliness, efficiency, accessibility. |
| *Extendibility* | Openness, interoperability, permeability, compatibility, standards. |
| *Security* | Defence, protection, safety, threat resistance, integrity, inviolable. |
| *Flexibility* | Adaptability, portability, customizability, plasticity, agility, modifiability. |
| *Reliability* | Stability, dependability, robustness, ruggedness, durability, availability. |
| *Connectivity* | Networkability, communicability, interactivity, sociability. |
| *Privacy* | Tempest proof, confidentiality, secrecy, camouflage, stealth, encryption. |

TABLE 2.3: Performance requirement synonyms.


FIGURE 2.2: Firefox add-ons are extendibility.



## 2.8 A GENERAL DESIGN SPACE

Figure 2.3 shows a general system design space, where the:

- *Area* is the system performance requirements met.
- *Shape* is the environment requirement weightings.
- *Lines* are design requirement tensions.

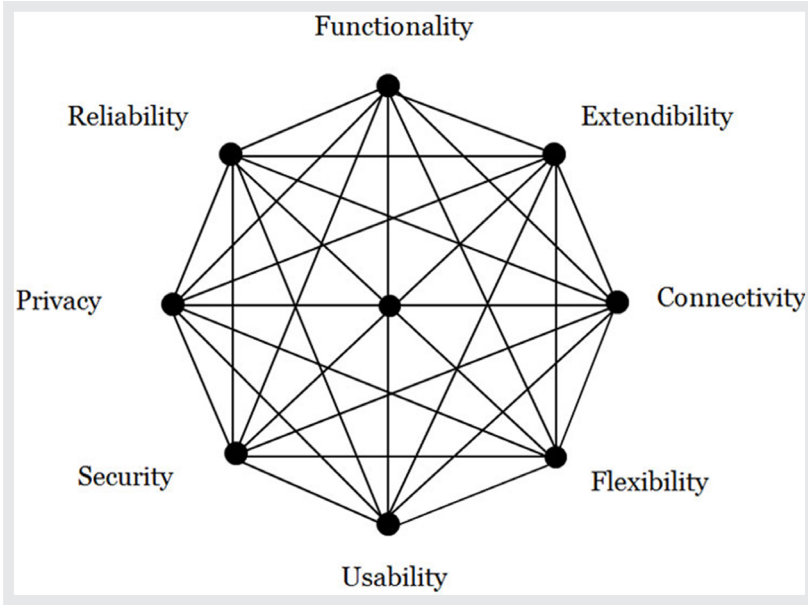



FIGURE 2.3: A general system design space.

*Courtesy of Brian Whitworth and Adnan Ahmad. Copyright: CC-Att-SA-3 (Creative Commons Attribution-ShareAlike 3.0).*

The space has four active requirements that are about taking opportunities[10] and four passive ones that are about reducing risks[11]. In system performance, taking opportunities is as important as reducing risk (Pinto, 2002).

The weightings of each requirement vary with the environment, e.g. security is more important in threat scenarios and extendibility more important in opportunity situations.

The requirement criteria of Figure 2.3 have no inherent contradictions, e.g. a bullet-proof plexi-glass room can be secure but not private, while encrypted files can be private but not secure. Reliability provides services while security denies them (Jonsson, 1998), so a system can be reliable but insecure, unreliable but se-

10. The active requirements are: functionality, flexibility, extendibility, connectivity.

11. The passive requirements are: security, reliability, privacy, usability.

cure, unreliable and insecure or reliable and secure. Likewise, functionality need not deny usability (Borenstein & Thyberg, 1991) or connectivity privacy. Cross-cutting requirements (Moreira et al., 2002) can be reconciled by innovative design if they are logically modular.

## 2.9 DESIGN TENSIONS AND INNOVATION

A design tension is when making one design requirement better makes another worse. Applying two different requirements to the same constituent often gives a design tension, e.g. castle walls that protect against attacks but have a gate to get supplies in. Computers that deny virus attacks but still need plug-in software hooks. These contrasts are not anomalies, but built into the nature of systems.

Design begins with no tensions. As requirements are met the Figure 2.3 performance area increases, so the lines between them tighten like rubber bands stretched. In advanced systems, the tension is so “tight” that increasing any performance criterion will pull back one or more others. In the *Version 2 Paradox*, a successful product improved to version 2 actually performs worse!

| Constituent | Code | Criteria | Analysis | Testing |
|---|---|---|---|---|
| **Actions** | Application | *Functionality* | Task | Business |
| | Interface | *Usability* | Usability | User |
| **Interactions** | Access control | *Security* | Threat | Penetration |
| | Plug-ins | *Extendibility* | Standards | Compatibility |
| **Changes** | Error recovery | *Reliability* | Stress | Load |
| | Preferences | *Flexibility* | Contingency | Situation |
| **Interchange** | Network | *Connectivity* | Channel | Communication |
| | Rights | *Privacy* | Legitimacy [12] | Social |

TABLE 2.4: Project specializations by constituent.

To improve a complex system one cannot just improve one criterion, i.e. just pull one corner of its performance web. For example, in 1992 Apple CEO Sculley introduced the hand-held Newton, claiming that portable computing was the future (Figure 2.4). We now know he was right, but in 1998 Apple dropped the line due to poor sales. The Newton's small screen made data entry hard, i.e. the portability gain was nullified by a usability loss. Only when Palm's Graffiti language improved handwriting recognition did the personal digital assistant (PDA) market revive. Sculley's portability innovation was only half the design answer — the other half was resolving the usability problems that the improvement had created. Innovative design requires cross-disciplinary generalists to resolve such design tensions.

12. Chapter 3 explains legitimacy analysis.

In general system design, too much focus on any one criterion gives diminishing returns, whether it is functionality, security (OECD, 1996), extendibility (De Simone & Kazman, 1995), privacy (Regan, 1995), usability (Gediga et al., 1999) or flexibility (Knoll & Jarvenpaa, 1994). Improving one aspect alone of a performance web can even reduce performance, i.e. "bite back" (Tenner, 1997), e.g. a network that is so secure no-one uses it. Advanced system performance requires balance, not just one dimensional design "excellence".

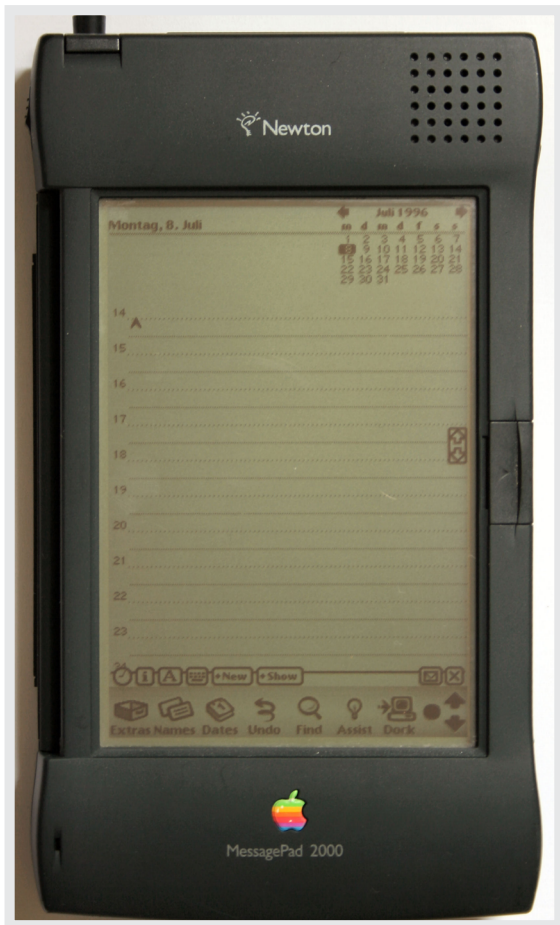



Figure 2.4: Sculley introduced the Newton PDA in 1992.

*Courtesy of Ralf Pfeifer. Copyright: CC-Att-SA-3 (Creative Commons Attribution-ShareAlike 3.0).*

## 2.10 PROJECT DEVELOPMENT

The days when programmers could list a system's functions and then just code them are gone, if they ever existed. Today, design involves not only many specialties but also their interaction. A system development could involve up to eight specialist groups, with distinct requirements, analysis and testing (Table 2.4). Smaller systems might have four groups (actions, interactions, changes and interchanges), two (opportunities and risks) or just one (performance). Design tensions can be reduced by agile methods where specialists talk more to each other and stakeholders, but advanced system development also needs innovators who can cross specialist boundaries to resolve cross-cutting design tensions.

## 2.11 DISCUSSION QUESTIONS

The following questions are designed to encourage thinking on the chapter and exploring socio-technical cases from the Internet. If you are reading this chapter in a class - either at university or commercial – the questions might be discussed in class first, and then students can choose questions to research in pairs and report back to the next class.

1. What three widespread computing expectations did not happen? Why not? What three unexpected computing outcomes did happen? Why?
2. What is a system requirement? How does it relate to system design? How do system requirements relate to performance? Or to system evaluation criteria? How can one specify or measure system performance if there are many factors?
3. What is the basic idea of general systems theory? Why is it useful? Can a cell, your body, and the earth all be considered systems? Describe Lovelock's Gaia Hypothesis. How does it link to both General Systems Theory and the recent film Avatar? Is every system contained within another system (environment)?

4. Does nature have a best species? If nature has no better or worse, how can species evolve to be better? Or if it has a better and worse, why is current life so varied instead of being just the "best"?[13] Does computing have a best system? If it has no better or worse, how can it evolve? If it has a better and worse, why is current computing so varied? Which animal actually is "the best"?
5. Why did the electronic office *increase* paper use? Give two good reasons to print an email in an organization. How often do you print an email? When will the use of paper stop increasing?
6. Why was social gaming not predicted? Why are MMORPG human opponents better than computer ones? What condition must an online game satisfy for a community to "mod" it (add scenarios)?
7. In what way is computing an "elephant"? Why can it not be put into an academic "pigeon hole"?[14] How can science handle cross-discipline topics?
8. What is the first step of system design? What are those who define what a system should do called? Why can designers not satisfy every need? Give examples from house design.
9. Is reliability an aspect of security or is security an aspect of reliability? Can both these things be true? What are reliability and security both aspects of? What decides which is more important?
10. What is a design space? What is the efficient frontier of a design space? What is a design innovation? Give examples (not a vacuum cleaner).
11. Why did the SGML academic community find Tim Berners-Lee's WWW proposal of low quality? Why did they not see the performance potential? Why did Microsoft also find it "of no business value"? How did the WWW eventually become a success? Given that business and academia now use it extensively, why did they reject it initially? What have they learnt from this lesson?

13. Success in evolutionary terms is what survives. Over 99% of all species that ever existed are now extinct, so every species existing today is a great success. Bacteria and viruses are as evolved as us in evolutionary terms.

14. A pigeon-hole is a small space used to roost pigeons.

12. Are NFRs like security different from functional requirements? By what logic are they less important? By what logic are they equally critical to performance?
13. In general systems theory (GST), every system has what two aspects? Why does decomposing a system into simple parts not fully explain it? What is left out? Define holism. Why are highly holistic systems also individualistic? What is the Frankenstein effect? Show a "Frankenstein" web site. What is the opposite effect? Why can "good" system components not be stuck together?
14. What are the elemental parts of a system? What are its constituent parts? Can elemental parts be constituent parts? What connects elemental and constituent parts? Give examples.
15. Why are constituent part specializations important in advanced systems? Why do we specialize as left-handers or right-handers? What about the ambidextrous?
16. If a car is a system, what are its boundary, structure, effector and receptor constituents? Explain its general system requirements, with examples. When might a vehicle's "privacy" be a critical success factor? What about its connectivity?
17. Give the general system requirements for browser application. How did its designers meet them? Give three examples of browser requirement tensions. How are they met?
18. How do mobile phones meet the general system requirements, first as hardware and then as software?
19. Give examples of usability requirements for hardware, software and HCI. Why does the requirement change by level? What is "usability" on a community level?
20. Are reliability and security really distinct? Can a system be reliable but insecure, unreliable but secure, unreliable and insecure, or reliable and

secure? Give examples. Can a system be functional but not usable, not functional but usable, not functional or usable, or both functional and usable? Give examples.

21. Performance is taking opportunities and avoiding risks. Yet while mistakes and successes are evident, missed opportunities and mistakes avoided are not. Explain how a business can fail by missing an opportunity, with WordPerfect vs Word as an example. Explain how a business can succeed by avoiding risks, with air travel as an example. What happens if you only maximize opportunity? What happens if you only reduce risks? Give examples. How does nature both take opportunities and avoid risks? How should designers manage this?
22. Describe the opportunity enhancing general system performance requirements, with an IT example of each. When would you give them priority? Describe the risk reducing performance requirements, with an IT example of each. When would you give them priority?
23. What is the Version 2 paradox? Give an example from your experience of software that got worse on an update. You can use a game example. Why does this happen? How can designers avoid this?
24. Define extendibility for any system. Give examples for a desktop computer, a laptop computer and a mobile device. Give examples of software extendibility, for email, word processing and game applications. What is personal extendibility? Or community extendibility?
25. Why is innovation so hard for advanced systems? Why stops a system being secure and open? Or powerful and usable? Or reliable and flexible? Or connected and private? How can such diverse requirements ever be reconciled?
26. Give two good reasons to have specialists in a large computer project team. What happens if they disagree? Why are cross-disciplinary integrators also needed?

## YOUR NOTES AND THOUGHTS ON CHAPTER 2

Record your notes and thoughts on this chapter. If you want to share these thoughts with others online, go to the bottom of the page at: http://www.interaction-design.org/books/the_social_design_of_technical_systems_2nd_ed/design_spaces.html

**NOTES:**

# CHAPTER 3

# Socio-Technical Design

"*Just because you can, doesn't mean you should.*"

Social ideas like freedom seem far removed from computer code but computing today *is* social. That technology designers are not ready, have no precedent or do not understand social needs is irrelevant. Like a baby being born, online society is pushing forward, whether we are ready or not. And like new parents, socio-technical designers are causing it, again ready or not. As the World Wide Web's creator observes:

"*... technologists cannot simply leave the social and ethical questions to other people, because the technology directly affects these matters*"

*-- Berners-Lee, 2000: p124*

Socio-technical design is the application of social and ethical requirements to HCI, software and hardware systems.

## 3.1 DESIGNING WORK MANAGEMENT

The term socio-technical was first introduced by the UK Tavistock Institute[1] in the late 1950s to oppose Taylorism — reducing jobs to efficient elements on assembly lines in mills and factories (Porra & Hirschheim, 2007). Community level performance needs applied to the personal level gave work-place management ideas including:

1. *Congruence*. Let the process match its objective — democratic results need democratic means.
2. *Minimize control*. Give employees clear goals but let them decide how to achieve them.
3. *Local control*. Let those with the problem change the system, not absent managers.
4. *Flexibility*. Without "extra" skills to handle change, specialization will precede extinction.
5. *Boundary innovation*. Innovate at the boundaries, where work goes between groups.
6. *Transparency*. Give information first to those it affects, e.g. give work rates to workers.
7. *Evolution*. Work system development is an iterative process that never stops.
8. *Lead by example*. As the Chinese say: "If a General takes an egg, his soldiers will loot a village."[2]
9. *Support human needs*. Work that lets people learn, choose, feel and belong gives loyal staff.

1. See http://www.strategosinc.com/sociotechnical.htm

2. While Steve Jobs worked for $1 per year, many other CEOs take as much as they can get - because being in charge means they can.

In computing it became a call for the ethical use of technology.

This book extends that foundation, to apply social requirements to technology design as well as work design, because technologies now mediate social interactions.

During the industrial revolution, when the poet William Blake wrote of "dark satanic mills", technology was seen by many as the heartless enemy of the human spirit. Yet *people* ran the factories that were enslaving people. The industrial revolution was the rich using the poor as they had always done, but machines let them do it better. Technology was the effect magnifier but not in itself good or evil. Today, we have largely rejected slavery but we embrace technology like the car or cell phone. Today technology is on the other side of the class war, as Twitter, Facebook and YouTube support the Arab spring. Yet the core socio-technical principle is still:

"*Just because you can, doesn't mean you should*"

Socio-technical design puts social needs above technical wants. The argument is that human evolution involves social and technical progress in that order, e.g. today's vehicles could not work on the road without today's citizenry. Technology structures like cars also need social structures like road rules.

Social inventions like credit were as important in our evolution as technical inventions like cars. Global trade needs not only ships and aircraft to move goods around the world, but also people willing to do that. Today, online traders send millions of dollars to people they have not seen for goods they have not touched to arrive at times unspecified. A trader in the middle ages, or indeed in the early twentieth century, would have seen that as pure folly. What has changed is not just the technology but also the society. Today's global markets work by social *and* technical support:

*"To participate in a market economy, to be willing to ship goods to distant destinations and to invest in projects that will come to fruition or pay dividends only in the future, requires confidence, the confidence that ownership is secure and payment dependable. ... knowing that if the other reneges, the state will step in..."*

*-- Mandelbaum, 2002: p272.*

## 3.2 SOCIAL REQUIREMENTS

One cannot design socio-technology in a social vacuum. Fortunately, while virtual society is new, people have been socializing for thousands of years. We know that fair communities prosper but corrupt ones do not (Eigen, 2003). Social inventions like laws, fairness, freedom, credit and contracts were bought with blood and tears (Mandelbaum, 2002), so why start anew online? Why reinvent the social wheel in cyber-space (Ridley, 2010)? Why re-learn electronically what we already know physically?

As the new bottle of information technology fills with the old wine of society, the stakes are raised. The information revolution increases our power to gather, store and distribute information, for good or ill (Johnson, 2001). Are we the hunter-gatherers of the information age (Meyrowitz, 1985) or an online civilization? A stone-age society with space-age technology is a bad mix, but what are the requirements for technology to support civilization? Computing cannot implement what it cannot specify.

We live in social environments every day, but struggle to specify them[3], e.g. a shop-keeper swipes a credit card with a reading device designed *not* to store credit card number or pin. It is designed to the social requirement that shopkeepers do not steal customer data, even if they can. Without this, credit would col-

3. We are social environment blind.

lapse, and social disasters like the depression can be worse than natural disasters. Credit card readers support legitimate social interaction *by design*.

Likewise, if online computer systems take and sell customer data such as home addresses and phone numbers for advantage, users will lose trust and either refuse to register at all, or register with fake data, like "123 MyStreet, MyTown, NJ" (Foreman & Whitworth, 2005). The way to satisfy online privacy is not to store data you do not need. To say it will never be revealed is not good enough, as companies can be forced by governments or bribed by cash to reveal data. One cannot be forced or bribed to give data one does not have. The best way to guarantee online trust is to not to store unneeded information in the first place[4].

## 3.3 THE SOCIO-TECHNICAL GAP

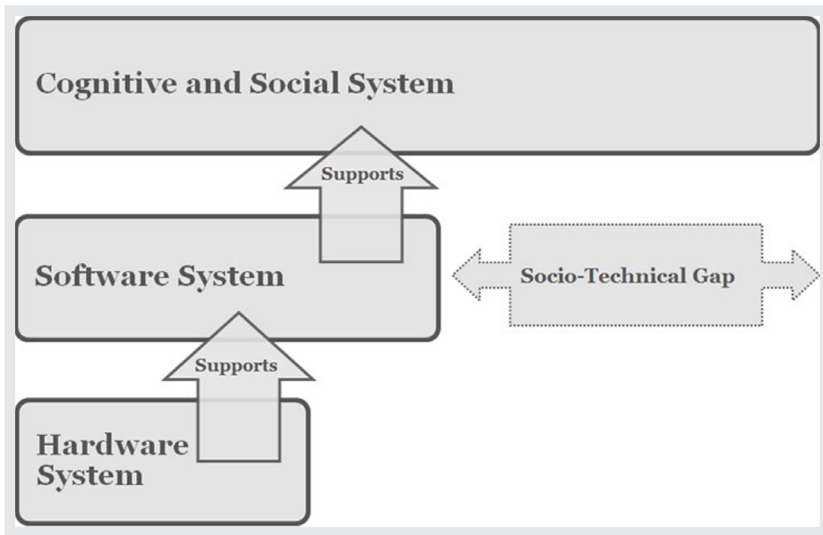



Figure 3.1: The socio-technical gap.

*Courtesy of Brian Whitworth and Adnan Ahmad. Copyright: CC-Att-SA-3 (Creative Commons Attribution-ShareAlike 3.0).*

4. Trying to gather all the information you can is information greediness.

Simple technical design gives a *socio-technical gap* (Figure 3.1), between what the technology allows and what people want (Ackerman, 2000). For example, email technology ignored the social requirement of privacy, letting anyone email anyone without permission, and so gave spam. The technical level response to this social level problem was inbox filters. They help on a local level, but transmitted spam as a system problem has never stopped growing. User *inbox* spam has been held constant due to filters, but the percentage of spam *transmitted by the Internet* commons has never stopped growing. It grew from 20% to 40% of messages in 2002-2003 (Weiss, 2003), to 60-70% in 2004 (Boutin, 2004), to 86.2% to 86.7% of the 342 billion emails sent in 2006 (MAAWG, 2006; MessageLabs, 2006), to 87.7% in 2009 and 89.1% of all emails sent in 2010 (MessageLabs, 2010). A 2004 prediction that within a decade over 95% of all emails transmitted by the Internet will be spam is coming true (Whitworth & Whitworth, 2004).

Due to spam, email users are moving to other media, but if we make the same socio-technical design error there, the problem will just follow, e.g. SPIM is instant messaging spam.

Filters see spam as a *user* problem but it is really a *community* problem — a social dilemma. Transmitted spam uses Internet storage, processing and bandwidth, whether users hiding behind their filter walls see it or not. Only socio-technology can resolve social problems like spam because in the "spam wars" technology helps both sides, e.g. image spam can bypass text filters, AI can solve captchas[5], botnets can harvest web site emails, and zombie sources can send emails. Spam is not going away any time soon (Whitworth and Liu, 2009a).

Right now, aliens observing our planet might think email is built for machines, as most of the messages transmitted go from one (spam) computer to another (filter) computer, untouched by human eye. This is not just due to bad luck. A communication technology is not a Pandora's box whose contents are unknown

5. CAPTCHA stands for Completely Automated Public Turing test to tell Computers and Humans Apart.

until opened *because we built it*. Spam happens when we build technologies instead of socio-technologies.

## 3.4 LEGITIMACY ANALYSIS

A social system is here *an agreed form of social interaction that persists* (Whitworth and de Moor, 2003). People seeing themselves as a community is, therefore, essentially their choice. In politics, a *legitimate* government is seen as rightful by its citizens, i.e. accepted. In contrast, illegitimate governments need to stay in power by force of arms and propaganda. By extension, *legitimate interaction* is accepted by the parties involved, who freely repeat it, e.g. fair trade. Legitimacy has been specified as: *fairness* and *public good* (Whitworth and de Moor, 2003). Physical and online citizens prefer legitimate communities because they perform better.

In physical society, legitimacy is maintained by laws, police and prisons that punish criminals. Legitimacy is the human level concept by which judges create new laws and juries decide on never before seen cases. A *higher affects lower* principle applies: communities engender human ideas like fairness, which generate informational laws, that are used to govern physical interactions. Communities of people create rules to direct acts that benefit the community, i.e. higher level goals drive lower level operations to improve system performance. Doing this online, applying social principles to technical systems, is socio-technical design.

Conversely, over time lower levels get a "life of their own" and the tail starts to wag the dog, e.g. copyright laws originally designed to encourage innovators have become a tool to perpetuate the profit of corporations like Disney that purchased those creations (Lessig, 1999)[6]. Unless continuously "re-invented" at the human level, information level laws inevitably decay and cease to work.

6. Disney copyrighted public domain stories like Snow White that they did not create, and then stopped others using them.

Lower levels are more obvious, so it is easy to forget that today's online society is a social *and* a technical evolution. The Internet is new technology but it is also a move to new social goals like service and freedom rather than control and profit. So for the Internet to become a hawker market, of web sites yelling to sell, would be a devolution. The old ways of business, politics and academia should follow the new Internet way, not the reverse.

There are no shortcuts in this social evolution, as one cannot just "stretch" physical laws into cyberspace (Samuelson, 2003), because they often:

1. *Do not transfer* (Burk, 2001), e.g. what is online "trespass"?
2. *Do not apply*, e.g. what law applies to online "cookies" (Samuelson, 2003)?
3. *Change too slowly,* e.g. laws change over years but code changes in months.
4. *Depend on code* (Mitchell, 1995), e.g. anonymity means actors cannot be identified.
5. *Have no jurisdiction.* U.S. law applies to U.S. soil, but cyber-space is not "in" America.

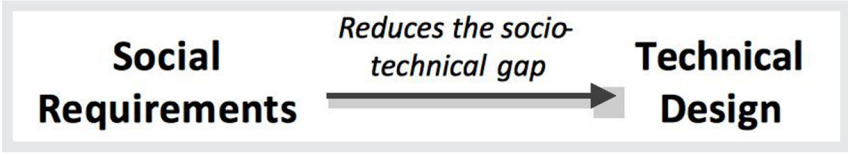



FIGURE 3.2: Legitimacy analysis.



The software that mediates online interaction has by definition full control of what happens, e.g. any application could upload any hard drive file on your computer to any server. In itself, code could create a perfect online police state, where everything is monitored, all "wrong" acts punished and all undesirables excluded, i.e. a perfect tyranny of code.

Yet code is also an opportunity to be *better than the law,* based on legitimacy analysis (Figure 3.2). Physical justice, by its nature, operates after the fact, so one must commit a crime to be punished. Currently, with long court cases and appeals, justice can take years, and justice delayed is justice denied. In contrast, code represents the online social environment directly, as it acts right away. It can also be designed to enable social acts as well as to deny anti-social ones. If online code *is* law (Lessig, 1999), to get legitimacy online we must build it into the system design, knowing that legitimatesocio-technical systems perform better (Whitworth and de Moor, 2003). That technology can positively support social requirements like fairness is the radical core of socio-technical design.

So is every STS designer an application law-giver? Are we like Moses coming down from the mountain with tablets of code instead of stone? Not quite, as STS directives are to software not people. Telling people to act rightly is the job of ethics not software, however "smart". The job of right code, like right laws, is to attach outcomes to social acts, not to take over people's life choices. Code as the social environment cannot be a social actor. Socio-technical design is socializing technology to offer fair choices, not technologizing society to be a machine with no choice at all. It is the higher directing the lower, not the reverse.

To achieve online what laws do offline, STS developers must re-invoke legitimacy for each application. It seems hard, but every citizen on jury service already does this, i.e. interpret the "spirit of the law" for specific cases. STS design is the same but for application cases. That the result is not perfect doesn't matter. Cultures differ but all have some laws and ethics, because some higher level influence is always better than none.

To try to build a community as an engineer builds a house is a levels error, i.e. choosing the wrong level for the job. Social engineering by physical coercion, oppressive laws, or propaganda and indoctrination is people using others as objects,

i.e. anti-social. A community is by definition many people, so an elite few enslaving the rest is not a community. Social engineering treats people like bricks in a wall which denies social requirements like freedom and accountability. Communities cannot be "built" from citizen actors because to treat them as objects denies their humanity. Communities can emerge as people interact, but they cannot be built.

## 3.5 THE WEB OF SOCIAL PERFORMANCE

Communities interact with others, using spies to act as "eyes", diplomats to communicate, engineers to effect, soldiers to defend, intellectuals to adapt and traders to extend, but a community can also interact *with itself,* to communicate or synergize, as follows:

1. *Productivity*. As functionality is what a system can produce, so communities produce bridges, art and science by citizen competence, based on education of symbolic knowledge, ethics and tacit skills. Help and FAQ systems illustrate this for an online community.
2. *Synergy*. As usability is less effort per result, so communities improve efficiency by synergy based on trust[7]. Public goods like roads and hospitals involve functional specialists giving to all. If all citizen specialists offer their services to others, all get more for much less effort. Wikipedia illustrates online synergy, as many specialists give a little knowledge so that all get a lot of knowledge.

   Both productivity and synergy require citizen participation, based on skills and trust respectively, yet are also in tension as one invokes competition and the other cooperation[8]. The former improves how citizen "parts" perform and the latter how they interact. *Service* by citizens reconciles productivity and synergy, as it invokes both.

7. In a society, if everyone gives, everyone gets, but if everyone takes, everyone is taken from.
8. The problem with competition is that **if you give peanuts you get monkeys but if you give honey you get wasps**.

3. *Freedom*. As flexibility is a system's ability to change to fit a changing environment, so communities become flexible when citizens have freedom, i.e. the right to act autonomously. This is the right not to be a slave[9]. Freedom allows local resource control, which increases social performance just as decentralized protocols like Ethernet improve network performance.
4. *Order*. Reliability is a system's ability to survive internal part failure or error. A community achieves reliability through order, when citizens, by rank, role or job, know and do their duty. Some cultures set up warrior or merchant castes to achieve this. Online order is also by roles, e.g. Sysop or Editor.

   Freedom and order are in tension, as freedom has no class but order does. The social invention of *democracy* merges freedom and order by letting citizens chose their governance, not just of the President or Prime Minister, but for all positions. Democracy is rare online, but Slashdot uses it.
5. *Ownership*. Security is a system's defence against outside takeover. A community is made secure internally by ownership, as to "own" a house guarantees that if another takes it, the community will step in[10]. Online, ownership works by access control (see Chapter 6).
6. *Openness*. Extendibility is a system's ability to use what is outside itself. A community doing this is illustrated by America's invitation to the world:

   "*Give me your tired, your poor, your huddled masses yearning to breathe free*"

   A society is open internally if any citizen can achieve any role by merit, just as Abraham Lincoln, born in a log cabin, became US president. The opposite is nepotism or cronyism, giving jobs to family or friends regardless of

9. Physical slavery is tyranny, informational slavery is propaganda and psychological slavery is political correctness, where the few tell the many how to think. Wanting to live others' lives is a sign of emptiness.

10. State ownership, as in communism or tribalism, is still ownership. Only if a state gives its ownership away do citizens get freedom.

merit. If community advancement is by who you know, not what you know, performance reduces. Open source systems like Source Forge let people advance by merit.

Ownership and openness are in tension, as the right to keep out denies the right to go in. *Fairness* can reconcile public access and private control. Offline fairness is supported by justice systems but online fairness must be supported by code.

7. *Transparency*. As connectivity lets a person communicate with others, so transparency lets a community communicate with itself by media like TV, newspapers, radio and now the Internet. In a transparent community, people can easily see what the group is doing but in an opaque one they cannot. Transparent governance lets citizens view public money spent and privileges given, as public acts on behalf of a community are not private. Systems like Wikipedia illustrate transparency online.
8. *Privacy*. Privacy as a citizen's right to control their personal information can also describe a community right given to all citizens, i.e. the same term works at the community level. In this case, it includes communication privacy, the right not to be monitored without consent.

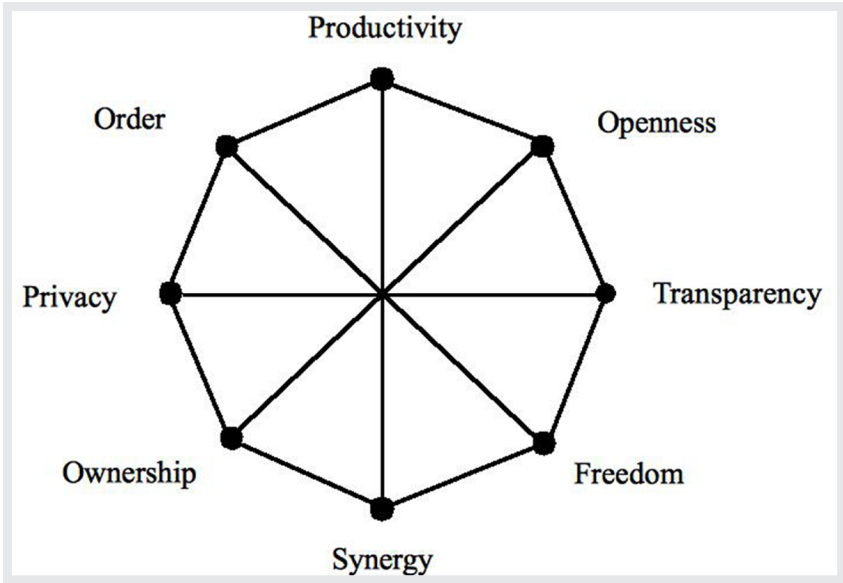



Figure 3.3: A social web of performance.

*Copyright status: Unknown (pending investigation). See section "Exceptions" in the copyright terms below.*

Transparency and privacy are in natural tension, as giving the right to see another may deny their right not to be seen. *Politeness* reconciles this tension by letting people of different rank, education, age and experience connect. Further details are given in Chapter 4.

The social invention of *rights* can also reconcile many aspects of the web of social performance, as discussed in Chapter 6.

In the performance web of Figure 3.3, a community increases citizen competence to be productive, increases trust to get synergy, gives freedoms to adapt and innovate, establishes order to define responsibilities, allocates ownership to prevent property conflicts, is open to outside and inside talent[11], communicates internally to generate agreement, and grants legitimate rights that stabilize social interaction. As all these together increase social performance but are in design tension, so each community must define its own social structure, whether online or off.

11. Sexism and racism are community level losses. If women cannot work, half the population cannot add to its productivity. If a race, like black people, are excluded, so are their contributions.

## 3.6 COMMUNICATION FORMS

Communication media transmit meaning between senders and receivers. Meaning is any change in a person's thoughts, feelings or motives[12]. The *communication performance* of a transmission is the total meaning exchanged, i.e. its human impact.It can be based on message richness and sender-receiver linkage as follows:

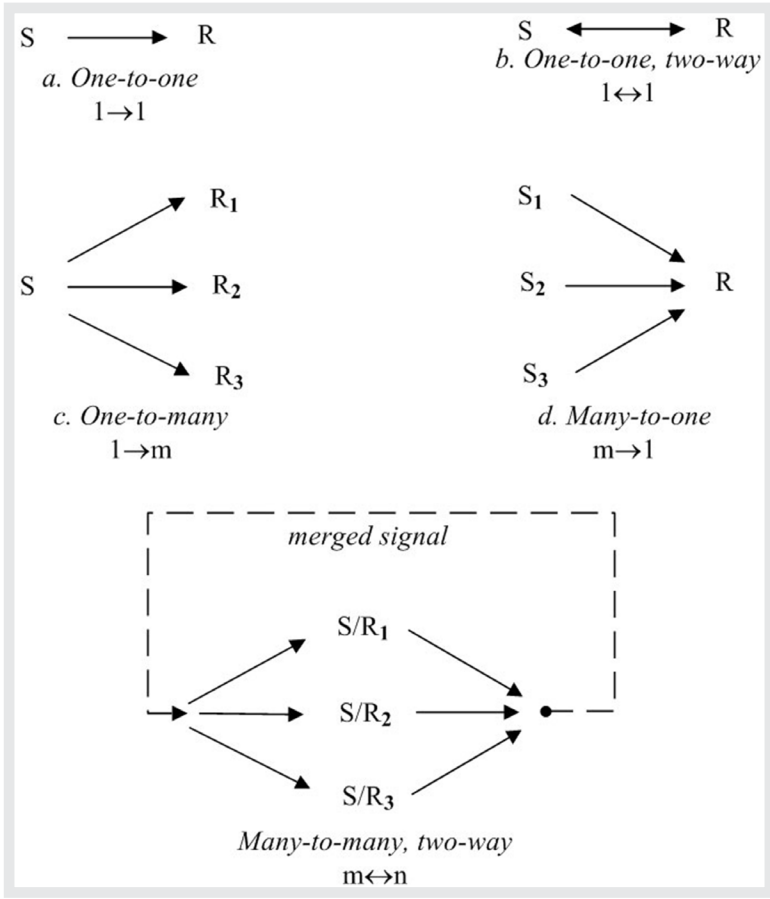



FIGURE 3.4: Communication linkage (S = Sender, R = Receiver).

*Courtesy of Brian Whitworth and Adnan Ahmad. Copyright: CC-Att-ND-3 (Creative Commons Attribution-NoDerivs 3.0 Unported).*

12. In psychology, *intellectual* refers to thoughts, *affective* refers to the emotions and *conative* refers to the will. All are cognitions formed from perceptions.

1. *Richness* is the amount of meaning a message conveys. To see video as automatically richer than text confuses meaning richness with information bandwidth. Meaning is the human impact, so texting "I'm safe" can have more meaning than a multi-media video[13]. *Media richness* can be classified by the symbols that generate meaning:

    a. *Position*. A single, static symbol, e.g. to raise one's hand. A vote is a position.
    b. *Document*. Many static symbols form a meaning pattern as words form a sentence by syntax or pixels form an object by Gestalt principles. Documents are text or pictures.
    c. *Dynamic-media (Audio)*. A dynamic channel with multiple semantic streams, e.g. speech has tone of voice and content[14]. Music has melody, rhythm and timbre.
    d. *Multi-media (Video)*. Many dynamic channels, e.g. video is both audio and visual channels. Face-to-face communication uses many sensory channels.

    Rich media have the potential to transfer more meaning.
2. *Linkage*. The meaning exchanged in a communication also depends on the sender-receiver link (Figure 3.4), namely:

    a. *Interpersonal* (one-to-one, two-way): Both parties can send and receive, usually signed.
    b. *Broadcast* (one-to-many, one-way): From one sender to many receivers, can be unsigned.
    c. *Matrix* (many-to-many, two-way): Many senders to many receivers, usually unsigned.

13. One may be overwhelmed to hear a text message that a loved one is safe, but indifferent to a multi-media sales video.

14. A semantic stream is the meaning produced by type of processing. One physical channel processed differently can have many semantic streams, e.g. tone of voice and message content are different streams.

Communication performance depends on both richness and linkage, as a message sent to one person that is then broadcast to many increases the human impact. At the individual level people communicate with people, but community level communication is group-to-group, i.e. many send and many receive in one transmit operation. Matrix communication is one-to-many (broadcast) plus many-to-one (merge) communication, where addressing an audience is one-to-many communication and applauding a speaker is many-to-one. The applauding audience to itself is matrix communication, as the group producing the clapping message also receives it. Matrix communication allows normative influence, e.g. when audiences start and stop clapping together. A choir singing is also matrix communication, so by normative influence if choirs go off key they usually do so together.


FIGURE 3.5: Tagging as matrix communication.

*Courtesy of Brian Whitworth and Adnan Ahmad. Copyright: CC-Att-ND-3 (Creative Commons Attribution-NoDerivs 3.0 Unported).*

Face-to-face groups also use matrix communication, as body language and facial expressions convey each member of the group's position on an issue to everyone

else in the group. A valence index calculated from member position indicators was found to predict a group discussion outcome as well as the words (Hoffman & Maier, 1961). Matrix communication is how online electronic groups can form social agreement without any rich information exchange or discussion, using only anonymous, lean signals, (Whitworth et al., 2001). Community voting, as in an election, is a physically slow matrix communication that computers can speed up. Tag cloud, reputation system and social book-mark technologies all illustrate online support for matrix communication (Figure 3.5).

If communication performance is richness plus linkage, a regime bombarding citizens 24/7 with TV/video propaganda can exchange less meaning by this one-to-many linkage than people talking freely by twitter, which is many-to-many linkage.

## 3.7 A MEDIA FRAMEWORK

Table 3.1 categorizes various communication media by richness and linkage with electronic media in italics, e.g. a phone call is an inter-personal audio but a letter is interpersonal text. A book is a document broadcast, radio is broadcast audio and TV is broadcast video. The Internet can broadcast documents (web sites), audio (podcasts) or videos (YouTube). Email allows two-way interpersonal text messages, while Skype adds two-way audio and video. Chat is few-to-few matrix text communication, as is instant messaging but with known people. Blogs are text broadcasts that also allow comment feedback. Online voting is matrix communication, as many communicate with many in one operation.

| | **Linkage** | | |
|---|---|---|---|
| *Richness* | *Broadcast* | *Interpersonal* | *Matrix* |
| *Position* | Footprint, Flare, Scoreboard, Scream, | Posture, Gesture, Acknowledgement, Salute, *Smiley* | Show of hands, Applause, An election, *Web counter, Karma system, Tag cloud, Online vote, Reputation system, Social bookmarks* |
| *Document* | Poster, Book, *Web site, Blog, Online photo, News feed, Online review, Instagram, Twitter*[1] | Letter, Note, *Email, Texting, Instant message, Social network*[2] | *Chat, Twitter*[1] *Wiki, E-market, Bulletin board, Comment system, Advice board, Social network*[2] |
| *Dynamic-media (Audio)* | Radio, Loud-speaker, Record, CD, *Podcast, Online music* | Telephone, Answer-phone, *Cell phone, Skype* | Choir, Radio talk-back, *Conference call, Skype conference call* |

| | | | |
|---|---|---|---|
| *Multi-media (Video)* | Speech, Show, Television, Movie, DVD, *YouTube video* | Face-to-face conversation, *Chatroulette, Video-phone, Skype video* | Face-to-face meeting, Cocktail party, *Video-conference, MMORPG Simulated world* |

1 Combines broadcast (text) and matrix (following).
2 Combines interpersonal and matrix.

Table 3.1: Communication media by richness and linkage.

Computers allow "anytime[15], anywhere" communication for less effort, e.g. an email is easier to send than posting a letter. Lowering the message threshold means that more messages are sent (Reid et al., 1996). Email stores a message until the receiver can view it[16], but a face-to-face message is ephemeral; it disappears if you are not there to get it. Yet being unable to edit a message sent makes sender state streams like tone of voice more *genuine*.

As media richness theory framed communication in one-dimensional richness terms, electronic communication was expected to move directly to video, but that was not what happened. EBay's reputations, Amazon's book ratings, Slashdot's karma, tag clouds, social bookmarks and Twitter are not rich at all. Table 3.1 shows that computer communication evolved by *linkage as well as richness*. Computer chat, blogs, messaging, tags, karma, reputations and wikis are all high linkage but low richness.

Communication that combines high richness and high linkage is interface expensive, e.g. a face-to-face meeting lets rich channels and matrix communica-

15. Asynchronous email communication lets senders ignore distance and time but synchronous communication like Skype does not. One cannot call someone who is sleeping. *The world is flat* (as Friedmann's book says) but the day is round.

16. Physical network asynchrony depends on the buffer capacity of its nodes.

tion give factual, sender state and group state information. The medium also lets real time contentions between people talking at once be resolved naturally. Everyone can see who the others choose to look at, so whoever gets the group focus continues to speak. To do the same online would require not only many video streams on every screen but also a mechanism to manage media conflicts. Who controls the interface? If each person controls his or her own view there is no commonality, while if one person controls the view like a film editor, there is no group action. Can electronic groups act democratically like face-to-face do?

In *audio-based tagging,* the person speaking automatically makes his or her video central (Figure 3.6). The interface is common but it is group-directed, i.e. democratic. *Gaze-based tagging* is the same except that when people look at a person, that person's window expands on everyone's screen. It is in effect a *group directed bifocal display* (Spence and Apperley, 2012). When matrix communication is combined with media richness, online meetings will start to match face-to-face meetings in terms of communication performance.


*Courtesy of Brian Whitworth and Adnan Ahmad;testing 2 author. Copyright: CC-Att-SA-3 (Creative Commons Attribution-ShareAlike 3.0).*


Figure 3.6 A-B: Audio based video tagging.

*Courtesy of Brian Whitworth and Adnan Ahmad. Copyright: CC-Att-SA-3 (Creative Commons Attribution-ShareAlike 3.0).*

A social model of communication suggests why video-phones are technically viable today but video-phoning is still not the norm. Consider its disadvantages, like having to shave or put on lipstick to call a friend or having to clean up the background before calling Mum. Some even prefer text to video because it is *less* rich, as they *do not* want a big conversation.

In sum, computer communication is about linkage as well as richness because communication acts involve senders and receivers as well as messages. Communication also varies by system level, as is now discussed.

## 3.8 SEMANTIC STREAMS

Communication goals can be classified by level as follows:

1. *Informational*. The goal is to analyze information about the world and decide a best choice, but this rational analysis process is surprisingly fragile (Whitworth et al., 2000).
2. *Personal*. The goal is to form relationships which are more reliable. Relating involves a turn-taking, mutual-approach process, to manage the emotional arousal evoked by the presence of others (Short et al., 1976)[17]. Friends are better than logic.
3. *Community*. The goal is to stay "within" the group, as belonging to a community means being part of it, and so protected by it. Communities outlast friends.

| Goal | Influence | Linkage | Questions |
|---|---|---|---|
| *Analyze task* information | *Informational* influence | *Broadcast* | What is right?<br>What is best? |
| *Relate to* other people | *Personal* influence | *Interpersonal* | Who do I like?<br>Who do I trust? |
| *Belong to a* community | *Normative* influence | *Matrix* | What are the others doing?<br>Am I "in" the group? |

TABLE 3.2: Human goals by influence and linkage.

Table 3.2 shows how the level goals map to influence and linkage types. People online or off *analyze* information, *relate* to others and *belong* to communities,

17. A Haka communicates between two Maori warbands to convey an intent as well as a state, see http://www.youtube.com/watch?v=c-lrE2JcO44

so they are subject to informational, personal and normative influences. Normative influence is based on neither logic nor friendship; an example is patriotism, loyalty to your country right or wrong. Even brothers may kill each other on a civil war battlefield.

People are influenced by community norms, friend views and task information, in that order, via different semantic streams. Semantic streams arise as people process a physical signal in different ways to generate different meanings. So *one* physical message can *at the same time* convey:

1. *Message content*. Symbolic statements about the literal world, e.g. a factual sentence.
2. *Sender state*. Sender psychological state, e.g. an agitated tone of voice.
3. *Group position*. Sender intent, which added up over many senders gives the group intent, e.g. an election.

Human communication is subtle because one message can have many meanings and people respond to them all at once. So when leaving a party I may *say:* "I had a good time" but by *tone* imply the opposite. I can say "I AM NOT ANGRY!" in an angry voice[18]. What is less obvious is that a message can also indicate a *position,* or intent to act, e.g. saying "I had a good time" in a certain tone or with certain body language can indicate an intention to leave a party.

In the cognitive model of Figure 3.7, physical level signals have as many semantic streams as the medium allows. Face-to-face talk allows three streams, but computer communication is usually more restricted, e.g. email text mainly just gives content information.

18. Some people may not process the sender state semantic stream, e.g. those with autism.

Community level systems using matrix communication include:

1. The reputation ratings of Amazon or E-Bay give community-based product quality control. Slashdot does the same for content, letting readers rate comments to filter out poor ones.
2. Social bookmarks, like Digg and Stumbleupon, let users share links, to see what the community is looking at.
3. Tag technologies increase the font size of links according to frequency of use. As people walk in a forest walk on tracks trod by others, so we can now follow web-tracks on browser screens.
4. Twitter's follow function lets leaders broadcast ideas to followers and people choose the leaders they like.

The power of the computer can allow very fast matrix communication by millions or billions. What might a global referendum on current issues reveal? The Internet could tell us.

How these three basic cognitive processes, of belonging, relating and fact analysis, work together in the same person is shown in Figure 3.7. The details are given elsewhere (Whitworth et al., 2000), but note that the rational resolution of task information is the third priority cognitive process, not the first. Only if all else fails do we stop to think things out or read intellectual instructions.

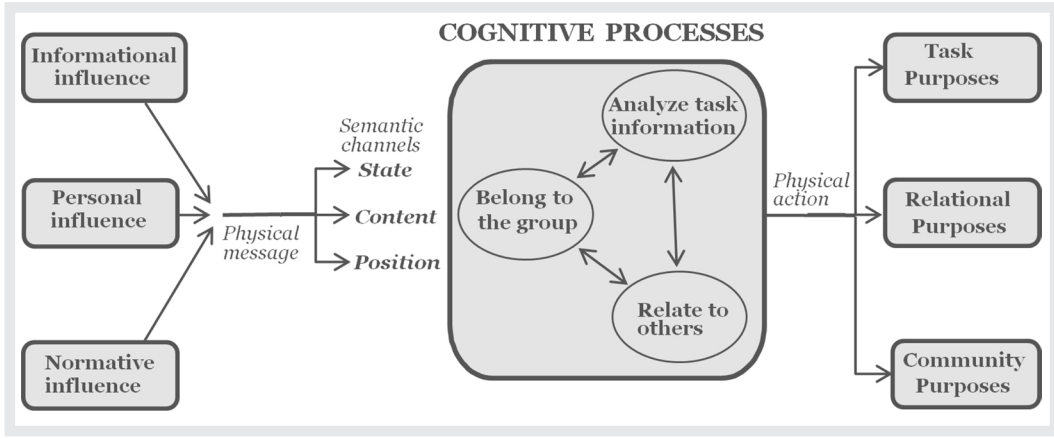



Figure 3.7: Cognitive processes in communication.

*Courtesy of Brian Whitworth and Adnan Ahmad. Copyright: CC-Att-SA-3 (Creative Commons Attribution-ShareAlike 3.0).*

The first level of the World Wide Web, an information library accessed by search tools, is well in place. Its second level, a medium for personal relations, is also well under way. The third, a civilized social environment, is the current and future challenge, as even a cursory study of Robert's Rules of Order will dispel any illusion that social dealings are simple (Robert, 1993).

Socio-technology lets hundreds of millions of people act together, but we still do not know what *Here Comes Everybody*[19] means (Shirky, 2008). How exactly many people act as one has always been a bit of a mystery, yet this is what we must learn to bring group applications to an Internet full of personal applications. Group writing and publishing, group browsing, group singing and music (online choirs and bands), group programming and group research illustrate the many areas of current and future potential in online group activity. One can also expect a bite-back in privacy demands, i.e. more "tight" communities that are hard to get in to.

19. The issue of community is illustrated as follows: *Question*: 'Where does an 800lb Gorilla sit when it comes to dinner?' *Answer*: Anywhere it wants to. Communities are like this, but must agree to act, which can take years

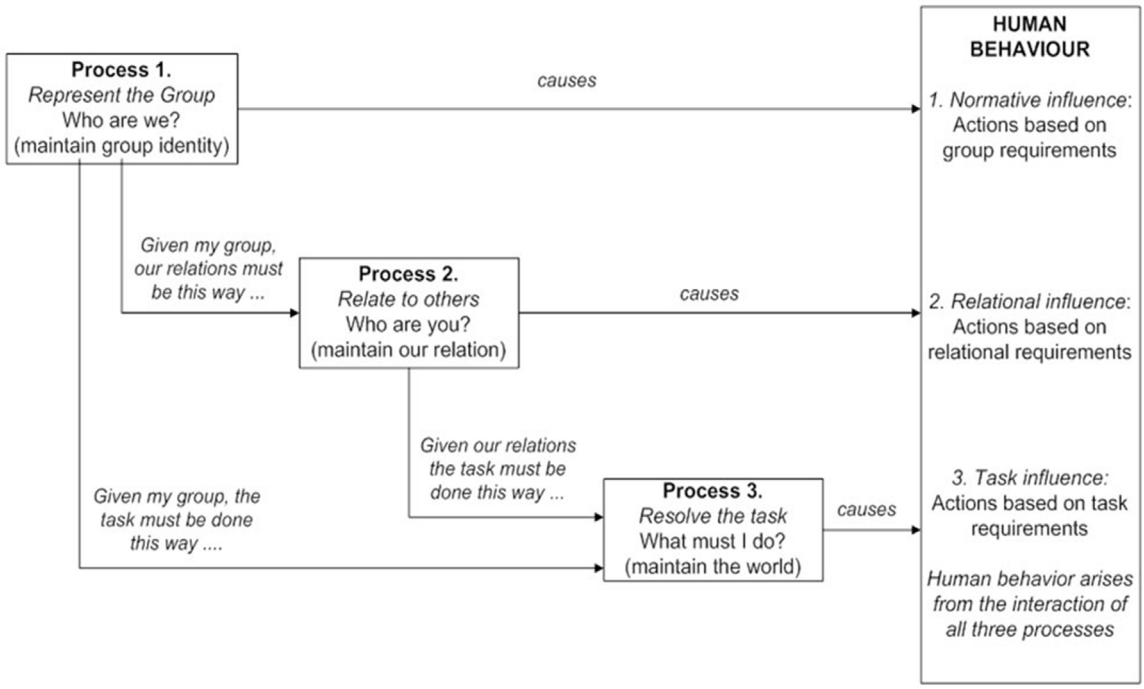



FIGURE 3.8: The cognitive processes that affect human behaviour.

*Courtesy of Brian Whitworth and Adnan Ahmad. Copyright: CC-Att-ND-3 (Creative Commons Attribution-NoDerivs 3.0 Unported).*

## 3.9 DISCUSSION QUESTIONS

The following questions are designed to encourage thinking on the chapter and exploring socio-technical cases from the Internet. If you are reading this chapter in a class - either at university or commercial – the questions might be discussed in class first, and then students can choose questions to research in pairs and report back to the next class.

1. Why can technologists not leave the social and ethical questions to non-technologists? Give examples of IT both helping and hurting humanity. What will decide, in the end, whether IT helps or hurts us overall?
2. Compare central vs. distributed networks (Ethernet vs. Polling). Compare the advantages and disadvantages of centralizing vs. distributing control. Is

central control ever better? Now consider social systems. Of the traditional socio-technical principles listed, which ones distribute work-place control? Compare the advantages and disadvantages of centralizing vs. distributing control in a social system. Compare governance by a dictator tyrant, a benevolent dictator and a democracy. Which type are most online communities? How might that change?

3. Originally, socio-technical ideas applied social requirements to work-place management. How has it evolved today? Why is it important to apply social requirements to IT design? Give examples.
4. Illustrate system designs that apply: mechanical requirements to hardware[20]; informational requirements to hardware[21]; informational requirements to software; personal requirements to hardware[22]; personal requirements to software; personal requirements to people; community requirements to hardware; community requirements to software; community requirements to people; community requirements to communities. Give an example in each case. Why not design software to mechanical requirements?
5. Is technology the sole basis of modern prosperity? If people suddenly stopped trusting each other, would wealth continue? Use the 2009 credit meltdown to illustrate your answer. Can technology solve social problems like mistrust? How can social problems be solved? Can technology help?
6. Should an online system gather all the data it can during registration? Give two good reasons not to gather or store non-essential personal data. Evaluate three online registration examples.
7. Spam demonstrates a socio-technical gap, between what people want and what technology does. How do users respond to it? In the "spam wars", who wins? Who loses? Give three other examples of a socio-technical gap. Of the twenty most popular third-party software downloads, which relate

20. As specified by engineering.

21. As specified by computer science.

22. As specified by psychology.

to a socio-technical gap?

8. What is a legitimate government? What is a legitimate interaction? How do people react to an illegitimate government or interaction? How are legitimacy requirements met in physical society? Why will this not work online? What will work?
9. What is the problem with "social engineering"? How about "mental engineering" (brainwashing)? Why do these terms have negative connotations? Is education brainwashing? Why not? Explain the implications for STS design.
10. For a well known STS, explain how it supports, or not, the eight proposed aspects of community performance, with screenshot examples. If it does not support an aspect, suggest why. How could it?
11. Can we own something but still let others use it? Can a community be both free and ordered? Can people compete and cooperate at the same time? Give physical and online examples. How are such tensions resolved? How does democracy reconcile freedom and order? Give examples in politics, business and online.
12. What is community openness for a nation? For an organization? For a club or group? Online? Why are organizations that promote people based on merit more open? Illustrate technology support for merit-based promotion in an online community.
13. Is a person sending money to a personal friend online entitled to keep it private? What if the sender is a public servant? What if it is public money? Is a person receiving money from a personal friend online entitled to keep it private? What if the receiver is a public servant?
14. What is communication? What is meaning? What is communication performance? How can media richness be classified? Is a message itself rich? Does video always convey more meaning than text? How can a lean medium like Twitter deliver more communication performance? Give online

and offline examples.

15. What affects communication performance besides richness? How is it classified? Is it a message property? How does it communicate more? Give online/offline examples.
16. If media richness and linkage both increase communication power, why not have both? Describe a physical world situation that does this. What is the main restriction? Can online media do this? What is, currently, the main contribution of computing to communication power? Give examples.
17. What communication media type best suits these goals: telling everyone about your new product; relating to friends; getting group agreement? Give online and offline examples. For each goal, what media richness, linkage and anonymity do you recommend. You lead an agile programming team spread across the world: what communication technology would you use?
18. State differences between the following media pairs: email and chat; instant messaging and texting; telephone and email; chat and face-to-face conversation; podcast and video; DVD and TV movie; wiki and bulletin board. Do another pair of your choice.
19. How can a physical message convey content, state and position semantic streams? Give examples of communications that convey: content and state; content and position; state and position; and content, state and position. Give examples of people trying to add an ignored semantic stream to technical communication, e.g. people introducing sender state data into lean text media like email.
20. Can a physical message generate many information streams? Can an information stream generate many semantic streams? Give examples. Does the same apply online? Use the way in which astronomical or earthquake data is shared online to illustrate your answer.
21. You want to buy a new cell-phone and an expert web review suggests model A based on factors like cost and performance. Your friend recommends B,

uses it every day, and finds it great. On an online customer feedback site, some people report problems with A and B, but most users of C like it. What are the pluses and minuses of each influence? Which advice would you probably follow? Ask three friends what they would do.

22. What is the best linkage to send a message to many others online? What is the best linkage to make or keep friends online? What is the best linkage to keep up with community trends online? List the advantages and disadvantages of each style. How can technology support each of the above?
23. Explain why reputation ratings, social bookmarks and tagging are all matrix communication. In each case, describe the senders, the message, and the receivers. What is the social goal of matrix communication? How exactly does technology support it?
24. Give three online leaders searched by Google or followed on Twitter. Why do people follow leaders? How can leaders get people to follow them? How does technology help? If the people are already following a set of leaders, how can new leaders arise? If people are currently following a set of ideas, how can new ideas arise? Describe the innovation adoption model. Explain how it applies to “viral” videos?

## Your notes and thoughts on chapter 3

Record your notes and thoughts on this chapter. If you want to share these thoughts with others online, go to the bottom of the page at: http://www.interaction-design.org/books/the_social_design_of_technical_systems_2nd_ed/socio-technical_design.html

**NOTES:**

# CHAPTER 4

# Polite Computing

"*Politeness makes a community a nice place to be*"

This chapter analyzes politeness as a socio-technical requirement

## 4.1 CAN MACHINES BE POLITE?

Software, with its ability to make choices, has crossed the border between inert machine and social participant, as the term human-computer interaction (HCI) implies. Computers today are no longer just tools that respond passively to directions but social agents that are online participants in their own right. Miller notes that if I hit my thumb with a hammer I blame myself, not the hammer, but people often blame equally mechanical programs for user initiated errors. (Miller, 2004, p. 31).

Computer programs are just as mechanical as cars, as each state defines the next, yet programs now ask questions, suggest actions and give advice. Software mediating a social interaction, like email, is like a social facilitator. As computing evolves, people increasingly see programs as active collaborators rather than passive media. These new social roles, of agent, assistant or facilitator, imply a new requirement – to be polite.

To treat machines as people seems foolish, like talking to an empty car, but words addressed to cars on the road are actually to their drivers. Cars are machines but the drivers are people. Likewise, a program is mechanical but people "drive" the programs we interact with. It is not surprising that people show significantly more relational behaviour when the other party in computer mediated communication is clearly human than when it is not (Shectman & Horowitz, 2003). Studies find that people do not treat computers as people outside the mediation context (Goldstein, Alsio, & Werdenhoff, 2002) – just as people do not usually talk to empty cars.

Treating a software installation program as if it were a person is not unreasonable if the program has a human source. Social questions like: "Do I trust you?" and "What is your attitude to me?" apply. If computers have achieved the status of social agents, it is natural for people to treat them socially.

A *social agent* is a social entity that represents another social entity in a social interaction. The other social entity can be a person or group, e.g. installation programs interact with customers on behalf of a company (a social entity). The interaction is social even if the agent, a program, is not, because an install is a social contract. Software is not social in itself, but to mediate a social interaction it must operate socially. If a software agent works for the party it interacts with, it is an *assistant*, both working for and to the same user. A human-computer interaction with an assistant also requires politeness.

If software mediates social interactions it should be designed accordingly. No company would send a socially ignorant person to talk to important clients,

yet they send software that interrupts, overwrites, nags, steals, hijacks and in general annoys and offends users (Cooper, 1999). Polite computing is the design of socially competent software.

## 4.2 SELFISH SOFTWARE

*Selfish software* acts as if it were the only application on your computer, just as a selfish person acts as if only he or she exists. It pushes itself forward at every opportunity, loading at start-up and running continuously in the background. It feels free to interrupt you at any time to demand things or announce what it is doing, e.g. after I (the first author) had installed new modem software, it then loaded itself on every start-up and regularly interrupted me with a modal[1] window saying it was going online to check for updates to itself. It never found any, even after weeks. Finally, after yet another pointless "Searching for upgrades" message I could not avoid, I uninstalled it. As in The Apprentice TV show, assistants that are no help are fired! Selfish apps going online to download upgrades without asking mean tourists with a smartphone can find themselves with high data roaming bills for downloads they did not want or need.

Users uninstalling impolite software is a new type of computing error – *a social error*. A computer system that gets into an infinite loop that hangs it has a software error; if a user cannot operate the computer system, it is a usability error; and if software offends and drives users away, it is a social error. In usability errors, people want to use the system but do not know how to, but in social errors they understand it all too well and choose to avoid it.

Socio-technical systems cannot afford social errors because in order to work, they need people to participate. In practice, a web site that no-one visits is as much a failure as one that crashes. Whether a system fails because the computer

1. In HCI, a modal message requires users to respond, i.e. it takes and holds the current mouse or keyboard focus.

cannot run it, the user does not know how to run it, or the user does not want to run it, does not matter. The end effect is the same - the *application does not run*.

For example, my new 2006 computer came with McAfee Spamkiller, which by design overwrote my Outlook Express mail server account name and password with its own values when activated. I then no longer received email, as the mail server account details were wrong. After discovering this, I retyped in the correct values to fix the problem and got my mail again. However the next time I rebooted the computer, McAfee rewrote over my mail account details again. I called the McAfee help person, who explained that Spamkiller was protecting me by taking control, and routing all my email through itself. To get my mail I had to go into McAfee and tell it my specific email account details, but when I did this it still did not work. I was now at war with this software, which:

1. Overwrote the email account details I had typed in.
2. Did nothing when the email didn't work.

The software "took charge" but didn't know what it was doing. Whenever Outlook started, it forced me to watch it do a slow foreground modal check for email spam, but in two weeks of use it never found any! Not wanting to be held hostages by a computer program, I again uninstalled it as selfish software.

## 4.3 POLITE SOFTWARE

Polite computing addresses the requirement for a social entities to work together. The Oxford English Dictionary (http://dictionary.oed.com) defines politeness as:

"*... behaviour that is respectful or considerate to others*".

So software that respects and considers users is polite, as distinct from software usefulness or usability, where usefulness addresses functionality and usability how easy it is to use. Usefulness is what the computer does and usability is how

users get it to do it. Polite computing in contrast is about social interactions, not computer power or cognitive ease. So software can be easy to use but rude, or polite but hard to use. While usability reduces training and documentation costs, politeness lets a software agent socially interact with success. Both usability and politeness fall under the rubric of human-centred design.

Polite computing is about designing software to be polite, not making people polite. People are socialized by society, but rude, inconsiderate or selfish software is a widespread problem because it is a software design "blind spot" (Cooper, 1999). Most software is socially blind, except for socio-technical systems like Wikipedia, Facebook and E-Bay. This chapter outlines a vision of polite computing for the next generation of social software.

## 4.4 SOCIAL PERFORMANCE

If politeness is considering the other in a social interaction, the predicted effect is more pleasant interaction. In general, politeness makes a society a nicer place to be, whether online or offline. It contributes to computing by:

1. Increasing legitimate interactions.
2. Reducing anti-social interactions.
3. Increasing synergy.
4. Increasing software use.

Programmers can fake politeness, as people do in the physical world, but when people *behave* politely, cognitive dissonance theory finds that people *feel* polite (Festinger, 1957). So if programmers design for politeness, the overall effect will be positive even although some programmers may be faking it.

Over thousands of years, as physical society became "civilized", it created more prosperity. Today, for the first time in human history, many countries are

producing more food than their people can eat, as their obesity epidemics testify. The bloody history of humanity has been a social evolution from *zero-sum* (win-lose) interactions like war to *non-zero-sum* (win-win) interactions like trade (Wright, 2001), with productivity the prize. Scientific research illustrates this: scientists freely giving their hard earned knowledge away seems foolish, but when a critical mass do it the results are astounding.

Social synergy is people in a community giving to each other to get more than is possible by selfish activity; e.g. Open Source Software (OSS) products like Linux now compete with commercial products like Office. The mathematics of synergy reflect its social interaction origin: competence gains increase linearly with group size but synergy gains increase geometrically, as they depend on the number of interactions. In the World Wide Web, we each only sow a small part of it but we reap from it the world's knowledge. Without polite computing, however, synergy is not possible.

A study of reactions to a computerized Chinese word-guessing game found that when the software apologized after a wrong answer by saying "*We are sorry that the clues were not helpful to you*," the game was rated more enjoyable than when the computer simply said "*This is not correct*" (Tzeng, 2004). In general, politeness improves online social interactions and so increases them. Politeness is what makes a social environment a nice place to be. Businesses who wonder why more people don't shop online should ask whether the World Wide Web is a place people want to be? If it is full of spam, spyware, viruses, hackers, pop-ups, nagware, identity theft, scams, spoofs, offensive pornography and worms then it will be a place to avoid. As software becomes more polite, people will use it more and avoid it less.

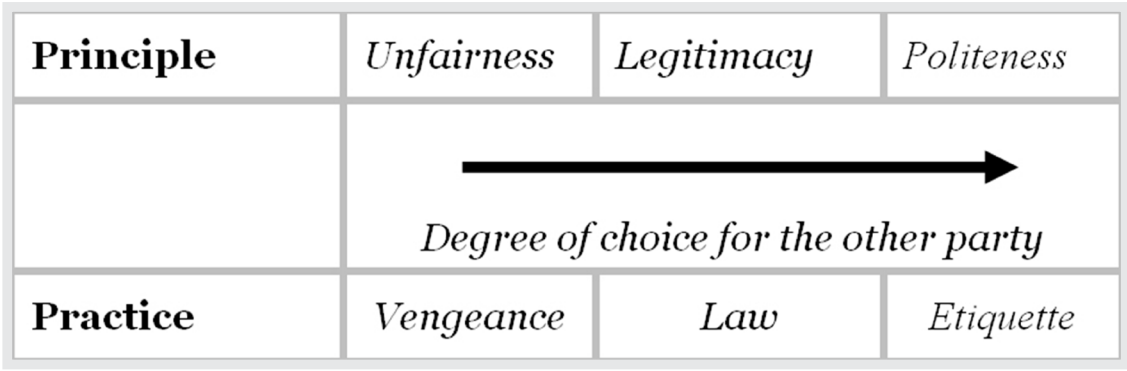



FIGURE 4.1: Social interactions by degree of choice given.

*Copyright status: Unknown (pending investigation). See section "Exceptions" in the copyright terms below.*

## 4.5 THE LEGITIMACY BASELINE

Legitimate interactions, defined as those that are both fair and in the common good, are the basis of civilized prosperity (Whitworth & deMoor, 2003) and legitimacy is a core demand of any prosperous and enduring community (Fukuyama, 1992). Societies that allow corruption and win-lose conflicts are among the poorest in the world (Transparency-International, 2001). Legitimate interactions offer fair choices to parties involved while anti-social crimes like theft or murder give the victim little or no choice. Figure 4.1 categorizes anti-social, social and polite interactions by the degree of choice the other party has.

So polite acts are *more than fair,* i.e. more than the law. To follow the law is not politeness because it is required. One does not thank a driver who stops at a red light, but one thanks the driver who stops to let you into a line of traffic. Laws specify what citizens *should* do but politeness is what they *could* do. Politeness involves *offering more choices in an interaction than the law requires,* so it begins where fixed laws end. If criminal acts fall below the law, then polite acts rise above it (Figure 4.2). Politeness increases social health as criminality poisons it.

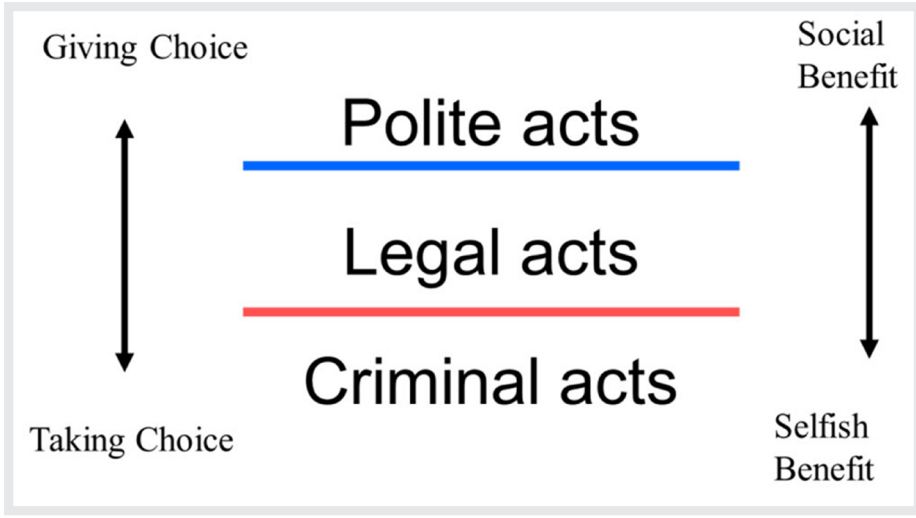



FIGURE 4.2: Politeness is doing more than the law.

*Courtesy of Brian Whitworth and Adnan Ahmad. Copyright: CC-Att-ND-3 (Creative Commons Attribution-NoDerivs 3.0 Unported).*

## 4.6 SECURITY

In computing design, politeness takes a back seat to security, but upgrading security every time an attack exploits another loophole is a never-ending cycle. Why not develop strategies to reduce the motivation to attack the community (Rose, Khoo, & Straub, 1999)? Polite computing does just this, as it reduces a common source of attacks — anger against a system that allows those in power to prey upon the weak (Power, 2000). Hacking is often revenge against a person, a company or the capitalist society in general (Forester & Morrison, 1994).

Politeness openly denies the view that "everyone takes what they can so I can too". A polite system can make those who are neutral polite and those who are against society neutral. Politeness and security are thus two sides of the same coin of social health. By analogy, a gardener defends his or her crops from weeds, but does not wait until every weed dies before fertilizing. If politeness grows social health, it complements rather than competes with security.

## 4.7 ETIQUETTE

Some define politeness as "being nice" to the other party (Nass, 2004). So if someone says "I'm a good teacher; what do you think?" then polite people respond "You're great", even if they do not agree. Agreeing with another's self-praise is called one of the "fundamental rules of politeness" (Nass, 2004, p36). Yet this is illogical, as one can be agreeably impolite and politely disagreeable. One can politely refuse, beg to differ, respectfully object and humbly criticize, i.e. disagree politely. Conversely one can give to charity impolitely, i.e. be kind but rude. Being polite is thus different from being nice, as parents who are kind to their child may not agree to let it choose its own bedtime.

To apply politeness to computer programming, we must define it in information terms. If it is considering others, then as different societies consider differently, what is polite in one culture is rude in another. If there is no universal polite behaviour, there seems no basis to apply politeness to the logic of programming.

Yet while different countries have different laws, the goal of fairness behind the law can be attributed to every society (Rawls, 2001). Likewise, different cultures could have different etiquettes but a common goal of politeness. In Figure 4.1, the physical practices of vengeance, law and etiquette derive from human concepts of unfairness, legitimacy and politeness.

So while societies implement different forms of vengeance, law and etiquette, the aims of avoiding unfairness, enabling legitimacy and encouraging politeness do not change. So politeness is the spirit behind etiquettes, as legitimacy is the spirit behind laws. Politeness lets us *generate* new etiquettes for new cases, as legitimacy lets us generate new laws. Etiquettes and laws are the information level reflections of the human level concepts of politeness and legitimacy.

If politeness can take different forms in different societies, to ask which implementation applies online is to ask the wrong question. This is like asking a country which other country's laws they want to adopt, when laws are generally

home grown for each community. The real question is how to "reinvent" politeness online, whether for chat, wiki, email or other groupware. Just as different physical societies develop different local etiquettes and laws, so online communities will develop their own ethics and practices, with software playing a critical support role. While different applications may need different politeness implementations, we can develop general design "patterns" to specify politeness in information terms (Alexander, 1964).

## 4.8 AN INFORMATION DEFINITION OF SOFTWARE POLITENESS

If the person being considered knows what is "considerate" for them, politeness can be defined abstractly as *the giving of choice to another in a social interaction*. This is then always considerate given only that the other knows what is good for him or her. The latter assumption may not always be true, e.g. in the case of a young baby. In a conversation where the locus of channel control passes back and forth between parties, it is polite to give control to the other party (Whitworth, 2005), e.g. it is impolite to interrupt someone, as that removes their choice to speak, and polite to let them finish talking, as they then choose when to stop. This gives a definition of politeness as:

*"... any unrequired support for situating the locus of choice control of a social interaction with another party to it, given that control is desired, rightful and optional." (Whitworth, 2005, p355)*

*Unrequired* means the choice given is more than required by the law, as a required choice is not politeness. *Optional* means the polite party has the ability to choose, as politeness is voluntary. *Desired by the receiver* means giving choice is only polite if the other wants it, e.g. "After you" is not polite when facing a difficult task. Politeness means giving desired choices, not forcing the locus of control, with its burden of action, upon others. Finally, *rightful* means that consideration

of someone acting illegally is not polite, e.g. to considerately hand a gun to a serial killer about to kill someone is not politeness.

## 4.9 THE REQUIREMENTS

Following previous work (Whitworth, 2005), polite software should:

1. ***Respect the user.*** *Polite software respects user rights, does not act pre-emptively, and does not act on information without the permission of its owner.*
2. ***Be visible.*** *Polite software does not sneak around changing things in secret, but openly declares what it is doing and who it represents.*
3. ***Be understandable.*** *Polite software helps users make informed choices by giving information that is useful and understandable.*
4. ***Remember you.*** *Polite software remembers its past interactions and so carries forward your past choices to future interactions.*
5. ***Respond to you****. Polite software responds to user directions rather than trying to pursue its own agenda.*

Each of these points is now considered in more detail.

### 4.9.1 Respectfulness

Respect includes not taking another's rightful choices. If two parties jointly share a resource, one party's choices can deny the other's; e.g. if I delete a shared file, you can no longer print it. Polite software should not preempt rightful user information choices regarding common resources such as the desktop, registry, hard drive, task bar, file associations, quick launch and other user configurable settings. Pre-emptive acts, like changing a browser home page without asking, act unilaterally on a mutual resource and so are impolite.

Information choice cases are rarely simple; e.g. a purchaser can use software but not edit, copy or distribute it. Such rights can be specified as privileges, in terms of specified information actors, methods, objects and contexts (see Chapter 6). To apply politeness in such cases requires a legitimacy baseline; e.g. a provider has no unilateral right to upgrade software on a computer the user owns (though the Microsoft Windows Vista End User License Agreement (EULA) seems to imply this). Likewise users have no right to alter the product source code unilaterally. In such cases politeness applies; e.g. the software suggests an update and the user agrees, or the user requests an update and the software agrees (for the provider). Similarly while a company that creates a browser owns it, the same logic means users own data they create with the browser, e.g. a cookie. Hence, software cookies require user permission to create, and users can view, edit or delete them.

### 4.9.2 Visibility

Part of a polite greeting in most cultures is to introduce oneself and state one's business. Holding out an open hand, to shake hands, shows that the hand has no weapon, and that nothing is hidden. Conversely, to act secretly behind another's back, to sneak, or to hide one's actions, for any reason, is impolite. Secrecy in an interaction is impolite because the other has no choice regarding things they do not know about. Hiding your identity reduces my choices, as hidden parties are untouchable and unaccountable for their actions. When polite people interact, they declare who they are and what they are doing.

If polite people do this, polite software should do the same. Users should be able to see what is happening on their computer. Yet when Windows Task Manager attributes a cryptic process like CTSysVol.exe to the user, it could be a system critical process or one left over from a long uninstalled product. Lack of

transparency is why after 2-3 years Windows becomes “old”. With every installation of selfish software it puts itself everywhere, fills the taskbar with icons, the desktop with images, the disk with files and the registry with records. In the end, the computer owner has no idea what software is responsible for what files or registry records.

Selfish applications consider themselves important enough to load at start-up and run continuously, in case you need them. Many applications doing this slow down a computer considerably, whether on a mobile phone or a desktop. Taskbar icon growth is just the tip of the iceberg of what is happening to the computer, as some start-ups do not show on the taskbar. Selfish programs put files where they like, so uninstalled applications are not removed cleanly, and over time Windows accretes a “residue” of files and registry records left over from previous installs. Eventually, only reinstalling the entire operating system recovers system performance.

The problem is that the operating system keeps no transparent record of what applications do. An operating system *Source Registry* could link all processes to their social sources, giving contact and other details. *Source* could be a property of every desktop icon, context menu item, taskbar icon, hard drive file or any other resource. If each source creates its own resources, a user could then delete all resources allocated by a source they have uninstalled without concern that they were system critical. Windows messages could also state their source, so that users knew who a message was from. Application transparency would let users decide what to keep and what to drop.

### 4.9.3 Helpfulness

A third politeness property is to help the user by offering understandable choices, as a user cannot properly choose from options they do not understood. Of-

fering options that confuse is inconsiderate and impolite, e.g. a course text web site offers the choices:

1. OneKey Course Compass
2. Content Tour
3. Companion Website
4. Help Downloading
5. Instructor Resource Centre

It is unclear how the "Course Compass" differs from the "Companion Website", and why both seem to exclude "Instructor Resources" and "Help Downloading". Clicking on these choices, as is typical for such sites, leads only to further confusing menu choices. The impolite assumption is that users enjoy clicking links to see where they go. Yet information overload is a serious problem for web users, who have no time for hyperlink merry-go-rounds.

Not to offer choices at all on the grounds that users are too stupid to understand them is also impolite. Installing software can be complex, but so is installing satellite TV technology, and those who install the latter do not just come in and take over. Satellite TV installers know that the user who pays expects to hear his or her choices presented in an understandable way. If not, the user may decide not to have the technology installed.

Complex installations are simplified by *choice dependency analysis,* of how choices are linked, as Linux's installer does. Letting a user choose to install an application the user wants minus a critical system component is not a choice but a trap. Application-critical components are part of the higher choice to install or not; e.g. a user's permission to install an application may imply access to hard drive, registry and start menu, but not to desktop, system tray, favorites or file associations.

### 4.9.4 Remembering

It is not enough for the software to give choices now but forget them later. If previous responses have been forgotten, the user is forced to restate them, which is inconsiderate. Software that actually listens and remembers past user choices is a wonderful thing. Polite people remember previous encounters, but each time Explorer opens it fills *its* preferred directory with files I do not want to see, and then asks me which directory *I* want, which is *never* the one displayed. Each time I tell it, and each time Explorer acts as if it were the first time I had used it. Yet I am the only person it has ever known. Why can't it *remember* the last time and return me there? The answer is that it is impolite by design.

Such "amnesia" is a trademark of impolite software. Any document editing software could automatically open a user's last open document, and put the cursor where they left off, or at least give that option (Raskin, 2000, p.31). The user logic is simple: If I close the file I am finished, but if I just exit without closing the document, then put me back where I was last time. It is amazing that most software cannot even remember the last user interaction. Even within an application, like email, if one moves from inbox to outbox and back, it "forgets" the original inbox message, so one must scroll back to it; cf. browser tabs that remember user web page position.

### 4.9.5 Responsiveness

Current "intelligent" software tries to predict user wants but cannot itself take correction, e.g. Word's auto-correct function changes i = 1 to I = 1, but if you change it back the software ignores your act. This is software that is clever enough to give corrections but not clever enough to take correction itself. However, responsive means responding to the user's direction, not ignoring it.

A classic example of non-responsiveness was Mr. Clippy, Office 97's paper clip assistant (Figure 4.3).

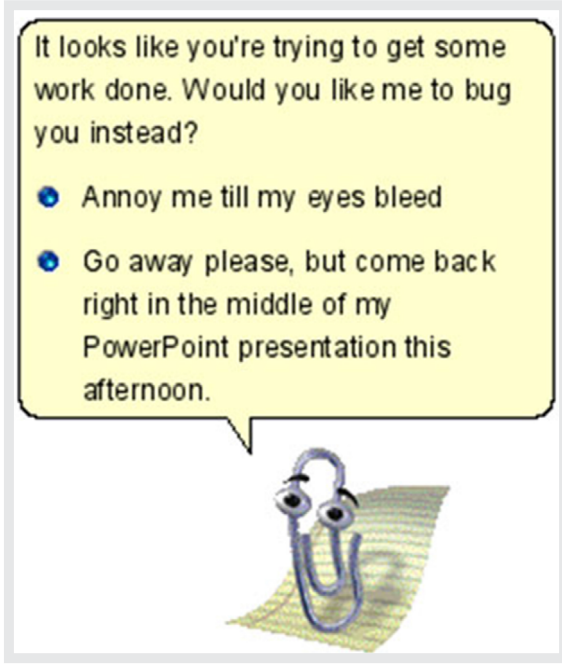



FIGURE 4.3: Mr. Clippy takes charge.

*Copyright © Microsoft Corp.. All Rights Reserved. Used without permission under the Fair Use Doctrine (as permission could not be obtained). See the "Exceptions" section (and subsection "allRightsReserved-Used-WithoutPermission") on the page copyright notice.*

Searching the Internet for Mr. Clippy" gives comments like "*Die, Clippy, Die*!" (Gauze, 2003), yet its Microsoft designer still wondered:

"*If you think the Assistant idea was bad, why exactly*?"

The answer is as one user noted:

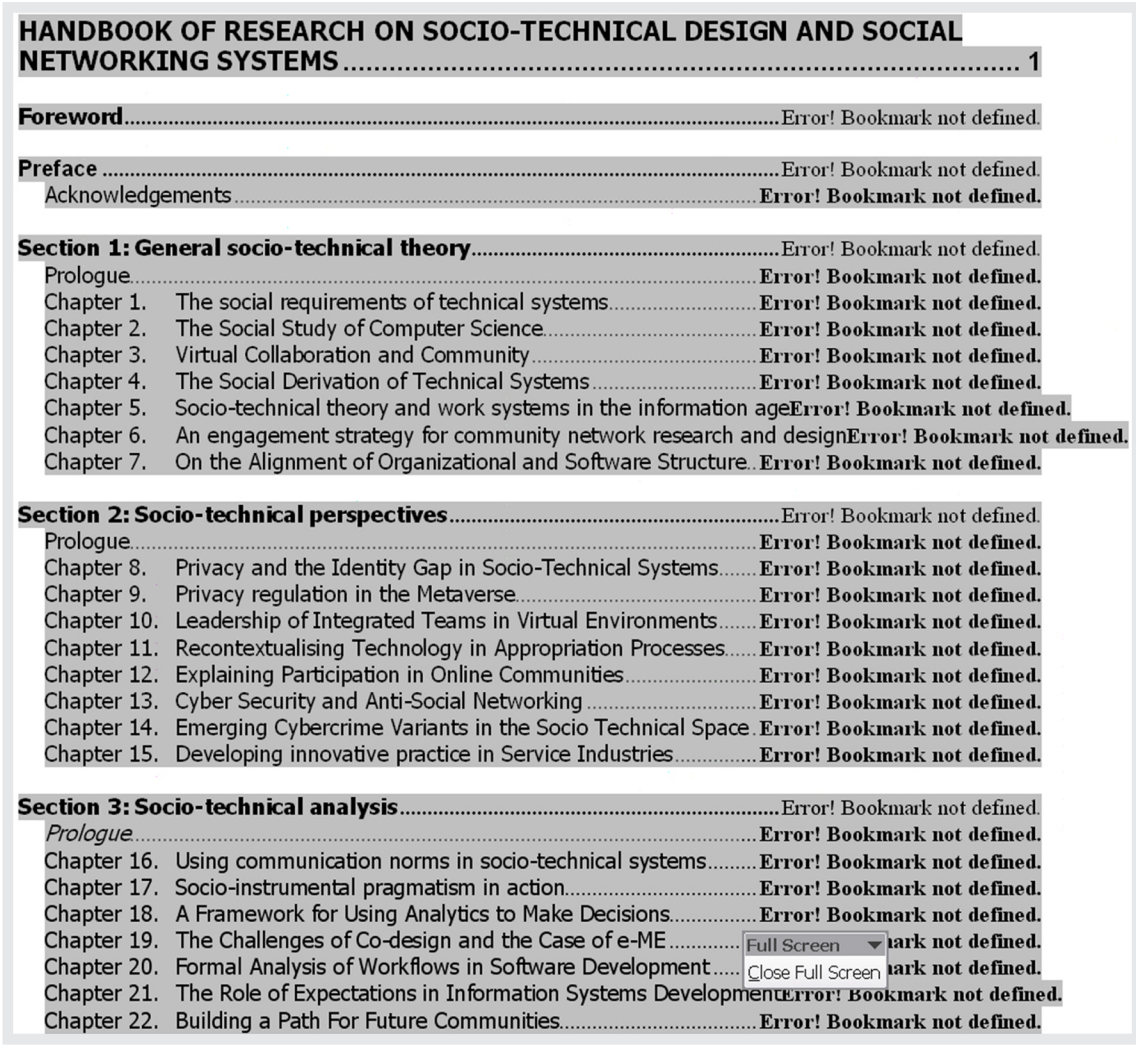
"*It wouldn't go away when you wanted it to. It interrupted rudely and broke your train of thought*. (Pratley, 2004).

To interrupt inappropriately disturbs the user's train of thought. For complex work like programming, even short interruptions cause a mental "core dump", as the user drops one thing to attend to another. The interruption effect is then not just the interruption time, but also the recovery time (Jenkins, 2006); e.g. if it takes three minutes to refocus after an interruption, a one second interruption every three minutes can reduce productivity to zero.

Mr Clippy was impolite, and in XP was replaced by tags smart enough to know their place. In contrast to Mr Clippy, tag clouds and reputation systems illustrate software that reflects rather than directs online users.

Selfish software, like a spoilt child, repeatedly interrupts, e.g. a Windows Update that advises the user when it starts, as it progresses and when it finishes. Such modal windows interrupt users, seize the cursor and lose current typing. Since each time the update only needs the user to press OK, it is like being repeatedly interrupted to pat a self-absorbed kiddie on the head. The lesson of Mr. Clippy, that software serves the user not the other way around, still needs to be learned.

It is hard for selfish software to keep appropriately quiet; e.g. Word can generate a table of contents from a document's headings. However if one sends just the first chapter of a book to someone, with the book's table of contents (to show its scope), every table of contents heading line without a page number loudly declares: "ERROR! BOOKMARK NOT DEFINED", which of course completely spoils the sample document impression (Figure 4.4). Even worse, this effect is not apparent until the document is received. Why could the software not just quietly put a blank instead of a page number? Why announce its needs so rudely? What counts is not what the software needs but what the user needs, and in this case the user needs the software to be quiet.



**HANDBOOK OF RESEARCH ON SOCIO-TECHNICAL DESIGN AND SOCIAL NETWORKING SYSTEMS ..... 1**



FIGURE 4.4: A table of contents as emailed to a colleague (Word).

*Copyright status: Unknown (pending investigation). See section "Exceptions" in the copyright terms below.*

## 4.10 IMPOLITE COMPUTING

Impolite computing has a long history; e.g. spam fills inboxes with messages users do not want (Whitworth & Whitworth, 2004). Spam is impolite because it takes choice away from email users. Pop-up windows are also impolite, as they "hijack"

the user's cursor or point of focus. They also take away user choices, so many browsers now prevent pop-ups. Impolite computer programs:

1. *Use your computer's services*. Software can use your hard drive to store information cookies, or your phone service to download data without asking.
2. *Change your computer settings*. Software can change browser home page, email preferences or file associations.
3. *Spy on what you do online*. Spyware, stealthware and software back doors can gather information from your computer without your knowledge, or record your mouse clicks as you surf the web, or even worse, give your private information to others.

For example, Microsoft's Windows XP Media Player, was reported to quietly record the DVDs it played and use the user's computer's connection to "phone home", i.e. send data back to Microsoft (Technology threats to Privacy, 2002). Such problems differ from security threats, where hackers or viruses break in to damage information. This problem concerns those we invite into our information home, not those who break in.

A similar concern is "software bundling", where users choose to install one product but are forced to get many:

> "*When we downloaded the beta version of Triton [AOL's latest instant messenger software], we also got AOL Explorer – an Internet Explorer shell that opens full screen, to AOL's AIM Today home page when you launch the IM client – as well as Plaxo Helper, an application that ties in with the Plaxo social-networking service. Triton also installed two programs that ran silently in the background even after we quit AIM and AOL Explorer*"
>
> *-- Larkin, 2005*

Yahoo's "typical" installation of their IM also used to download their Search Toolbar, anti-spyware and anti-pop-up software, desktop and system tray shortcuts, as well as Yahoo Extras, which inserted Yahoo links on your browser, altered the user's home page and made auto-search functions point to Yahoo by default. Even Yahoo employee Jeremy Zawodny disliked this:

"*I don't know which company started using this tactic, but it is becoming the standard procedure for lots of software out there. And it sucks. Leave my settings, preferences and desktop alone*"

*-- http://jeremy.zawodny.com/blog/archives/005121.html*

Even today, many downloads require the user to opt-out rather than opt-in to offers. One must carefully check the checkboxes, lest one says something like: "Please change all my preferences" or "Please send my endless spam on your products".

A similar scheme is to use security updates to install new products, e.g.:

"*Microsoft used the January 2007 security update to induce users to try Internet Explorer 7.0 whether they wanted to or not. But after discovering they had been involuntarily upgraded to the new browser, they next found that application incompatibility effectively cut them off from the Internet*"

*-- Pallatto, 2007*

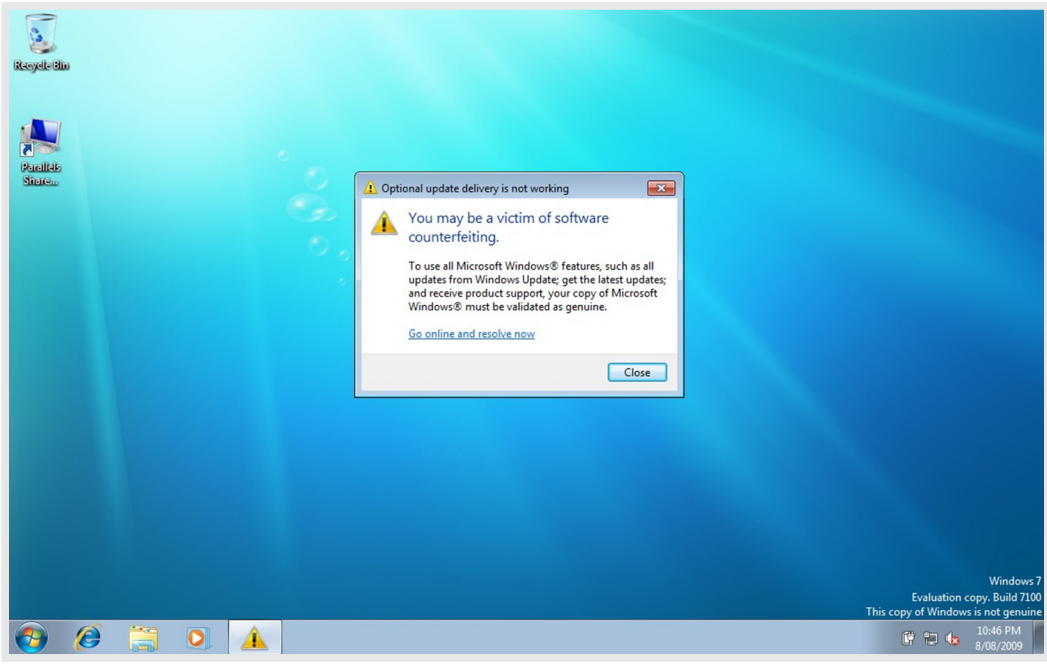



FIGURE 4.5: Windows genuine advantage nagware.

*Copyright status: Unknown (pending investigation). See section "Exceptions" in the copyright terms below.*

After installing Windows security update KB971033, many Windows 7 owners got a nag screen that their copy of Windows was not genuine, even though it was (Figure 4.5). Clicking the screen gave an option to purchase Windows (again). The update was silently installed, i.e. even with a *Never check for updates* setting, Windows did check online and installed it anyway. Despite the legitimacy principle that my PC belongs to me, Windows users found that Microsoft can unilaterally validate, alter or even shut down their software, by its End User License Agreement (EULA). The effect was to make legitimate users hack Windows with tools like RemoveWAT, suspect and delay future Windows updates, avoid Microsoft web sites in case they secretly modify their PC and to investigate more trustworthy options like Linux. Yet an investigation found that while 22% of computers failed the test, less than 0.5% had pirated software[2]. Disrespecting

2. See https://en.wikipedia.org/wiki/Windows_Genuine_Advantage

and annoying honest customers while trying to catch thieves has never been a good business policy.

Security cannot defend against people one invites in, especially if the offender is the security system itself. However, in a connected society social influence can be very powerful. In physical society the withering looks given to the impolite are not toothless, as what others think of you affects how they behave towards you. In traditional societies banishment was often considered worse than a death sentence. An online company with a reputation for riding roughshod over user rights may find this is not good for business.

### 4.10.1 Blameware

A fascinating psychological study could compare computer messages when things are going well to times when they are not. While computers seem delighted to be in charge in good times, when they go wrong software seems universally to agree that *you* have an error, not *we* have an error. Brusque and incomprehensible error messages like the "*HTTP 404 – File not Found*" suggest that you need to fix the problem you have clearly created. Although software itself often causes errors, some software designers recognize little obligation to give to users information the software has in a useful form, let alone suggest solutions. To "take the praise and pass the blame" is not polite computing. Polite software sees *all* interactions as involving "we". Indeed studies of users in human-computer tutorials show that users respond better to "Let's click the Enter button" than to "Click the Enter button" (Mayer et al., 2006). When there is a problem, software should try to help the user, not disown them.

### 4.10.2 Pryware

Pryware is software that asks for information it does not need for any reasonable purpose; e.g. Figure 4.6 shows a special interest site which wants people to regis-

ter but for no obvious reason asks for their work phone number and job title. Some sites are even more intrusive, wanting to know your institution, work address and Skype telephone. Why? If such fields are mandatory rather than optional, people choose not to register, being generally unwilling to divulge data like home phone, cell phone and home address (Foreman & Whitworth, 2005). Equally, they may spam an address like "123 Mystreet, Hometown, zip code 246". Polite software does not pry, because intruding on another's privacy is offensive.

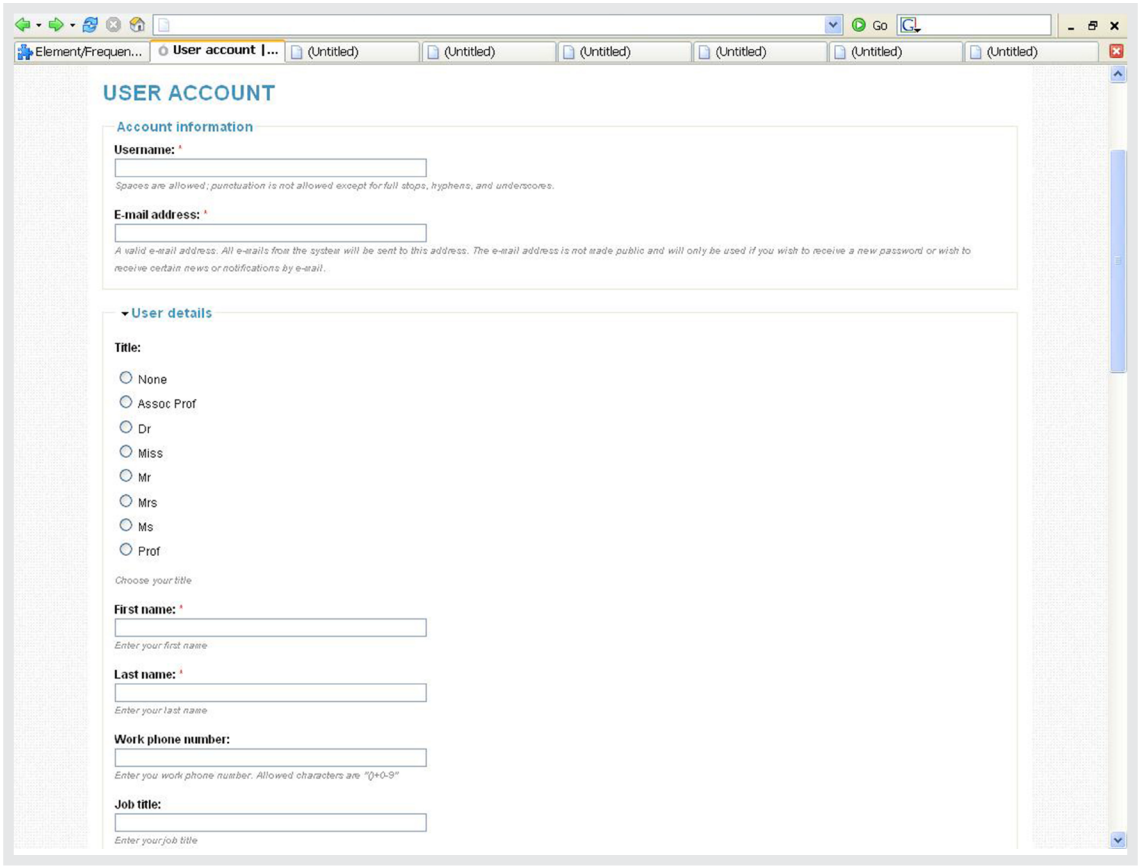



FIGURE 4.6: Pryware asks for unneeded details.

*Courtesy of Brian Whitworth and Adnan Ahmad. Copyright: CC-Att-ND-3 (Creative Commons Attribution-NoDerivs 3.0 Unported).*

### 4.10.3 Nagware

If the same question is asked over and over, for the same reply, this is to pester or *nag*, like the "Are we there yet?" of children on a car trip. It forces the other party to give again and again the same choice reply. Many users did not update to Windows Vista because of its reputation as nagware, that asked too many questions. Polite people do not but a lot of software does, e.g. when reviewing email offline in Windows XP, actions like using Explorer triggered a "Do you want to connect?" request every few minutes. No matter how often one said "No!" it kept asking, because it had no memory of its own past. Yet past software has already solved this problem, e.g. uploading a batch of files creates a series of "Overwrite Y/N?" questions, which would force the user to reply repeatedly "Yes", but there is a "Yes to All" *meta-choice* that remembers for the choice set. *Such choices about choices* (meta-choices) are polite. A general meta-choice console (GMCC) would give users a common place to see or set meta-choices (Whitworth, 2005).

### 4.10.4 Strikeware

Strikeware is software that executes a pre-emptive strike on user resources. An example is a zip extract product that, without asking, put all the files it extracted as icons on the desktop! Such software tends to be used only once. Installation programs are notorious for pre-emptive acts; e.g. the Real-One Player that added desktop icons and browser links, installed itself in the system tray and commandeered all video and sound file associations. Customers resent such invasions, which while not illegal are impolite. An installation program changing your PC settings is like furniture deliverers rearranging your house because they happen to be in it. Software upgrades continue the tradition; e.g. Internet Explorer upgrades that make MSN your browser home page without asking. Polite software does not do this.

### 4.10.5 Forgetware

Selfish software collects endless data on users but is oblivious to how *the user* sees it. Like Peter Sellers in the film "Being There", selfish software likes to watch but cannot itself relate to others. Someone should explain to programmers that spying on a user is not a relationship. Mr. Clippy watched *the user's* acts on the document, but could not see *his* interaction with the user, so was oblivious to the rejection and scorn it evoked. Most software today is less aware of its users than an airport toilet. Software will make remembering user interactions its business when it considers the user, not itself.

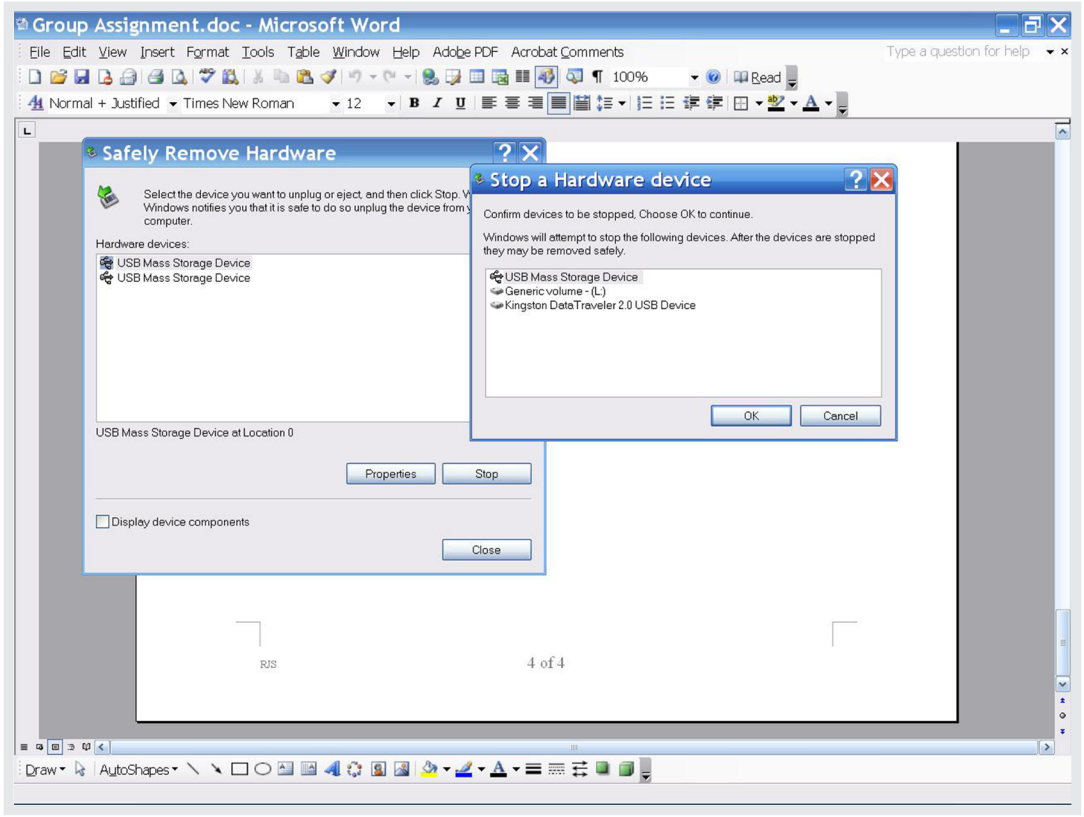



FIGURE 4.7: Windows XP eject USB interface.

Working with forgetware makes the simplest interaction hard; e.g. Figure 4.7 shows the hoops users had to jump through to get a USB flash drive back from Windows XP. Imagine a human assistant that worked like this:

*User*: Hi Bill, please read this book for me (*puts book in Bill's left hand*).
*Assistant*: Bill has found a book in his hand!
*User*: OK, I want the book back now, Bill.
*Assistant*: Should I safely remove the book in my left or right hand?
*User*: The book in your left hand, Bill. The one I just gave you.
*Assistant*: Please confirm the book you mean by pointing to it, or the hand, or both, and say "OK". Then I will try to give it to you.
*User*: (*points to left hand*) OK!
*Assistant*: (*left hand tries to give book but the right hand is holding it too*) I can't give the book to you right now; try again later.
*User*: Give me the book back!
*Assistant*: Do you want me to safely remove the book in my left hand or my right hand ...
*User*: (*grabs book and walks away*.)

A human helper who acted like this would be insolent. If the Windows software can discover a USB is busy *after* the user selects it, why not check it *before*? Is its time more important than that of the user? Most users of course just check the USB drive light and if it is not flashing pull it out, avoiding all the above.

In a similar vein, a computer voice reports my phone messages like this:

"*There are five new messages. The first message received at 12.15pm on Wednesday the 14th of November is "<hang-up click>" To save this message press 1, to forward it press 3, to reply to it press 5, ..., to delete it press 76.*"

Note: "76" was the actual delete message code number, even though it is probably the most used option, especially for hang up calls. Again, imagine a hu-

man secretary who felt the need to report every detail of a call before telling you that the caller just hung up.

## 4.11 THE WIZARD'S APPRENTICE

The problem presented here, of computers taking over what they do not understand, is embodied in the story of the wizard who left his apprentice in charge of his laboratory. Thinking he knew what he was doing, the apprentice started to cast spells, but they soon got out of hand and only the wizard's return prevented total disaster. Smart software, like that apprentice, often acts on its own, causing things to get out of control, and leaving the user to pick up the pieces.

As software evolves it can get things wrong more. For example, Endnote software manages citations in documents like this one by embedding links to a reference database. Endnote Version X ran itself whenever I opened the document, and if it found a problem took the focus from whatever I was doing to let me know right away (Figure 4.8). It used square brackets for citations, so assumed any square brackets in the document were for it, as little children assume any words spoken are to them. After I had told it every time to ignore some [ ] brackets I had in the document, it then closed, dropping the cursor at whatever point it was and leaving me to find my own way back to where I was when it had interrupted.

I could not turn this activation off, so editing my document on the computer at work meant that Endnote could not find its citation database. It complained, then handled the problem by clearing all the reference generating embedded links in the document! If I had carried on without noticing, all the citations would have had to be later re-entered. Yet if you *wanted* to clear the Endnote links, as when publishing the document, it had to be done manually.

Selfish software comes with everything but an off button. It takes no advice and preemptively changes your data without asking. I fixed the problem by uninstalling Endnote and installing Zotero.

Cooper compares smart software to a dancing bear: we clap not because it dances well, but because we are amazed it can dance at all (Cooper, 1999). We need to stop clapping.

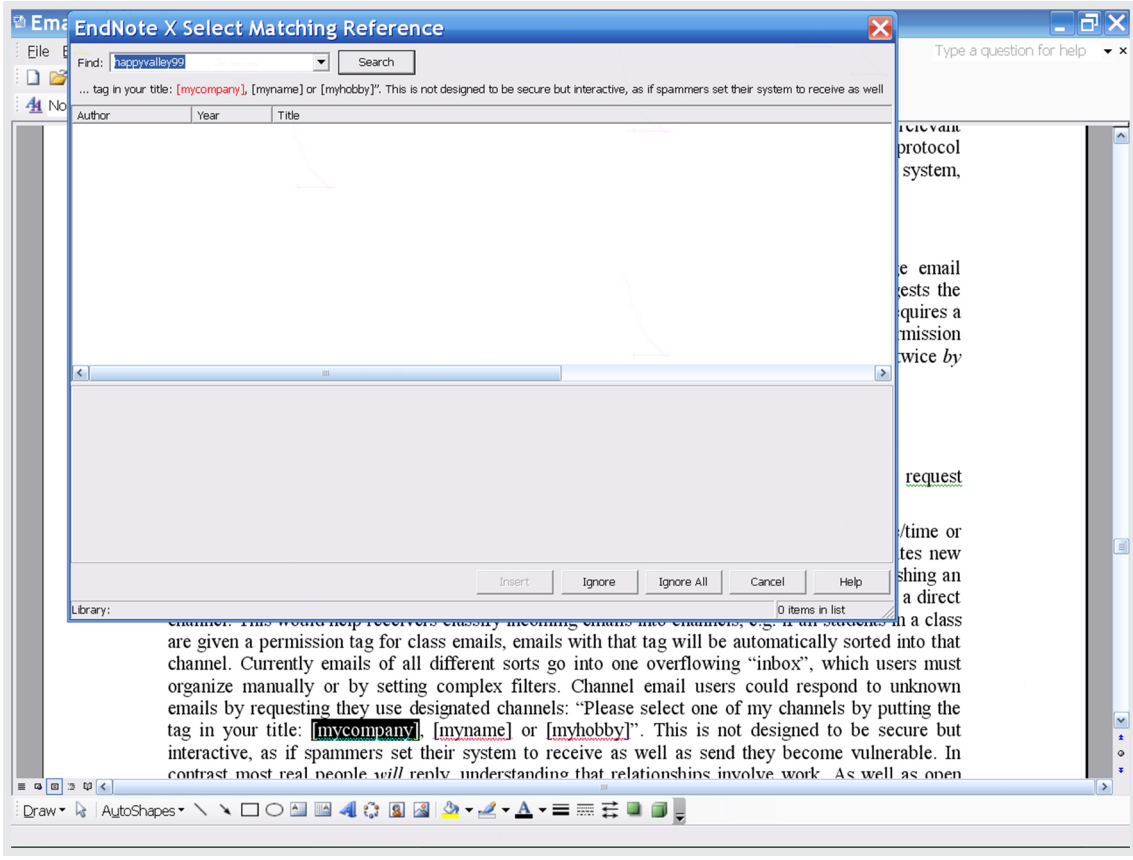



FIGURE 4.8: Endnote X takes charge.

*Copyright status: Unknown (pending investigation). See section "Exceptions" in the copyright terms below.*

For many reasons, people should control computers, not the reverse. *Firstly*, computers manage vast amounts of data with ease but handle context changes poorly (Whitworth, 2008), so smart computing invariably needs a human minder. *Secondly*, computers are not accountable for what they do, as they have no "self" to bear

any loss. If people are accountable for what computers do, they need control over computer choices. *Thirdly*, people will always resist computer domination. Software designers who underestimate the importance of user choice invite grass-roots rebellion. An Internet movement against software arrogance is not inconceivable.

Today many users are at war with their software: removing things they did not want added, resetting changes they did not want changed, closing windows they did not want opened and blocking e-mails they did not want to receive. User weapons in this war, such as third party blockers, cleaners, filters and tweakers, are the most frequent Internet download site accesses. Their main aim is to put users back in charge of the computing estate they paid for. If software declares war on users, it will not win. If the Internet becomes a battlefield, no-one will go there. The solution is to give choices, not take them, i.e. polite computing.

The future of computing lies not in it becoming so clever that people are obsolete but in a human-computer combination that performs better than people or computers alone. The runaway IT successes of the last decade (cell-phones, Internet, e-mail, chat, bulletin boards etc.) all support people rather than supplant them. As computers develop this co-participant role, politeness is a critical success factor.

## 4.12 THE POLITE SOLUTION

For modern software, often downloaded from the web, users choose to use it or not; e.g. President Bush in 2001 chose not to use e-mail because he did not trust it. The days when software can hold users hostage to its power are gone.

Successful online traders find politeness profitable. EBay's customer reputation feedback gives users optional access to valued information relevant to their purchase choice, which by the previous definition is polite. Amazon gives customers information on the books similar buyers buy, not by pop-up ads but as a view

option below. Rather than a demand to buy, it is a polite reminder of same-time purchases that could save customer postage. Politeness is not about forcing users to buy, which is anti-social, but about improving the seller-customer relationship, which is social. Polite companies win business because *customers given choices come back*. Perhaps one reason the Google search engine swept all before it was that its simple white interface, without annoying flashing or pop-up ads, made it pleasant to interact with. Google ads sit quietly at screen right, as options not demands. Yet while many online companies know that politeness pays, for others, hit-and-run rudeness remains an online way of life.

Wikipedia is an example of a new generation of polite software. It does not act pre-emptively but lets its users choose, it is visible in what it does, it makes user actions like editing easy rather than throwing up conditions, it remembers each person personally, and it responds to user direction rather than trying to foist preconceived good knowledge conditions on users. Social computing features like post-checks (allowing an act and then checking it later), versioning and rollback, tag clouds, optional registration, reviewer reputations, view filters and social networks illustrate how polite computing gives choices to people. In this movement from software autocracy to democracy, it pays to be polite because *polite software is used more and deleted less*.

## 4.13 A NEW DIMENSION

Polite computing is a new dimension of social computing. It requires a software design attitude change, e.g. to stop seeing users as little children unable to exercise choice. Inexperienced users may let software take charge, but experienced users want to make their own choices. The view that “software knows best” does not work for computer-literate users. If once users were child-like, today they are grown up.

Software has to stop trying to go it alone. Too clever software acting beyond its ability is already turning core applications like Word into a magic world, where moved figures jump about or even disappear entirely, resized table column widths reset themselves and moving text gives an entirely new format. Increasingly only Ctrl-Z (Undo) saves the day, rescuing us from clever software errors. Software that acts beyond its ability thinks its role is being to lead, when really it is to assist.

Rather than using complicated Bayesian logic to predict users, why not simply *follow the user's lead?* I repeatedly change Word's numbered paragraph default indents to my preferences, but it never remembers. How hard is it to copy what the boss does? It always *knows better*, e.g. if I ungroup and regroup a figure it takes the opportunity to reset my text wrap-around options to *its* defaults, so now my picture overlaps the text again! Software should leverage user knowledge, not ignore it.

Polite software does not act unilaterally, is visible, does not interrupt, offers understandable choices, remembers the past, and responds to user direction. Impolite software acts without asking, works in secret, interrupts unnecessarily, confuses users, has interaction amnesia, and repeatedly ignores user corrections. It is not hard to figure what software type users prefer.

Social software requirements should be taught in system design along with engineering requirements. A "politeness seal" could mark applications that give rather than take user choice, to encourage this. The Internet will only realize its social potential when software is polite as well as useful and usable.

## 4.14 DISCUSSION QUESTIONS

The following questions are designed to encourage thinking on the chapter and exploring socio-technical cases from the Internet. If you are reading this chapter in a class - either at university or commercial – the questions might be discussed

in class first, and then students can choose questions to research in pairs and report back to the next class.

1. Describe three examples where software interacts as if it were a social agent. Cover cases where it asks questions, makes suggestions, seeks attention, reports problems, and offers choices.
2. What is selfishness in ordinary human terms? What is selfishness in computer software terms? Give five examples of selfish software in order of the most annoying first. Explain why it is annoying.
3. What is a social computing error? How does it differ from an HCI error, or a software error? Take *one* online situation and give examples of all three types of error. Compare the effects of each type of error.
4. What is politeness in human terms? Why does it occur? What is polite computing? Why should it occur? List the ways it can help computing.
5. What is the difference between politeness and legitimacy in a society? Illustrate by examples, first from physical society and then give an equivalent online version.
6. Compare criminal, legitimate and polite social interactions with respect to the degree of choice given to the other party. Give offline and online examples for each case.
7. Should any polite computing issues be left until all security issues are solved? Explain, with physical and online examples.
8. What is a social agent? Give three common examples of people acting as social agents in physical society. Find similar cases online. Explain how the same expectations apply.
9. Is politeness niceness? Do polite people always agree with others? From online discussion boards, quote people disagreeing politely and agreeing impolitely with another person.
10. Explain the difference between politeness and etiquette. As different cultures are polite in different ways, e.g. shaking hands vs. bowing, how can

politeness be a general design requirement? What does it mean to say that politeness must be "reinvented" for each application design case?

11. Define politeness in general information terms. By this definition, is it always polite to let the other party talk first in a conversation? Is it always polite to let them finish their sentence? If not, give examples. When, exactly, is it a bad idea for software to give users choices?
12. For each of the five aspects of polite computing, give examples from your own experience of impolite computing. What was your reaction in each case?
13. Find examples of impolite software installations computing. Analyze the choices the user has. Recommend improvements.
14. List the background software processes running on your computer. Identify the ones where you know what they do and what application runs them. Do the same for your startup applications and system files. Ask three friends to do the same. How transparent is this system? Why might you want to disable a process, turn off a startup, or delete a system file? Should you be allowed to?
15. Discuss the role of choice dependencies in system installations. Illustrate the problems of: 1) being forced to install what is not needed and 2) being allowed to choose to not install what is.
16. Find an example of a software update that caused a fault; e.g. update Windows only to find that your Skype microphone does not work. Whose fault is this? How can software avoid the user upset this causes? Hint: Consider things like an update undo, modularizing update changes and separating essential from optional updates.
17. Give five online examples of software amnesia and five examples of software that remembers what you did last. Why is the latter better?
18. Find examples of registration pryware that asks for data such as home address that it does not really need. If the pry fields are not optional, what happens if you add bogus data? What is the effect of your willingness to register? Why might you register online after installing software?

19. Give three examples of nagware – software on a timer that keeps interrupting to ask the same question. In each case, explain how you can turn it off. Give an example of when such nagging might be justified.
20. Why is it in general not a good idea for applications to take charge? Illustrate with three famous examples where software took charge and got it wrong.
21. Find three examples of "too clever" software that routinely causes problems for users. Recommend ways to design software to avoid this. Hint: consider asking first, a contextual turn off option and software that can be trained.
22. What data drove Mr. Clippy's Bayesian logic decisions? What data was left out, so that users found him rude? Why did Mr. Clippy not recognize rejection? Which users liked Mr. Clippy? Turn on the auto-correct in Word and try writing the equation: i = 1. Why does Word change it? How can you stop this, without turning off auto-correct? Find other examples of smart software taking charge.

## Your notes and thoughts on chapter 4

Record your notes and thoughts on this chapter. If you want to share these thoughts with others online, go to the bottom of the page at: http://www.interaction-design.org/books/the_social_design_of_technical_systems_2nd_ed/polite_computing.html

**NOTES:**

# CHAPTER 5

# The Social Environment Model

"*People are social environment blind*"

The social environment model works for any society, modern or traditional, socio-technical or socio-physical, and for any social level or community size. This chapter links the arcane role of society to modern technology. It describes:

1. The social dilemma inherent to all social systems,
2. Traditional social responses to it,
3. A social system as an environment within an environment,
4. Enron and the credit meltdown as higher social errors,
5. STS as a new social design based on an old inspiration.

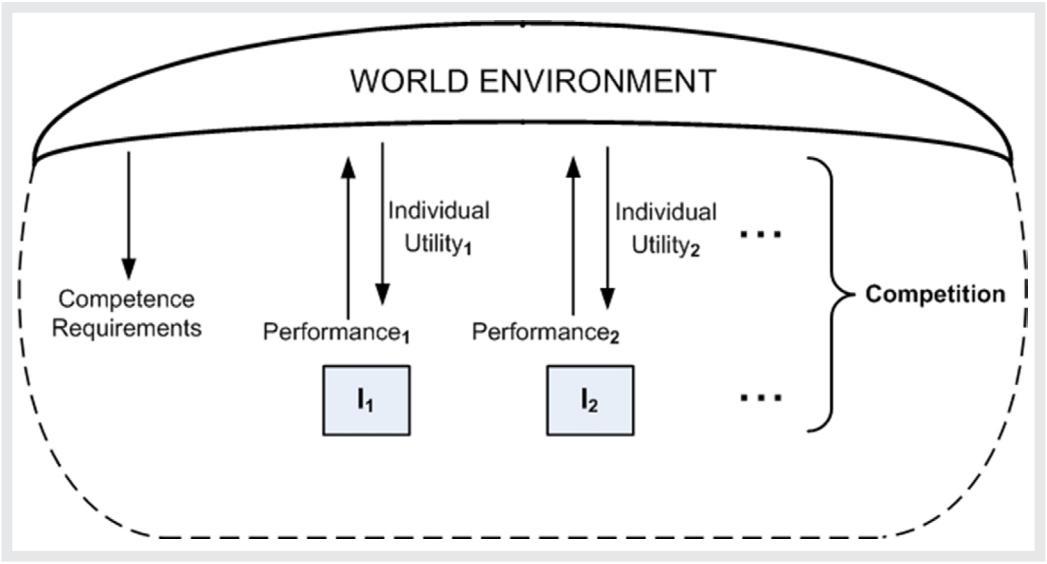



FIGURE 5.1: Individuals competing in a world environment.

*Courtesy of Brian Whitworth and Adnan Ahmad. Copyright: CC-Att-SA-3 (Creative Commons Attribution-ShareAlike 3.0).*

In the following discussion, bear in mind that social systems online and offline follow the same principles. A social system is always people interacting with people, regardless of the base architecture. Social implementations tried in the physical world, like communism or capitalism, can be tried online, and those tried online can be applied offline, as the Arab spring illustrates.

## 5.1 HOMO ECONOMICUS

In a limited resource environment, if two beetles independently compete for the same food and one gets it, then the other loses out. As the beetle with the food is more likely to survive, in natural selection individuals competefor advantage. The farmer growing the food also competes with the beetles and both compete with bacteria that would also consume it. Limited resource environments reward

individual competencies like strength or speed that increase success and survival. In Figure 5.1, *competition* for limited resources develops individuals to be *competent* to succeed in the world.

The *homo-economicus* model of society is that individuals seek to benefit themselves by less effort, more gain or both (Persky, 1995). Individuals competing for advantage favours the evolution of competencies by competition. Mill's *economic man* seeks wealth, leisure, luxury and procreation above all else, and Adam Smith postulates that such individuals in a free market also help society, as by the market everyone produces more (Smith, 1776/1986).

This model assumes people are rational actors who can calculate their own best interests, though they may actually use heuristics—psychologically efficient versions of rational logic (Tversky & Kahneman, 1982). Competition drives self-interested individuals to *competence gains* in the evolutionary process of natural selection.

That free individuals act in self-interest is a defeasible rule that can be described as follows:

**Rule 1:** If freely acting individuals $\{I_1, I_2 \ldots\}$ face action choices $\{a_1, a_2 \ldots\}$ with expected individual utility outcomes $\{IU(a_1), IU(a_2), \ldots\}$ then:

$$\textit{If } IU(a_i) > IU(a_j) \textit{ then prefer } a_i \textit{ over } a_j$$

**In words:** *Free individuals prefer acts expected to give more value to themselves.*

The concept "value" here is deliberately left vague, so it may include physical gains like food, social information tokens like money, psychological gains like appreciation, or social gains like reputation.

## 5.2 HOMO SOCIOLOGICUS

While Rule 1 is evident in nature, social cooperation is equally common; e.g. in the animal kingdom, social insects like ants form massively cooperative societies and are so successful they account for at least a third of all insect biomass. The genetics that drives their behaviour evolved because individuals working together can create more value than working alone (Ridley, 1996). For ants, the unit that competes and survives is not the individual but the colony; e.g. soldier ants die protecting the colony, as without it they cannot survive anyway. In this model, individuals combine into a community that performs, in evolutionary terms, based on the sum of the actions of its members (Figure 5.2).

So biologists now argue for *multi-level selection*—evolutionary selection for groups as well as individuals (Wilson & Sober, 1994). Social cooperation changes the evolutionary reward rule—individuals still act but the acts selected are those that create value for the community, not those that create value for the individual. That socialized individuals can generate community value suggests a defeasible social alternative to game theory's Rule 1:

**Rule 2: I**f a social unit S of $\{ I_1, I_2 \ldots\}$ individuals faces social action choices $\{a_1, a_2 \ldots\}$ with expected social utilities $SU(a_1)$, $SU(a_2)$, ...} then:

$$\textit{If } SU(a_i) > SU(a_j) \textit{ then prefer } a_i \textit{ over } a_j$$

**In words:** *Socialized individuals prefer social acts expected to give more value to the community.*

If it is the colony that lives or dies, not just the ant, the unit of evolution changes. Value outcomes are calculated for the group as a whole. Natural selection now favours Rule 2, i.e. the evolution of behaviours to help the colony survive. For ants, natural selection favours acts giving community, not individual, gain.

This then is a basis for Rule 2 to operate in people, allowing for social evolution. *Social acts* are those that reference the social unit not the individual; e.g. defending society is a social act independent of any individual state. Social castes can be dedicated to social acts, like worker or soldier ants.

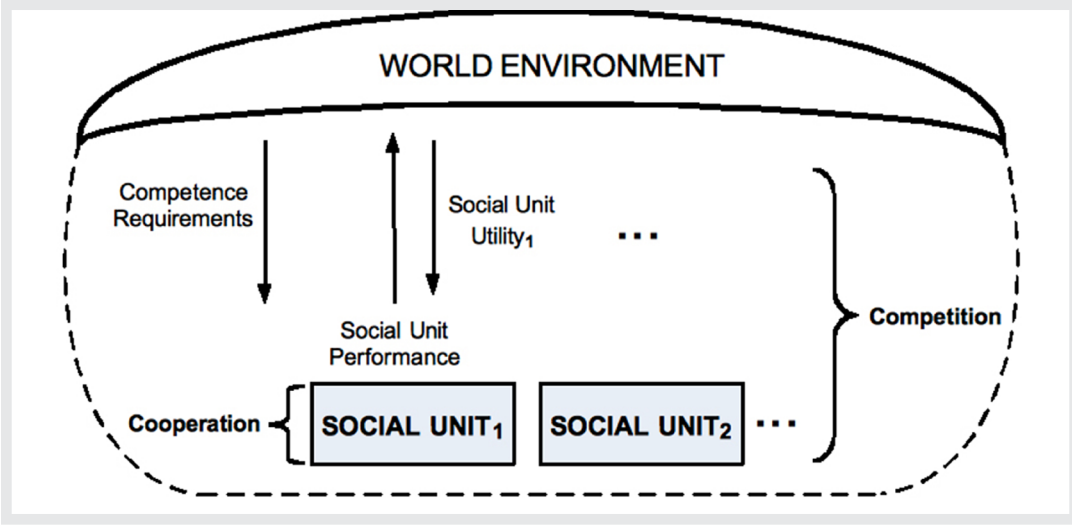



FIGURE 5.2: A community cooperating in a world environment.

*Copyright status: Unknown (pending investigation). See section "Exceptions" in the copyright terms below.*

Rule 2 applied to human society gives *homo sociologicus,* who prefers acts that benefit the community (Bone, 2005). This is Marx's *communist man*, who is politically motivated to common acts that benefit the community. A psychological basis for this motive is Social Identity Theory (Hogg, 1990), where groups form when members share a common identity, or idea of themselves. If anyone attacks *one* member of such a group, *all* group members feel attacked and respond accordingly. Most countries' defence forces work by this rule, as servicemen and women are expected to give their lives for society. In Figure 5.1 individuals reap the consequences of their individual performance but in Figure 5.2 the social unit bears the consequences of its performance.

These two pragmatic rules, one at the individual level and one at the community level, now interact to create social dilemmas.

## 5.3 THE PRISONER'S DILEMMA

Game theory is the systematic study of the rational choices of individuals in interdependent social interactions. It usefully presents the essentials of social situations for analysis and underlies many key economic, political and group management theories. Yet its validity was challenged by the discovery of a scenario called the prisoner's dilemma.

In the prisoner's dilemma two prisoners (Bill and Bob) each face two-year jail terms on circumstantial evidence for a crime they *did* commit. Each is separately offered a plea bargain, to testify against the other. If the other does not testify, he walks free but his partner gets seven years in jail. If both testify, both get six years (one off for testifying). In outcome utility terms the options are:

1. Bill and Bob stay silent, and each gets two years in jail.
2. Bill confesses for immunity, and Bob gets seven years.
3. Bob confesses for immunity, and Bill gets seven years.
4. Bill and Bob both confess, and both get six years jail.

Table 5.1 shows the outcome payoff matrix as free years out of seven. If both keep quiet, or *cooperate*, both get five free years, but if both testify, or *defect*, they only get one free year each. The *temptation* isfor one to defect and get seven free years, while the other cooperating "sucker" gets none. *As individuals* following Rule 1, each prisoner concludes:

- Whether he cooperates or defects doesn't depend on my choice.
- If he defects, it pays me to defect, as then I get 1 not 0.
- If he cooperates, it still pays me to defect, as I get 7 not 5.

| *Years free (Bill/Bob)* | | *Bob* | |
|---|---|---|---|
| | | *Cooperate* | *Defect* |
| *Bill* | *Cooperate* | *5/5* | *0/7* |
| | *Defect* | *7/0* | *1/1* |

TABLE 5.1: Prisoner's dilemma—Individual outcomes.

Game theory, following Rule 1, concludes that it always pays individuals to defect, as the expected defect average gain is 4 but the cooperate average is only 2.5. If both parties follow game theory, defect/defect is the *equilibrium state*. People rationally maximizing profit creates the worst possible result for all.

*However working as a social unit,* i.e. following Rule 2, gives a different result. The available social acts forthe pairare mutual cooperation and mutual defection, with expected gains of 10 and 2 respectively (Table 5.2). If both parties follow Rule 2 mutual cooperation is the new equilibrium state. So when social cohesion is allowed, simulated agents in a prisoner's dilemma situation evolve a cooperative equilibrium (Dayton-Johnson, 2003).

In performance terms, Rule 1 gives only 2 free years of value but Rule 2 generates 10 free years, a considerable improvement. Game theory *assumes* that rational beings *must* calculate payoffs for the individual, but it is just as rational to calculate payoffs for the social unit as a whole (Table 5.2). After all, Rule 2 is just Rule 1 applied to the social unit instead of the individual unit. It is just as logical and just as pragmatic.

Conversely, it is illogical to label alternatives to individual self-interest irrational if they generate more value. Indeed, we ourselves are a society of cells, and cancer is the result when one of them acts for itself alone, regardless of the

rest. Since what people *define* as "self" often includes the community around them (Persky, 1995), both rules are equally rational and pragmatically grounded.

| *Years free (Pair)* | | *Social Outcome* |
|---|---|---|
| *Social Act* | *Cooperate* | *10* |
| | *Defect* | *2* |

TABLE 5.2: Prisoner's dilemma—Social outcomes.

## 5.4 THE TRAGEDY OF THE COMMONS

The tragedy of the commons (Hardin, 1968) extends the two-person prisoner's dilemma to a case of many people in a group[1]. In it, some farmers each with cows and a plot of land live by a common grass area. If a farmer's herd also grazes the commons it grows fat, but if over 50% of farmers do so, the commons is over-grazed and dies off. This parallels many forest and river conservation problems.

*Working as individuals* each farmer's logic is:

- My actions are independent of those of the other farmers.
- If ≤ 50% graze it pays me to graze, as I get more value.
- If > 50% graze, it still pays me to graze, as I get more value initially.

As it always pays each farmer to graze the commons, by game theory they must destroy it. In a hypothetical case, 100 farmers each get a ton of beef per month grazing their own plots, and three more tons grazing the commons. This reduces by one ton each month of overgrazing, to become barren in three months. Table 5.3 shows farmer outcomes by choice for 10 months. By Rule 1, the average graze benefit is 28, while the average not-graze benefit is 10, so graze is preferred. Destroying the commons is the equilibrium point.

1. The tragedy of the commons depicts whaling, forest and wildlife conservation issues.

| **Outcome** | | *Others* | |
|---|---|---|---|
| | | *49% graze* | *Over 49% graze* |
| *Farmer* | *Do not graze* | *10* | *10* |
| | *Graze* | *40* | *16* |

Table 5.3: Tragedy of the commons individual outcomes.

*Working as a social unit* gives a different conclusion. The social acts available to *the village* are by what percentage to graze the commons. Table 5.4 shows the expected outcome per farmer for overgrazing is 1,600 tons over 10 months, while the expected not overgrazing value is 2,500, making it the preferred choice. A village following Rule 2 will save the commons as a valuable community resource.

The fact is that social cooperation works: Axelrod invited programs for a simulated survival of the fittest social interaction tournament to see which survived. He found that none of the eight most successful programs initiated defection (Axelrod, 1984). Nasty programs succeeded at first but in time ran out of victims and met only other nasties. Cooperative programs found allies and prospered.

| **Outcome** | | *Social Outcome* |
|---|---|---|
| *Social Act* | *Not overgraze* | *2,500* |
| | *Overgraze* | *1,600* |

Table 5.4: Tragedy of the commons social outcomes.

## 5.5 SYNERGY

The ability of a group to perform at a social level depends on its ability to generate *synergy*. Social synergy is the difference between what individuals produce by acting together compared to what they produce apart. It can be positive or negative, where trade is a positive synergy and war a negative one. People tend to join communities with positive synergy and leave those with negative synergy, whether people leaving websites plagued by conflicts or refugees fleeing war torn nations.

Synergy is a property of the social interaction, not the social individuals. In the prisoner's dilemma, the interaction is the loyal friendship total (10) less the defect total (2), i.e. 8 years. In the tragedy of the commons, it is the village cooperative total (2,500) less the competitive total (1,600), i.e. 900 tons.

Science shows the benefits of synergy. To illustrate, suppose 100 solitary researchers each make different knowledge advances of equal value. Following a *zero-sum model,* as private companies do, they will keep their results secret, as why let competitors benefit from my work? In contrast, academics following a *non-zero sum* model will share their research[2]. In the first case, the total knowledge increase is 100 units of human understanding, but in the second case it is 1000 units, as each researcher gets 99 new ideas from others as well as their own. This is a hundred-fold gain vs. keeping research secret. If the scientists of history had kept their research secret, benefits like electricity may not have occurred. Yet the decoding of the human genome sequence was nearly patented for commercial gain rather than made available to all. What is the justification for people patenting what Nature created?

Social synergy arises when people work to create *others'* outcomes. It is not just people adding their efforts, say to lift a heavy log together. In positive synergy,

2. This may change if academic journals become knowledge fortresses, see http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/2609/2248

specialists create value for others; e.g. if a fisherman trades excess fish for a farmer's excess grain, both turn their excess into value. Each gives an extra they do not really need for a deficit they do. Conflict is the reverse, as each creates negative outcomes for the other.

Large communities produce more because more citizens do more work, so productivity based on individual competence increases linearly with group size. In contrast, synergy depends on citizen interactions, which increase geometrically with group size. So large communities produce more but synergize *much more*. Synergy is especially important for large societies. When millions synergize, as today, the output gains are enormous. Synergy is why ordinary people today have better food, healthcare and entertainment than the richest aristocrats of the middle ages, and today's rich have more money than they can spend in a lifetime.

Synergy is also especially applicable online, as one can give information to others without losing it oneself. So when the Internet allows millions to synergize information, the effect is especially powerful, as systems like Google illustrate. The prosperity of modern society is based on people specializing and synergizing.

## 5.6 DEFECTION

Anti-social acts like stealing are individuals taking from a social interaction but not giving anything back. The drive to get something for nothing permeates the current system from crime to bargains and gambling. So every social synergy has a corresponding defection, e.g. in trade, sellers can defect by false advertising, shoddy products or bad warranties. If so, buyers buy less. Buyers can also defect; e.g. buy an expensive ball gown, wear it to a ball, then falsely request a refund, saying it did not fit. If many customers do that, sellers offer less, e.g. refuse refunds (also defect), even though refunds benefit both seller (more sales) and buyer (less risk).

Game theory, by formally calculating personal social interaction outcomes, points out the fly in the social ointment of synergy. If my acts give your gains and yours give mine, what if I take from you but give nothing back? On a personal level it always pays to defect, e.g. for a seller to give shoddy goods or for a buyer's cheque to bounce.

Yet if the cheated "sucker" does not repeat the interaction, both lose their synergy gains, so cheaters destroy their own success. If a crime-wave "succeeds", the social benefits it feeds on dry up. Crime is like a social parasite that kills its host. The idea that one can get something for nothing is the big lie of our generation. All crime is essentially socially unsustainable. The myth of pure profit is an impossible dream.

Game theory suggests that mutual synergy is like a ball balanced on the crest of a hill, that must sooner or later roll permanently down into the valley of mutual defection, yet we still generate synergy even after thousands of years of society. Crime can short-circuit the link between social acts and synergy, but it has not prevailed, even though defections can cascade into social collapse. If I defect I not only gain personally but also reduce the synergy gains of others, increasing the pressure on them to also defect. If another person also defects, this increases the pressure on the remainder to defect, etc. Hence a common reason given for cheating is that "everyone is doing it" (Callahan, 2004). A few defections can cause a chain reaction that destabilizes an entire social system.

## 5.7 SOCIAL DILEMMAS

The prisoner's dilemma was thought to be an exceptional case until it was found to be the social rule. *Social dilemmas* like the volunteer dilemma[3] represent a generic problem inevitable in synergistic societies (Diekmann & Lindenberg, 2001).

3. In the volunteer dilemma, a group needs volunteers to survive but it pays individuals to leave it to others so it collapses.

In the social environment model, *social dilemmas arise when Rule 1 contradicts Rule 2*, i.e. when what benefits the individual does not benefit the group, or when what benefits the group does not benefit the individual. The dilemma is not that people selfishly follow Rule 1 or that they ethically follow Rule 2, but that they are caught between. Certainly game theory's Rule 1 is insufficient, as people in social dilemma games are much more cooperative than game theory predicts (Poundstone, 1992). The mystery is not why people synergize or why they are selfish, but how they can be both.

Game theory implies that society cannot succeed but it already has. Deducing that social cooperation is irrational (von Neumann & Morgenstern, 1944) is like deducing that bumblebees cannot fly by the laws of physics, when in fact they do. In science, we change the theory not the facts. Human instincts know what human logic does not: *that synergy works*.

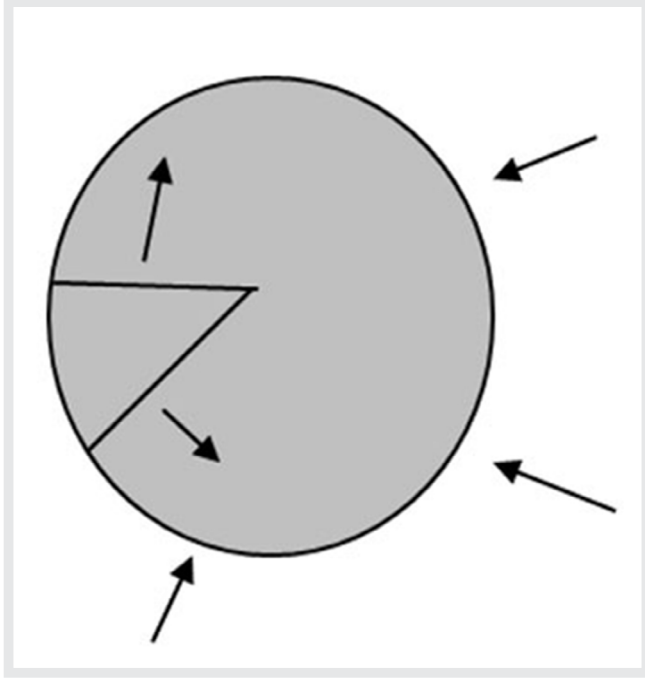
Social dilemmas cannot be solved at the personal level, because an honest person among cheats is just a fool. *Individuals alone cannot solve social dilemmas*. One person trying to synergize in a social dilemma is just a *sucker*. In the tragedy of the commons, the farmer who on principle does not graze just misses out and the commons is destroyed anyway. The choices for individuals in social dilemmas are all bad, so how did we achieve synergy at all?

The solution to all social dilemmas is to evolve a higher social unit to change the gain-loss equation. This is not easy. It has taken thousands of years of often bitter struggle to stabilize massive synergies like global trade and international peace. The heroes of our social evolution were those who saw beyond themselves. The path to social synergy has on both sides the cliffs of defection. We alone among the mammals have crossed thezero-sum barrier into the lush valley of massive social synergy (Wright, 2001). We were lucky.

## 5.8 THE ZERO-SUM BARRIER

Game theory differentiates between zero-sum and non-zero-sum games. In zero-sum games, like poker, your loss is my gain, so if you lose I win at your expense. If everyone fights, the winner gets the biggest share. In these *zero-sum games*, taking another's slice of the reward pie just increases your share.

In contrast, in *non-zero-sum games*, which give social dilemmas, we all connect, so your loss is my loss. If we all fight, we all become poor, as dog-eat-dog societies evidence. In non-zero-sum games, taking another's slice shrinks the whole pie, e.g. Hitler conquered Europe but the process left it in ruins. Conversely, synergy works by increasing the shared pie for all, making every slice larger, as market trade benefits all (Figure 5.3). *Non-zero-sumness* is an unpleasant term, but the argument that it is the secret to modern prosperity is a strong one (Wright, 2001).



*Courtesy of Brian Whitworth and Adnan Ahmad. Copyright: CC-Att-ND-3 (Creative Commons Attribution-NoDerivs 3.0 Unported).*

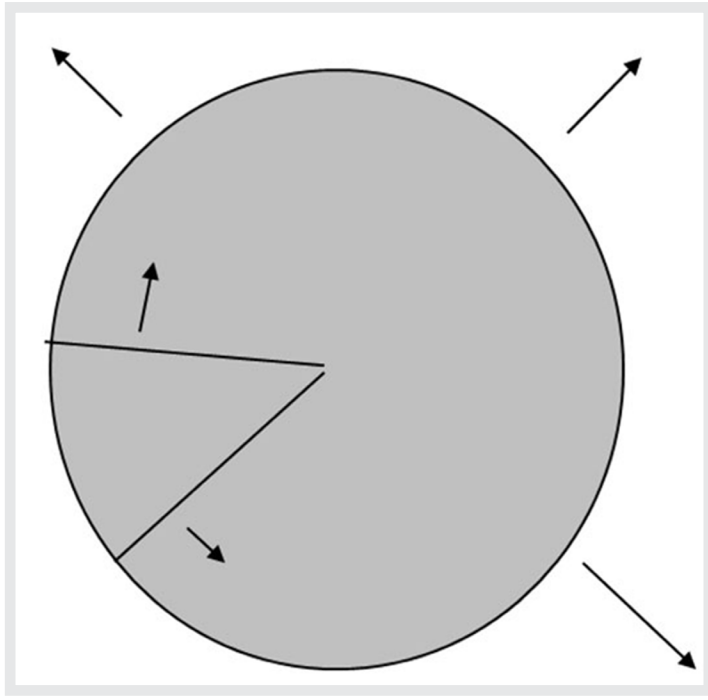


Figure 5.3 A-B: Zero-sum vs. Non-zero-sum interactions: A. a. Zero-sum: Expand my slice but shrink the pie, e.g. Hitler's conquest of Europe. A: Non-zero-sum: Expand the pie and everyone's slice, e.g. the evolution of civilization.



Synergy is also the key to socio-technical system design, as these systems must enable synergies and deny defections to succeed. While traditional word processing software just has to increase user competence, socio-technical systems must also increase community synergies and defend against anti-social defections. If users just followed Rule 1, systems like Wikipedia would not work, as no one would give to others for no gain. Conversely, if people just followed Rule 2, these systems would not need defenses against anti-social defections.

In forums like AnandTech, if anyone in a group solves a problem then everyone gets the answer. In online security, if one person gets a virus the community responds. The larger the group, the more likely someone can solve in seconds a

problem you might take days to solve. Same again functions let Amazon readers tap into the experiences of others, to find books bought by those who bought the book they are looking at now. Wikipedia users correct errors of fact, supply references and offer examples for everyone. Table 5.5 shows how socio-technologies increase synergy and reduce defections.

| Purpose | Examples | Synergy | Defection |
|---|---|---|---|
| *Communicate* | Email, Chat, ListServ, IM | *Communication.* Send useful messages | *Spam.* Spammers require spam filters |
| *Learn* | Moodle, Blackboard | *Share learning:* Pupils help others learn | *Plagiarism.* Students copy (turnitin.com) |
| *Know* | Wikipedia, Tiddlywiki | *Share knowledge.* Tap group knowledge | *Trolls.* Wikipedia monitors "trolls" |
| *Friend* | Facebook, Myspace, Bebo | *Relationships.* Relate to friends, family | *Predation:* Social networks banish predators |
| *Keep current* | Digg, Del.icio.us | *Share bookmarks:* Show online trends | *Advocates:* Mark their own web sites |
| *Play* | Second Life, The Sims, MP games | *Shared play.* People interact in a virtual world | *Bullies/Thieves.* Rob newbies who need "safe" areas |

| | | | |
|---|---|---|---|
| *Trade* | E-Bay, Craig's List, Amazon | *Item trading*. People trade more goods | *Scams*. Reputation systems cut scams |
| *Work* | Monster | *Job trading:* People find and offer work better | *Faking*. Padded CVs and fake job offers |
| *Download* | Webdonkey, Bit-Torrent | *Shared downloading:* Groups share download work | *Piracy*. Prosecution by society's copyright laws. |
| *Publish* | Flickr, YouTube | *Share viewing:* Share photo and video experiences | *Offensiveness:* Editors remove items that offend. |
| *Advice* | Help boards like AnandTech | *Share advice:* People help others solve problems | *Confusers*. People put old questions in new threads |
| *Discuss* | Slashdot, Boing-Boing | *Shared views*. People comment and share opinions | *Caviling:* Karma systems deselect negativity |
| *Follow* | Twitter | *Links leaders and followers* | *Identity theft*. A persona is hijacked |

TABLE 5.5: Synergies and defections in socio-technologies.

## 5.9 THE SOCIAL ENVIRONMENT MODEL

In general, we do not see environments, not because they are too far away but because they are too close. As a fish is the last to see water, or a bird the air, so we as social animals tend to be social environment blind. In the social environment model, a social system is an *environment within an environment* (Figure 5.4). This model combines the homo economicus and homo sociologicus views.

A social system is an environment to its members because it imposes requirements on them (laws and norms) and dispenses their gains and losses (salaries and taxes). It does the latter by social *tokens* likemoney, which are exchanged for world value like food.

A social system can then fail by incompetence with respect to its environment, as a wasteful company going bankrupt, or by internal conflict, if it fails to create synergy or prevent crime. People in a society thus operate under two environments, one that rewards them for competing and one that rewards them for cooperating. Social problems can arise if either environment is not satisfied.

Table 5.6 shows how people label various combinations of individual and community outcomes. The "selfish" Rule 1 directs individuals to choose the first *row*, while the "good" Rule 2 directs them to choose the first *column*. It works for biological as well as social behaviour, as the first row is symbiosis, commensalism and predation respectively. People however have a choice, between their *instincts* for personal gain (Rule 1) and their *social training* for synergy (Rule 2).

If people only followed Rule 1 crime would prevail and society would collapse. If they only followed Rule 2, we would be like ants, locked in under genetically bred kings or pharaohs. Our actual social path was a pragmatic combination of Rules 1 and 2.

The problem facing humanity was how to combine Rule 1 and Rule 2 in a way that is feasible, i.e. doable by people. A well known solution is the utilitarian ideal of "*the greatest good for the greatest number,*" as popularized by Dr Spock's

sacrifice in Star Trek II: The Wrath of Khan. It seems simple to prefer the greatest good but what is that for millions of people over time? Is an aircraft crash that loses hundreds of lives today but causes safety changes that saves thousands of future lives "good"? Should governments then crash planes or sink ferries to introduce needed safety measures? The problem with the utilitarian solution is that the greatest good of millions of people over hundreds of years is not calculable by anyone, let alone a majority of people. It is valid but not feasible.

Nor does a simple AND of Rules 1 and 2 work well. It is feasible but not optimal, as people acting only to benefit *both* themselves and society would often not act at all. Finally, any sort of weighted trade of social utility against individual utility raises complex questions, like how much social good is my individual loss worth, or how much individual good warrants a social loss? On this logic, old people would go to war, having less to lose, while young people would want to stay at home.

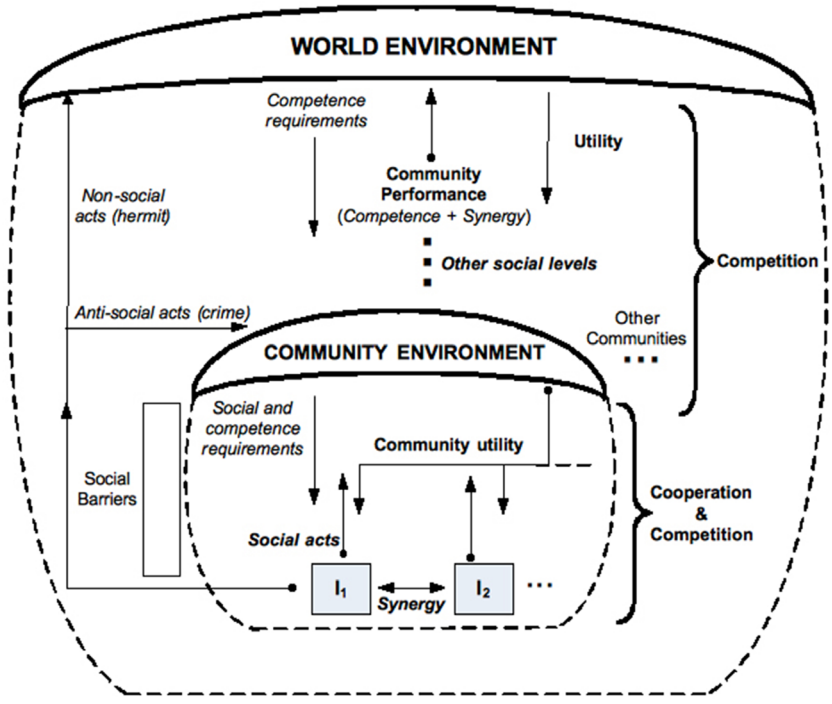



FIGURE 5.4: The social environment model.

What is needed is a rule combination that is simple enough for people to conceive. Such a solution is *cognitive anchoring*, fixing one rule and then applying the other (Tversky & Kahneman, 1974). This approach suggests the following combinations:

**Rule 3a:** *If* $\{SU(a_i) \geq SU(a_j)$ *and* $IU(a_i) > IU(a_j)\}$ *then prefer* $a_i$ *to* $a_j$

**In words:** *Choose acts that do not harm society but benefit oneself*

**Rule 3b:** *If* $\{IU(ai) \geq IU(a_j)$ *and* $SU(a_i) > SU(a_j)\}$ *then prefer* $a_i$ *to* $a_j$

**In words:** *Choose acts that do not harm oneself but benefit society*

Following Rule 3a, individuals seek opportunities that do not hurt society, as it gives in its own laws, i.e. one competes by the rules.

Following Rule 3b, individuals help others in society if it does not involve too much personal loss.

These rules are much easier to apply than ideals of calculating the greatest good for the greatest number or percentage trade-offs of individual vs. community benefits.

Applying Rule 3 to Table 5.6 gives *synergy, opportunity and service* as the main directives of human society. Those following Rule 1 exclusively are self-interested criminals, whose acts collapse society. Those following Rule 2 exclusively are altruistic saints willing to sacrifice themselves for society at any time.

In contrast, most people follow neither crime nor altruism alone. If the community goes to war so do they, but they still try to survive personally. They try to get ahead without disobeying laws, but also help community others out if the cost is not too much. That people *naturally* help others is *not* part of game theory. Only the social environment model recognizes both Rule 1 and Rule 2.

| | | COMMUNITY | | |
|---|---|---|---|---|
| | | *Gain* | *Minor Effect* | *Loss* |
| **SELF** | *Gain* | **Synergy** | **Opportunity** | Crime |
| | *Minor Effect* | **Service** | Null | Malice |
| | *Loss* | Sacrifice | Self-harm | Conflict |

Table 5.6: Synergies and defections in socio-technologies.

## 5.10 SOCIAL ORDER

Before considering modern ways to cross the zero-sum barrier, including those enhanced by technology, it is worth considering the traditional ways. By rule 2, *social dilemmas are solved when people form a higher social unit*. If the commons farmers form a village, they can institute a cooperative grazing roster to preserve the common grazing. Game theory arbitrarily excludes the social agreements critical to solving social dilemmas (Aumann, 1998).

A society's *social order* is the degree to which its members follow common rules. In perfect social order everyone is of "one mind", like an ordered crystal whose constituent atoms all move as one. Social anarchy in contrast is like a gas, whose atoms all move randomly according to individual exigencies. *Freedom* lets members of a society choose whether to act as one or in opposition, and allows creative acts against the norm. Enforcing order avoids anarchy, but also reduces freedom and creativity.

If a community acts as one (social order), whether by religion, culture, law or coercion, social dilemmas give way to synergy; e.g. a village can set up a game reserve to stop poaching, individuals killing animals for personal gain. The village

can conserve its resource to create, for example, a tourist income. It can do this by physical barriers like fences or by declaring the land sacred, so those who defy the gods become its enemies and can expect banishment.

Enforcing order, even psychologically, is a blunt instrument. It makes members effectively ants, i.e. denies freedom. Socializing citizens to follow Rule 2 engages social evolution to synergy but disengages the individual evolution of competence and creativity. The struggle between social and individual evolution is reflected in the historical swings between the rise of civilizations and their fall at the hands of more vigorous barbarians. Social performance, in this view, requires both individual competence and social synergy.

## 5.11 SOCIAL HIJACK

Centralizing control structures to create social order, by means of a king, emperor or pharaoh, invites social hijack. Social hijack is when individuals take control of a community for their own ends, as a virus can hijack a biological organism to serve its purposes. Plato's ideal leader was a benevolent dictator, who enforced social order to create synergy but justly returned society's gains to its citizens, i.e. enforced social synergy but returned it to those who created it.

Yet dictators are also the worst of leaders if they use society's performance for their own personal luxury or power ends. This is not legitimate, so they must *repress* individuality by police state control and *indoctrinate* the masses into blind service by media propaganda. This makes dictatorships:

1. *Unstable*. If those who create social wealth gain nothing from it, they have, as Marx notes, "nothing to lose but their chains". A privileged aristocracy living in luxury while the workers who create that wealth starve invites a grass-roots revolution.

2. *Impermanent*. Royal bloodline dynasties ensure that when kings, emperors, pharaohs eventually die the power vacuum is filled by their offspring. Yet inevitably time produces incompetent or even insane offspring whose foolish acts lead to a civil war and the collapse of the dynasty.
3. *Unproductive*. When a society blindly follows the whims of leader(s) isolated by wealth from world realities, it becomes incompetent and fails to generate produce from the world.

Societies with absolute rulers, like Burma and North Korea, tend to be poor. Their rulers replace natural productivity requirements by their personal social agendas; e.g. in Zimbabwe Mugabe addressed social inequity by driving white farmers off productive farms. Then he gave them to his cronies, who looted but did not plant, grow or harvest. Equity without productivity turned what was the bread-basket of Africa into the basket-case of Africa.

Social hijack is an evolutionary dead-end, changed only by the leader's death, social collapse or both. The physical starvation of millions by incompetence does not trigger revolution as only a social challenge can defeat a social system. Yet a prosperous society needs both competence and synergy. Synergy is like the interest paid on the capital of competence—if there is no capital there is no interest either. Synergies from social order (Rule 2) add to the competence gains of natural competition (Rule 1) but cannot displace them.

Yet neither is Rule 1 alone sufficient, as it ignores synergy. Most now reject the Social Darwinist argument that since nature weeds out the weak so should society. As Nature's web of performance tolerates an extraordinary diversity of life, so it behooves a society to be tolerant. Who knows which citizen will generate progress?

## 5.12 SOCIAL INVENTIONS

Periodically, society discovers new ways to increase social performance. Individuals in competition can attribute their results to their acts, but in social environments, our gains arise from the acts of others. We therefore invented accountability, that we are responsible for the effects of our acts not only on ourselves, but also on others. Who actually discovered this is lost in the mists of history, if it was even one person, but even so it was an invention, a social invention.

While forming a social unit engages synergy, it disconnects individuals from the direct consequences of their acts. *Justice*—punishing unfairness—is one way people restore this. Unfairness is here not merely inequity—the unequal distribution of outcomes—but *distributing outcomes according to contribution*[4]. Studies suggest that people recognize justice, that value gained matches contribution made, and tend to avoid unjust situations (Adams, 1965). People even prefer fairness to personal benefit (Lind & Tyler, 1988). In contrast, chimpanzees are simple outcome maximizers, who follow Rule 1 entirely (Jensen et al., 2007).

Criminals destabilize society, but justice changes the dynamic by punishing unfair acts. If individuals seek revenge on those who "wronged" them or their family, cheating is unprofitable over time, as today's defection is paid back with interest tomorrow. If a society can make unfair interactions a bad choice, selfish people will prefer mutual synergy to mutual conflict, i.e. justice aligns individual good and social good. Unfortunately, in "an eye for an eye" cultures one revenge act creates another, giving endless vendetta cycles, as in the Middle East. Revenge was the precursor of justice, as individuals administered justice personally, rather than leaving it to society. The case has been made that our entire justice system of police, laws, courts and prisons is simply to *deny unfair acts* (Rawls, 2001).

4. An old Zen master worked in the community garden. His pupils told him he shouldn't work due to his age. He continued to do so, until one day they hid his garden tools. From that day he didn't eat. Scared their master might die, they returned his tools. As he raked the lawn again, he said to them: *No work, no eat*. He understood synergy.

People often fail to see how the community level operates. A theft is "good" for the robber but is always bad for the community. If someone steals $100 and is caught penniless, a court may sentence them to a year in jail. If the police, trial and incarceration costs are over $100,000, and the robbed get no return, where is the value? If everyone loses, why waste money prosecuting? The error is to apply a personal perspective to a social level problem. For a community, $100,000 may be a small price to pay for social order. The state calculates at the community level, not the individual level; e.g. depression reduces productivity but no laws deny it because it affects the outcome of people not communities.

Social rules are about *changing social interaction contingencies,* not individual profit or loss; e.g. the 1980 clean up of New York crime changed the social environment, from one where shootings were common to one where it was safe to walk the streets. The increased productivity of an entire city was worth the effort. Spending thousands of dollars in police, court and prison costs to prosecute a hundred dollar theft is a community level good deal, as successful crimes create copycats and one defection can snowball into a social collapse[5]. Giuliani's clean up of crime in New York cost millions but generated a synergy gain of billions[6].

*Democracy* is another social "invention", that a community selects its leaders by information vote rather than physical conflict (Mandelbaum, 2002). Democracy vests the power to control the community in the community itself rather than in a king or dynasty, so democracies also have constitutions limiting terms of office. A dictatorship has a centre to hijack but a democracy that distributes control to the people does not. This turns out to be better than trusting central elites, however benevolent, not because it is more efficient but because it allows anarchy free transitions of power. Given our human history of bloody power struggles, it is always amazing to watch a democratic leader who has lost a vote peacefully hand over control to a successor.

5. If a fast-food restaurant is kept clean people drop less rubbish, but if it is messy they drop more. So it is less work to keep it fully clean than partly clean.

6. He followed Wilson's Broken Windows Theory: https://en.wikipedia.org/wiki/Broken_windows_theory

Democracies combine individual freedom, social order and resistance to hijack. They produce more because free people contribute more work, more ideas and more research. They also self-regulate more, reducing security costs (Tyler, 1999). The allied democracies overcame the Axis dictatorships in World War II by producing more as well as fighting better. Democratic nations have increased over time not because democracy is "nice" but because it is productive.

## 5.13 SOCIAL HEALTH

It is not obvious that social change often requires social health. Those who can't conceive freedom can't achieve it. If democracy is unthinkable then it is unworkable. Freeing oppressed people who are not ready only leads to anarchy and a return to autocracy, e.g. the French revolution gave freedom, then the Terror, then a return to an emperor (Napoleon)[7]. Yet America and England tried the same and it worked. They crossed the zero-sum barrier to democracy and today it is unclear why our predecessors ever settled for less. Democracies out-produce autocracies because free people produce more and online is no different (Beer & Burrows, 2007). The gain is in the synergy, based on social interactions, based on ethical people.

So the synergy a community can generate depends on the social health of its citizens. Social evolution requires individual evolution because each level emerges from the previous one. Social health has also been called social capital, defined as the *"norms and networks facilitating collective action for mutual benefits*" (Woolcock, 1998).

Unlike ants, people learn to socialize, e.g. young players in soccer trail the ball like a cloud of insects, as each tries to get the ball and score a goal. Inevitably, they obstruct each other and the team results are poor. Only with social training do players learn positions like forward or fullback, and engage in social acts like passing the ball. In sports, both individual competence and team synergy are important.

7. A community can be governed in various ways. Autocracy is control by one person, aristocracy is control by an elite, plutocracy is control by the rich, democracy is control by all the citizens and anarchy is no-one in charge.

As one can test individual skills, so one can test community synergy, e.g. if a group offers cheap coffee on an "honesty" system of 25¢ per cup, what percentage cheat and take the coffee but leave no money? If everyone defects and takes the coffee for free, the synergy (and the coffee) fails.

Fast-food chains in different suburbs illustrate this with respect to the synergy of self-service, where customers can help themselves to tea, coffee, milk and sugar etc. Serving speeds improve as servers just give customers a cup, reducing long queues. However, if social health is low, people loot the beverage resources put out, so they are then kept behind the counter. This causes delays and long lines, as servers must pour the milk and get the sugar for each customer.

Similarly, the social invention of supermarkets required a certain amount of customer social evolution, as traditional shopkeepers kept goods behind the counter to prevent theft. Only if most customers do not steal, can goods go out on shelves to allow customer self-help, improving efficiency enormously.

Social health, the *percentage who defect on social synergies,* affects social performance. It is equally important online, where social systems are subject to natural selection, as people flow into communities that perform and out of those that do not. A physical barrier like the Berlin Wall can stop such flows for a while, but even it was unsustainable. It is even less possible online, where people join and leave communities with the click of a mouse.

Online communities can improve their social health; e.g. by a constitution that members accept as they register, by routine posts to remind people of ethics, or by moderation. If senior members help new members, the newcomers are more likely to give back. Banning trolls also raises the social health of the group if not too oppressive. Regular social health feedback of a group defection measure, like the number of posts deleted by moderators, would raise social health awareness and improve social performance.

## 5.14 COMMUNISM AND CAPITALISM

In the political conflict of capitalism and communism that dominated the last century, competitive value (Rule 1) was the assumed opposite of community value (Rule 2). However in the social environment model, they are underneath the same, as Rule 2 is just Rule 1 applied to the social instead of the individual unit. If the synergy gains are returned to the people who generated them, both options allow stable societies. Capitalism focuses on individual productivity via markets while communism focuses on community sharing, yet a community that produces little and shares it equally is hardly better than one that produces a lot but shares it unfairly. Humanity need not choose between the unequal distribution of wealth and the equal distribution of poverty, between individual productivity and community synergy. We can we have both personal productivity and community synergy given personal freedom and social goodness, as section 5.16 suggests.

Adam Smith linked individual to public good by suggesting the "invisible hand" of a market maximizing profits gives group value (Smith, 1776/1986). So in the context of a social market, the "greed is good" maxim applies: the harder people work for themselves the more value they generate for the community. Yet Smith's argument *for* competition is not an argument *against* cooperation, which he also believed in. In competitive sport, referees penalize illegal acts and free markets have common good rules, e.g. against insider trading. As sociologists like Granovetter argue, individual economics is always embedded in a larger social context outside any competitive framework (Granovetter, 1985). Playing fields and competitive environments work best when level.

The social environment model both supports Smith's link and lets it work the other way, i.e. as competition can support public good, so public good can support competition. To contrast the two is like arguing that mother is better than father or father is better than mother, when really both together are best. If capitalist models included synergy, businesses like Microsoft would not have found Berners-Lee's World Wide Web "uneconomic".

Rule 3, in both its forms, combines the *capitalist* view of society as self-interested individuals and the *communist* view of society as ant-like cooperatives. That citizens help both themselves and society is neither pure capitalism (Rule 1) nor pure communism (Rule 2). If pure communist societies have lower productivity and pure capitalist societies lower socialization, a hybrid will perform better than either alone. This predicts that communist countries will move to acquire a business face and capitalist countries will move to increase public good, as both are. Eventually these apparent political opposites will meet in the middle to be indistinguishable. In this model, the invisible hand of market competition works best with the visible hand of public good.

## 5.15 SOCIAL INFLATION

It is an error for any social environment to try to insulate its members from the demands of *its* environment. This cannot be done, as outer environment demands ultimately "cascade" over inner ones. Social environments that ignore the demands of *their* environment experience *social inflation*, where the value of the tokens it distributes to members lose their external value. Monetary inflation illustrates this, as money (a social token) loses value relative to external standards (like a loaf of bread).

Grade inflation occurs when professors give all students A's regardless of competence, and the token "A grade" loses value in the University's environment, i.e. with employers. Internally giving high grades seems to benefit everyone, as grading is easier, students are happier, and high pass rates attract more students. Yet externally it gives no value to the society at large, so is unsustainable.

Crime and corruption deny the requirements of the social environment but social inflation is denying the requirements of theenvironment of the social environment. It builds gradually, like a choir slowly going off-key together, but ends suddenly, in the failure of the entire social unit.

In social inflation the social unit as a whole goes against its environment, i.e. does not satisfy its requirements. Unless there is an internal rectification, eventually there must be an *external rectification*. World events like the great depression and the world wars illustrate external rectifications, as did the 2007–2012 global financial crisis. This world gives gains at the cost of risk, but banks and credit companies began offering loans regardless of risk. Internally this seemed to benefit everyone—lenders got more interest, borrowers got needed money and bank popularity increased. As some banks increased lending, others followed suit to keep in the market. Finally, when companies could not recover their loans, bad debt decreased the share token value.

The expected result of letting an external rectification "run its course" is the collapse of the social system and its synergy, in this case the global credit system, usually followed by depression or war. Knowing this, the US and other governments stepped in with billion dollar bailouts, but without an accompanying internal rectification this will only delay the inevitable external rectification. No social system can deny its environment. As shown in Figure 5.4, a social system must not only demand synergy from its members but also pass on the requirements of *its* environment to them.

Businesses leaders who cheat society of billions are removed but what of those who lost even more money by incompetent risk management? Both cases are social errors. If the same people who caused the credit collapse still draw bonuses based on their business "skills", no correction has been made. As Enron was a higher level of unethicality, so the credit collapse was a higher level of incompetence. In the social environment model, competence and synergy are both important to social performance. A society need not punish bank leaders for negligence but should remove them for the same reason it removes criminals— for the good of society.

Fiascoes like the credit crunch and Enron highlight the issue of private business and the state. When Wall St's credit froze, through its own errors of judgment, the US state stepped in to pay the $700 billion heating bill, quoting the public good. Similarly, when Enron's naughty boys, playing with the matches of cheating, nearly burnt down the share market house, the state again stepped in, again for the public good. To expect state bailouts in hard times but no state interference in good times is like a child who wants to be left alone but expects its parents to pay the bills. If public good is important, it is important all the time, not just when there is trouble. If in times of trouble the nation must pay the piper, then in times of plenty it can call the tune. For corporate cheats, it can set public-good rules of financial disclosure or rule that no company can pay zero tax. For corporate incompetence, it can replace the incompetent by those with real skills. Any society that fails to act in its own interests in such ways invites its own collapse.

As crime arises if citizens are *under-socialized,* social inflation arises if they are *over-socialized*. Too much of a good thing becomes a bad thing, as the organization becomes:

1. *Bureaucratic*. People follow social rules regardless of practical consequences. When rule following is the primary directive, the group becomes externally incompetent.
2. *Image focused*. When social appearances supersede practical skills, people with fake qualifications can get high positions. As image wins over substance, the group becomes incompetent.
3. *Reality denying*. Outside problem "shocks" are covered up or denied rather than dealt with, and whistle-blowers who point them out suppressed or fired. No competence learning occurs.
4. *Political*. Members are too busy with internal power struggles to attend to outside problems, so the group handles them poorly.

5. *Negatively driven*. Socialization works by applying negative sanctions, so avoiding them becomes the key to advancement. Leaders practice non-failure not success. Yet budget cuts and monitoring are no substitute for incentives and a positive vision. Negatively driven citizens become submissive or apathetic.

The above are maladaptive because they try to use social means (rules, image, conformity, politics, sanctions) to achieve competence goals. This does not work, as the goals of a level require operation at that level. The solution to incompetence is competence, which is created not by more rules and regulations but by rule breaking, image deviations, criticism and citizen incentives. The challenge of social design is not satisfying the requirements of one environment.

## 5.16 HIGHER SOCIAL SYSTEMS

The vertical ellipses of Figure 5.4 mean that a social environment can be contained by another, e.g. as many people can form into a company so many companies can form into a stock market. The company is a social group to its members, and the stock market is a social group to its members, with both social systems adding value to members. Companies reward employees with pay and stock markets reward companies with share prices that allow public investment. Equally, both environments place synergy requirements on members: companies ask employees not to steal their product value (stock) and stock markets ask companies not to steal their product value (ratings) by falsely reporting profits.

Rule 3 can be universalized to the multi-environment case where $S_1$ contains $S_{2\ldots}$:

**Rule 3'a:** *If* $\{\{SU_1(a_i) \geq SU_1(a_j)$ *or* $SU_2(a_i) \geq SU_2(a_j) \ldots\}$ *and* $IU(a_i) > IU(a_j)\}$ *then prefer* $a_i$ *to* $a_j$

**In words:** *Choose acts that do not significantly harm higher environments but benefit oneself.*

*OR*

**Rule 3'b:** *If* $\{ IU(a_i) \geq IU(a_j)$ *and* $\{S_1U(a_i) > S_1U(a_j)$ *or* $(S_2U(a_i) > S_2U(a_j)...\}\}$ *then prefer* $a_i$ *to* $a_j$ < >

**In words:** *Choose acts that do not significantly harm oneself but benefit higher environments.*

It follows that a social dilemma solved for one social system can reappear if a higher one is formed. When the same social dilemmas operate at higher social levels, "new" problems arise; e.g. the Enron debacle, with estimated losses of over $60 billion, occurred when Enron executives cheated the stock market by reporting false profits to raise their stock price. Other companies laid off staff to "compete" with Enron's imaginary profits of over 80%. Within the stock market social system, Enron defected on the rule by which the stock market creates synergy, as if everyone made false claims, no one would invest. If false reporting had not been illegal before Enron, it would have to have been made so, for the stock market to survive. The stock market, a higher social system, had to act against the Enron cheats or collapse itself.

Confusion arises because advocates of Rule 1 or Rule 2 cherry-pick cases to support doctrinal positions. In this model, competition is good but not always, e.g. if business did operate by pure competition (Rule 1), then Enron's innovative methods of obtaining value in the stock market environment would have been a competitive advantage, as would have been their paying zero U.S. tax for seven years. So the business maxim "greed is good" does *not* apply to defecting on a social contract. Cheating one's colleagues is not "competitive advantage", as its bottom line is a loss of value for the whole society.

Higher social system defections involve hypocrisy, e.g. Enron bosses hypocritically asked workers to serve the Enron company environment while themselves cheating Enron's social environment, the stock market. Gangs like the Mafia have a similar hypocrisy, demanding strict loyalty and service within their community,

while as a group pillaging the community at large. In general, a social rule that does not apply at every level creates an inconsistency that must eventually be resolved, e.g. it is inconsistent for member states that do not give their citizens democratic rights to have democratic rights themselves within the U.N. assembly.

Wildlife conservation in Africa also illustrates the importance of acting at the right social system level. Poaching is a classic tragedy of the commons case, yet public ownership has generally been a disaster for conservation in third world countries (Ridley, 1996). Under nationalization, the government could not stop locals poaching the large animals that damaged their crops. The trend was only reversed when wild-life titles were "privatized" to local communities, like the Campfire programme, where hunters purchase the right to kill game from local villages (Ridley, 1996, p.236). When the village "owned" the wild animals, it looked after its resources, prevented members poaching and wildlife numbers grew.

In contrast, whales roam international oceans not owned by any country, so we could hunt them to extinction. If a global humanity owned the whales, it would be as foolish to hunt them to extinction as for a farmer to kill all the cows in his herd. So as nations hold local power, perhaps rights should be "privatized" to nations, who would then physically prevent whale poaching in their zones.

Rule 3 can be idealized to define categorically good acts as those that give value "all the way up", not just for oneself, but for the community, for humanity, and even the planet we live upon. The principle that there are levels of "good" was made clear in the Nuremburg trials—where citizens following national laws were convicted of "crimes against humanity", i.e. held to a higher standard of right and wrong.

If social environments are within a world environment, is the higher good to serve the world not society? Conversely, if Rule 1 is to meet the world's requirements, is that then the higher good? Obeying Rule 1 is not following a higher good. It is individuals seeking their own gain not that of their environment. If the latter occurs, they are unconscious of it; e.g. innumerable animals have lived, fought, re-

produced and died with no idea at all of the natural evolution of life on earth. To unconsciously contribute, as unknowing grist to a mill, is not acting for a higher good.

However *consciously* contributing to a higher environment, as Rule 2 proposes for social environments, is acting for a higher good. This is why Rule 2 is the ethical rule. So by the Rule 3 extensions, the highest good is to recognize the requirements of the highest level environment while still living in lower ones. The pragmatic ideal is to serve the highest environment one can conceive while surviving the demands of lower environments.

## 5.17 THE GOLDEN RULES

The current prosperity of humanity has been linked to its social evolution which in turn traces back to ethical advances like:

"*Do unto others as you would they do unto you*"

This golden rule has been expressed in many different cultures and contexts:

1. Rabbi Hillel's sum of all rules: "*If you don't like it done to you, don't do it to others*".
2. Kant's proposal: "*Act only on that maxim by which you can at the same time will that it become a universal law*", i.e. *if everyone does it, is it still successful?*
3. Pareto's optimality principle: "*Good actions benefit at least one other and do no harm.*"
4. Rawl's "*veil of ignorance*" requires state justice to be "blind" to individual needs.
5. Harsanyi's approach rules out immoral or anti-social acts from consideration (Harsanyi, 1988).

These and other forms suggest a solid universal social principle that is equally applicable to information technology (Siponen & Vartiainen, 2002). Anti-social acts fail all golden rule tests; e.g. Hillel rejects stealing as one does not wish to be stolen from, Kant finds it wrong as if everyone does it, it does not work, and Pareto finds it harms another. Rawls from behind his veil of ignorance cannot advocate it without knowing who is stealing from whom, and Harsanyi finds stealing an anti-social act. Rule 3 rejects stealing because overall it is a social loss, e.g. when a wallet is stolen there is not just the money lost but also disruptive losses like renewing credit cards.

All golden rules sit above the individual economics of game theory. Kant distinguished categorical imperatives from hypothetical ones, i.e. the rule is *not* "Do unto others *so* they will do likewise unto you". Such "deals" are merely *instruments* to individual benefit. Kant's imperative is to do what is *categorically* the right thing to do, regardless of the outcome for oneself. He advocates operating on a higher level than one's small self.

The golden rules ask free people to act beyond their own interests, hypothetically to flip every social interaction equation to see if it still works the other way, to stand in the shoes of others and to think of the society as a whole.

This model frames this call as not just to "goodness" but also to social productivity. The ethic of serving a community is logical because Rule 2 is just Rule 1 applied to the social instead of the individual unit. Higher social levels are more productive because synergy works. Ethics is just *pragmatics at a community level*.

Using the golden rules, the ship of human society has navigated a middle way between the evolutionary dead-ends of endless tribal conflict and mindless social conformity. This struggle began thousands of years ago, at the dawn of the agricultural revolution, when "civilized" farmers cultivating the land battled "barbarian" hunter-gatherers, whose essential code was:

"*Take what you can and give nothing back.*"

Humanity is fortunate that reason pulled us out of the dark ages, but reason was too fragile to get us across the original zero-sum barrier of the agricultural revolution (Whitworth, Van de Walle, & Turoff, 2000). That needed faith and religion. The first religion seems to have been started over six thousand years ago by the first Zoroaster in Persia[8]. He called upon the *people of the herd* to hear the *immortal shining ones,* do right not wrong and to reject the *followers of the lie*. It is easy to forget how huge that first step was. Today all golden rules address free people, not mindless followers, so what is advocated is not just goodness but *free-goodness*.

## 5.18 FREE-GOODNESS

All golden rules advocate free-goodness, doing right for no personal benefit. Another way to say this is that the best things are done for no particular reason. In socio-technical systems, free people doing what they want to help others they have never met, for no particular reason.

Neither Rule 1 (compete for yourself) nor its Rule 3a extension (compete by the rules) explain why online experts help others with hardware problems. People routinely help others in physical society too, e.g. giving lost visitors directions even though they probably never see them again. Only in socio-technology, is this the basis of the social system. The theory base is the philanthropy Rule 3b, that people with *enough* will give back to the community, even if there is *no* benefit to themselves.

Rule 3b is not that individuals in markets *unconsciously* help society. It reverses the logic of Rule 3a, to say that if individual needs are met, a positive urge to social value remains. Granted that people are self-motivated, it says they are also socially-motivated, to help others if they can, e.g. BitTorrent system users

8. See https://en.wikipedia.org/wiki/Zoroaster

help each other download large files although the community lets them download and leave. Yet it survives because many do not. Even in a community of people downloading free, and perhaps pirated, web content, people help each other.

In the past, *heroes* in philosophy, art, science, music, politics and other fields sacrificed their lives, often literally, to create works whose value was often only realized after their death, e.g. Tolkien, Socrates, Van Gogh, etc. Today technology lets us all be *small heroes*, to do good for no reason *now*. We can add a page to Wikipedia, tweet against injustice or start a free web site like Kickstarter[9], a funding platform for creative projects. The same technology that magnifies social negativity like spam can magnify positive social acts. It lets many small heroes combine, until they are more than any past hero, e.g. no one person could write Wikipedia. Good acts can cascade too, as the movie *Pay it Forward* suggests.

Socio-technology, as a new social system design, may arise from the *digital divide*, that only the better off have computers. If social evolution requires personal evolution, online communities may work because they have more social health. If so, online communities predict our social future.

Whatever the reason, systems like Wikipedia threw themselves on the goodwill of their citizens and not only survived but prospered. That virtue supported by technology is productive is an important social discovery (Benkler & Nissenbaum, 2006). Socio-technical systems succeed not by technical efficiency but social efficiency. The social invitation to do small selfless acts of community service was taken up. Socio-technology enables social synergy, but if people did not follow Rule 3b it could not work.

In history, societies that enforced order got synergy, e.g. Egypt's pyramids were a synergy product. We know today that markets can incentivize individuals to synergy given contextual legal systems. What we did not know, until the Internet arrived, was that people not subject to coercion, nor enticed by incentives, can

9. http://www.kickstarter.com/

freely synergize. We knew people could be forced to be good citizens, or enticed by reward or punishment to be so, but not that they could *freely* be so.

The socio-technical systems that are today transforming the Web differ in significant ways from current physical society (Kolbitsch & Maurer, 2006). A community level focus gives social systems that are decentralized, transparent and participative. Initiatives like SETI, and FLOSS (Free, Libre, Open Source Software), community sites like SourceForge and FreshMeat, deny all forms of social control, whether of acts (repression) or of information (propaganda), and *believe* quite simply that free people will do the right thing. The Creative Commons is founded on the principle that people will freely make their creative work public (synergy) if receivers do not copyright or sell it (defect).

This is not communism, because individuals are free to express themselves without social control. It is not socialism, because individuals can take from the community and not give back. It is not anarchy, as there are anti-social defenses to oppose disorder. It is not altruism, as no-one has to sacrifice for the community. It is not capitalism because the primary goal is not personal profit but community service. Socio-technical systems illustrate free-goodness, *a new social system design,* based on freedom, service, legitimacy and transparency. They are the online evidence of the social future of humanity today.

## 5.19 NEW BUSINESS MODELS

The advent of free-good citizens synergizing without payment or central control is reflected in new online business models, necessary if socio-technology supports not just online society but *all* society. The historical trend is to stabilize synergy for larger groups, from tribes to mega-states like Europe, India, China and America (Diamond, 1998). Each stage of this social evolution needs more complex social mechanisms, which today's technologies allow.

The social evolution happening today online has led to business models based on community service rather than personal profit. Current research on *trust* generally frames the problem as how to get people's trust to make a business sale. Yet asking how to trick people into buying products is viewing the issue at the wrong level. Even if tricking customers into buying was possible, those susceptible would soon go bankrupt and so no more be customers. No business can survive on *stupid* customers. Marketing that tries to trick customers to buy what they do not need is self-defeating.

Advertising works on the personal level, but on the community level it does not pay communities to let product propaganda lie to citizens. It is claimed that advertising funds TV, but currently it is killing it, as advertisements overpowering programs just turns people to other options like the Internet. As people get more technology savvy, they mute the ads or flip to other channels, just as they set their browsers to block web popups.

The profit motive online is illustrated by the dot.com bubble of the late 1190s, when greedy online business start-ups found that online customers were not as gullible as supposed. It seems obvious to say, but *that is why they are still customers*. A few con men can succeed, by the synergy of society, but con-societies cannot succeed.

Community level business models change the question, from how to trick customers to how to synergize with them. Instead of seeing customers as competitors for profit, to be kept in the dark, they are partners in value creation. Systems like eBay illustrate that customers are not the competition, but part of the business.

The Chinese philosopher Lao Tse, founder of Taoism, best stated how to govern a community:

"*One should govern a state as a cook fries small fish, that is without scaling or cleaning them.*"

His advice was to "cook" a state without interfering with its citizen parts directly, using community level not personal level acts. Yet current management barely even works on the personal level, based as it is on information forms, regulations and organization structures. Managing people like pieces on a chess board is working at the wrong level[10]. Treating people like numbers just lowers morale, increases apathy and sometimes gives paralyzing strikes, i.e. causes the system to fail entirely[11].

Seeing people as people not information on a spreadsheet or wallboard is the difference between management and leadership. Leaders practice *management by walking about* and talking to staff, rather than sitting in their office analyzing reports. The TV program Undercover Boss is successful because most CEOs dont do this. Yet acommunity inspired will always outperform individuals seeking profit e.g. wartime factory production is double or more that of peacetime. Semler details how his company activated human level performance productivity, based on ideals like transparency (Semler, 1989). Yet while people want leaders who inspire not managers who manipulate, most companies seem to want to employ slaves. It is counter-intuitive to zero-sum thinkers that giving workers freedom can increase productivity, but Google gives its employees half a day a week to work on personal projects for just that reason. Increasingly, employees succeed despite their managers not because of them, as the comic strip Dilbert illustrates[12].

10. A university provost once said to me: *Managing faculty is like herding cats*. If the university wanted to herd, they should have hired sheep. If they hired cats, they should put out milk. Hiring cats and then trying to herd them is stupidity.

11. Management efficiency is illustrated by a man with a donkey who worked hauling goods but had trouble making ends meet. A friend who found him drowning his sorrows in the pub advised him to cut his operating costs to get more profit. So instead of ten carrots a day he gave his donkey only nine. Finding it did more work, he did the same the next day, and so on. A week later he was again found drowning his sorrows. When asked why, he said, "*It was all going so well. Had he not died, I would have got him working for no carrots at all!*"

12. See http://www.dilbert.com/

## 5.20 A NEW SOCIAL FORM

Game theory's specification of "rational" decision making directs many decision strategies in business, economics and politics, but fails utterly to explain how humanity crossed the zero-sum barrier to achieve the synergies of civilization. *Homo-economicus* who seeks self-interest has joined with *homo-sociologicus* who seeks community good. The first seems "selfish" and the latter "good" but both rules are essentially pragmatic. Community service is as rational as self-service.

In the social environment model, citizens combine these rules by anchoring one and applying the other. Anchoring social good then seeking self-gain explains the highly successful market trade systems of the last century, where individuals sought profit under social good laws. Anchoring individual good then seeking community gain explains the highly successful socio-technical systems of this century, where contented people help a community for no personal reason.

Technology is the effect magnifier that is forcing humanity to resolve its social choices, e.g. last century, nuclear technology magnified the power of war until humanity had a choice: to destroy itself by nuclear holocaust or not. The decision *not* to destroy itself was made *by our choice,* not by the technology that forced us to make it. Technology just upped the ante. Nuclear weapons exposed the illusion of world domination and highlighted the folly of mutually assured destruction (MAD), but people still had to make the choice.

Likewise the industrial revolution the century before brought the feudalism myth to a head, as serfs became factory slaves instead of farm slaves. This century, information technology challenges the illusion of profit, the myth of getting something for nothing[13]. The profit motive is to get more for less, so the obvious

13. It is a myth because in this world there is no such thing as something for nothing. By the second law of thermodynamics, increasing order requires effort. The rich version of the myth is that one is entitled perpetually to skim a percentage from the labour of others. The poor version is that one is entitled to be supported by others forever. The alternate view presented here is that we should give back to the community that gives to us.

ideal is to get something for nothing at all. Enron, World Corp and the credit meltdown banks all sought perpetual profit. Yet just as perpetual motion contradicts physical laws, so perpetual profit contradicts social laws—everyone taking from everyone else cannot succeed. We scorn medieval myths of a physical elixir of perpetual youth, but today seek equally impossible algorithms of perpetual profit, as pyramid profit schemes show.

People using technology to pursue profit increase the chances of a global social collapse, but how can a society based on personal profit call them to ethics? The profit motive is ending its useful life. The profit idea allowed higher performance by replacing physical results with data, as sales and salaries replaced harvests, but today it distorts rather than improves. The profit motive turns health systems into sickness systems generating money from illness. The academic search for truth becomes publishing for promotion and grants. Taxation systems become tax evasion systems, as companies pay no tax, traffic regulation becomes revenue gathering, and the list goes on. What was progress is now an obstacle to it. The solution is not more of the same, whether budget cuts or bailouts, but, as always, to evolve a new system. In this view, that system will be based on good people not just good calculations and data.

In our social evolution, the *dictatorship* of kings or emperors was an advance over hunter-gathering. It worked until it was replaced by *republicanism*, government by an elite like the Roman Senate. This worked until replaced by *feudalism* in the middle ages. That worked until the industrial revolution gave us capitalism and communism, both claiming to give power to the people. Yet both are also based on profit (for the individual or community). The profit motive as a social driving force has served us well, but it must give way to *a new social form*. Tokens like money that focused people on productivity now distract them from synergy. If one offers peanuts one gets monkeys, but if one offers honey one gets wasps.

The Internet invention is to offer people nothing personal at all, but to go directly to community synergy, without passing out any money. Instead of bribing people to synergize with money, just ask them to do it for us. The technology shows the benefits of that virtue immediately, so it is obvious that it works. There is no need for propaganda and marketing to convince people that something is good if it manifestly is. Systems with no reward tokens have nothing to steal and with no central control there is no centre to hijack. If anti-social acts are transparent then people abusing the system are exposed, shamed and can be banned. In the light of public scrutiny, defections shrivel like ghosts in the sunlight. Right now, the physical world has no society not driven by profit, but the virtual world says that it could. This is not just a theory, but a social practice already tried and proven.

Yet make no mistake, the socio-technical vision of the future is radical beyond capitalism and communism. Rule 3b states that making a profit is OK until you get enough, then profit must give way to service. It asks first "*Do you have enough?*" and then if the answer is yes, "*What have you given back?*" In this view, the super-rich who continue to take but give nothing back are social bludgers, because today social synergy is the cause of wealth. In the future, the rich list will be a list of shame not envy.

If the profit motive is entrenched, how can Rule 3b supersede Rule 3a? The answer is that when our social health reaches a certain level, it will seem the obvious thing to do. No revolutions are necessary, only evolutions. One day, a physical community will just ask its people to synergize, not by bribery or coercion, but by common sense, and as online, they will respond.

The beginning of this new vision is seen in the rise of independent voters, who sit between the right and left wings of politics and increasingly decide elections. Tradition calls them swinging voters, but really they are free voters who decide each issue on its merits, not on a formal doctrinal rule. Members of this *free*

*choice party* have no conventions, no rules, no formats and no utopias. For them, *every political vote is a conscience vote*. They believe that:

*a) I am free*. I am not a slave to anyone, however righteous or powerful they seem.

*b) I am good*. I seek the benefit of others as well as myself.

*c) I am a citizen*. I am not alone, but part of a larger group.

Free-good citizens reject personal profit and community control as evolutionary dead-ends that were tried and failed. They hold that each person should freely do as they think best, and let others do the same. They *believe in us*, that some people may err, but most will not. This faith in humanity as a whole is the basis of social evolutions, e.g. science is the belief that if people talk openly, the truth will out. Likewise the answer to evil is transparency, as what plot is so secret that someone somewhere in the community does not know of it? The negative view is that without control people will revert to savages, but the positive view is that people left alone will naturally synergize. Free-good citizens don't *need* a boss. The proof of this today is no logical treatise, no Word of God or issued doctrine, but in the online experience itself. *The manifesto of the Arab Spring lies in the social structure of the Internet itself*.

This social structure is sustained by people doing what they *don't* have to, not by any formal rule or theory. Humanity is tapping the social goodness of small heroes to transform itself at the informal personal level not the formal informational level. If freedom is the price of individual evolution and goodness the price of social evolution, the socio-technical experiment of doing both has worked. Beyond the profit of self and the virtue of community service is choosing to do things for no reason but that they seem like a good idea. The greatest advantages arise when we stop seeking advantages and just do what we think is right.

## 5.21 GLOSSARY

Here is a summary of the key terms of this model:

1. *Rule 1*. Competing self-interested individuals evolve competencies (individual evolution).
2. *Rule 2*. Cooperating socialized individuals evolve social synergies (social evolution).
3. *Synergy*. The difference between what individuals produce as a social unit vs. independently.
4. *Anti-social acts*. Taking individual benefit from society without contributing to its synergy.
5. *Social dilemmas*. When individual outcomes (Rule 1) contradict social outcomes (Rule 2).
6. *Social instability*. When social systems generating synergy are unstable to anti-social chain reactions.
7. *Social order*. That all members of a social group act as one.
8. *Social freedom*. That members of a social group are free to act from their own choice.
9. *Social hijack*. When leaders hijack society for their own ends, and maintain control by:
    - *Repression:* Coerce people to not follow Rule 1.
    - *Brainwashing*: Convince people to follow Rule 2.
10. *Social inventions*. New ways to get synergy and competence:
    - *Justice*: Punish unfair anti-social interactions by laws, police, and sanctions.
    - *Democracy*: The group periodically changes its leaders by freely voting.

- *Legitimacy*: The allocation of "rights" that are:
  - *Fair*. Individual consequences match the individual contribution (Rule 1).
  - *In the public good*: Benefit society as a whole (Rule 2).

11. *The golden rules*. That individuals can freely choose to serve an environment above themselves.
12. *Social environment model*. That social units are environments within environments.
13. *Rule 3*. That citizens combine Rules 1 and 2 by anchoring one and applying the other:
    - *Rule 3a*. If social laws are not broken, compete for individual advantage (markets).
    - *Rule 3b*. If one has free time or money, give to others in the community (service).
14. *Rule 3'*. Extends Rule 3 to apply to nested social structures.
15. *Rule merging*. That communism (Rule 2) and capitalism (Rule 1) will merge into a hybrid.
16. *Social health*. The percentage of individuals in a community that *freely* support social synergy.
17. *Social token*. How a society distributes the value it generates, e.g. money, grades or stock ratings
18. *Social inflation*. When social tokens lose external value because the group does not satisfy *its* environment's needs.
19. *External rectification*. When a society's collective incompetence has consequences for its citizens.

## 5.22 DISCUSSION QUESTIONS

The following questions are designed to encourage thinking on the chapter and exploring socio-technical cases from the Internet. If you are reading this chapter in a class - either at university or commercial – the questions might be discussed in class first, and then students can choose questions to research in pairs and report back to the next class.

1. What is social synergy? How do communities encourage synergy? How do they prevent its destruction? How do trust and synergy relate? Give physical and electronic examples.
2. Give five examples of defections in ordinary life. What happens to a community if everyone defects? Give five online examples of defections, and for two specify how technology lowers the defection rate.
3. Would you prefer to be a middle class citizen now or a lord three hundred years ago? Research history to compare factors like diet, health, clothes, leisure, travel, etc. Where did the lord's wealth mainly come from? Where does the power of your salary to buy many things come from today? How does the principle apply online?
4. What is a social dilemma? Give three physical examples from your experience. Why cannot individuals solve them? How are they solved? Give three online social dilemmas. How can they be solved? Relate this to socio-technical design.
5. What happens if one suggests things in a group? Conversely, what happens if no-one in a group suggests anything? How can groups manage this dilemma? Answer the same questions for volunteering. Give examples in both cases from both offline and online.
6. What percentage of online users are *lurkers* who look but do not post? Go to a popular board you have not used before. What stops you contributing? Add something anyway. How could the board increase participation?

7. Is ethics idealism or pragmatism? Explain the statement: Personal ethics is community pragmatics. How can STS design affect ethics?
8. Analyze the case of a thief who steals a wallet and is not caught. List the thief gains and the victim losses to get the net community result. What if everyone in a community steals? Generalize to the online case where spam "steals" a few seconds of your time. How does this differ from an offline theft?
9. Why is synergy important for larger communities and especially important for socio-technical systems? How can technology help increase synergy? Report the current estimated sizes of popular socio-technical systems. Clarify what is exchanged, who interacts, the synergy and the defections.
10. Look at the objects you use every day. How many could you make? How many are even produced in your country? How hard would it be for you to make them? Compare the cost it would take you to make say a simple table with how much you pay for it. Relate this to social synergy.
11. Discuss whether people are rational actors acting in natural self-interest. Give physical examples of common acts that are irrational, i.e. done knowing they will cause suffering, or knowing that they will not give any benefit. In general, when are people irrational? How does this affect STS design?
12. Describe the volunteer dilemma. Is it that people will not volunteer or that they will? Look at the extreme case of people volunteering to go war. Why did not only many Japanese but also those in other armies volunteer for kamikaze type missions? Explain this in terms of Rule 2. How important is volunteering in online communities? How can technology support it?
13. Describe how online babysitting exchange cooperatives work. Based on a game theory matrix, how would one act if one followed Rule 1? How about Rule 2? How do people actually act? How does the technology affect this?
14. Social networks like Facebook are about friends but what is a friend? In anyone who helps you a friend? Define a friend in synergy terms. Does

friendship imply trust? Is it a one-way or two-way thing? If you "friend" someone on Facebook, are they really a friend? Explain the difference referring to information and human levels.

15. Consider how one person cheating causes others to cheat, e.g. in sports. Draw a diagram to show how social defections cumulate as each one reduces the probability that cooperation will give synergy benefits. How do social defenses alter these percentages? Use an online case to illustrate.
16. What is social order? Explain its strengths and weaknesses, with examples. Is social order possible on the Internet? Discuss the success or not of attempts by countries like China and Iran to control the use of the Internet by their citizens. Mention proxy software designed to thwart that control.
17. What is social hijack? Are all dictators social hijackers? Give physical examples of past communities ruled against their will that eventually rebelled. Can the same occur in online communities? How does STS design affect this?
18. Can a social system exist if it is not a physical "thing"? If so, are new social designs like democracy social inventions? What then was invented? Make a list of the social inventions of physical society over the last two thousand years. How does socio-technology add to that list?
19. In the middle ages, whether in China, England or Russia, democracy was not only unthinkable but also impossible. Why? What changed in the last thousand years to make it possible? How is the same change factor allowing new social designs to develop on the Internet?
20. Describe some well-known peasant revolts of the past that were successfully put down. If the Arab Spring is the same, but based on modern socio-technology, why is it harder to put down? Discuss how the information revolution is changing how Arab states have been governed for centuries.

21. Describe the communism vs. capitalism ideological battle that dominated the last century in Rule 1 and 2 terms. What is the result today, i.e. which social design won? Is China purely communist? Is America purely capitalist? How would you describe the successful social designs of this century?
22. What exactly is democracy? Is the Democratic People's Republic of Korea democratic? Were the Greeks of Athens democratic? Is the USA democratic? Are there any online democracies? Is Wikipedia a democracy? Describe online systems that have at least some democratic features.
23. The Enron collapse lost millions but the 2007–2012 credit meltdown lost billions. Explain how one was an ethical failure and the other a competence failure. What happened to the perpetrators in both cases? Can such things happen online? Describe what would happen to say Wikipedia if it became a. Corrupt or b. Incompetent. Can socio-technology reduce the likelihood of such "crashes"?
24. Do the various golden rules of history follow Rule 1 or Rule 2? How do these golden rules, which began thousands of years ago, affect the design of information technology today?
25. What is social transparency? Is it making everything visible to all? How does it relate to privacy? How can STS support both transparency and privacy?
26. If an ant in an ant colony serves the community, are ants ethically good? Are people forced to serve a community also good? Explain why without freedom there is no goodness. Relate to the issue of whether smart software should take over human choice or whether software should always leave the ethical choices to people.

## YOUR NOTES AND THOUGHTS ON CHAPTER 5

Record your notes and thoughts on this chapter. If you want to share these thoughts with others online, go to the bottom of the page at: http://www.interaction-design.org/books/the_social_design_of_technical_systems_2nd_ed/the_social_environment_model.html

**NOTES:**

# CHAPTER 6

# Online Rights

"*Communities prosper by granting citizens rights.*"

This chapter relates technology to basic social axioms like:

1. *Ownership*. To specify object rights to reduce conflicts.
2. *Freedom*. To own oneself, not to be a slave.
3. *Fairness*. That social consequences reflect action contributions (Rawls, 2001)[1].
4. *Privacy*. To control personal information released to others.
5. *Transparency*. A citizen's right to know of their governance.

The aim is to specify social requirements in information terms, to develop socio-technical standards for online social interaction.

1. If A's acts cause B's loss, it is unfair to B. If A's acts cause B's gain, it is unfair to A.

## 6.1 ACCESS CONTROL

In computing, decision support systems recommend decisions, access control systems permit them, and control systems implement them. Access control began with multi-user computing as users sharing the same system came into conflict (Karp et al., 2009). Traditional access control systems use a subject by object access permission matrix to allocate rights (Lampson, 1969). As computing evolved, the logic offered local access control for distributed systems and roles for many person systems. With these variants, the matrix approach has worked for military (Department of Defense, 1985), commercial (Clark & Wilson, 1987), organizational (Ferraiolo & Kuhn, 2004), distributed (Freudenthal et al., 2002), peer-to-peer (Cohen, 2003) and grid environment (Thompson et al., 1999) applications.

Today, access control in *social networks* (SNs) is more about access than control. The permission matrix for friend interactions increases geometrically not linearly, with group size, so for hundreds of millions of people the possible connections are astronomical. Each person may add hundreds or thousands of photos or comments a year, and they want the sort of domain control previously reserved for system administrators. Social networkers want *local control*, not just to read, write and execute (Ahmad & Whitworth, 2011), but to define their own social structure without a central permission (Sanders & McCormick, 1993): e.g. to restrict a photo to family or friends. Social networks vastly increase ACS complexity, as millions of users want rights to billions of resources. STS is the perfect storm for the traditional ship of access control.

The current rules of social network interaction are based on designer intuitions rather than formal models, so they vary between systems and over time, with public outrage the only check. There is no agreed scheme for allocating permissions to create, edit, delete or view object entities, let alone manage roles. The aim here is to fill that gap, to develop a socio-technical access control model that is legitimate, efficient, consistent and understandable.

## 6.2 RIGHTS

Communities, by norms, laws or culture, grant citizens *rights,* or social permissions to act. Rights reduce physical conflict, as parties who agree on rights do not have to fight. The conflict moves from the physical level to the informational or legal level[2]. Physical society expresses rights in terms of ownership (Freeden, 1991), so specifying who owns what online can specify rights in a way that designers can support and users can understand (Rose, 2000). This does not mechanize online interaction, as rights are choices not obligations; e.g. the right to sue does not force us to sue. Legitimate access defines what online actors *can* do, not what they *must* do.

In traditional computing people are software *users,* as if software were a drug they depended on. Socio-technology calls people *actors* because they are not just part of the software. Users cannot switch software but actors can. As sales assistants can see *a sale* or a *customer*, so IT designers can choose to see a user or an actor. One is the human level and the other the information level.

An online actor is any party in a social interaction that can act independently of input, i.e. it is able to act not just react. Actors initiate acts based on internal choice or autonomy[3]. A program that only responds to input has no autonomy, so is not an actor[4].

A person is an actor whose ego-self can be held to account. A citizen is a person a community holds to account[5]. To hold people to account is the basis of all social interaction[6]. Currently, only people can both initiate acts and be accountable for them.

2. Personal acts between people is the level after that, when people forget rules and *trust* each other.

3. From the Greek *autos* 'self' and *nomos* 'law,' i.e. a system that makes its own laws. Light a fire under a table and it burns, light a fire under a dog and it runs away, light a fire under a man and he cooks his dinner.

4. A program can however be an agent.

5. Accountability assumes choice at some point, e.g. so a drunk can be fined because he earlier chose to drink. A drug addict cannot stop now but once he could. If they are no longer accountable, the community can take control of them.

6. Community justice and law began with revenge, where people personally held others to account. Revenge is personal justice.

While philosophers argue over free will, all communities find their citizens accountable and govern accordingly, e.g. criminals are punished and mentally incompetent are put into care. Communities hold citizens to account for the effects of their acts, not just on themselves but alsoon others[7]. Accountability is the social key without which communities fail. It only applies to people, e.g. in car accidents the driver is accountable not the car[8]. Likewise online, the company that writes an installation program is accountable for it.

Rights arise as the formal expression of legitimacy concepts like fairness. In physical communities, police and courts direct citizens to follow laws that grant rights. Online, the same applies, but now the code is law, police, judge and jury. The following derives informational rights from community requirements stated on the personal level.

In information terms, a right is a community permission given to an actor (A) applying an operation (O) to an entity (E):

$$Right = (Actor, Entity, Operation) = (A, E, O)$$

Rights can be stored as (Actor, Entity, Operation) triplets, where an actor represents an accountable peson, an entity is any object, actor[9] or right, and an operation is any one available to the entity. A right transmitted or stored is often called a permission.

7. A citizen is a person in a community. A foreigner is not a citizen but still an accountable person. A citizen who is not accountable is a criminal locked up.

8. A company is an information entity with no ego-self that cannot be accountable. Punishing a company by declaring it bankrupt just lets its owners start another company to do the same again. Treating companies as people in the law was the great ethical error of the last century. It underlies most of the scams of the wealthy.

9. An actor can act on itself, even to delete itself, as a person commits suicide.

## 6.3 A RIGHTS SPECIFICATION

Socio-technical systems can be modeled as data *entities* and program *operations* as follows:

1. ***Entities***. Stored as static information.
   - *Actor*. A data entity representing a social participant[10].
     - *Persona*. Represents an offline person or group.
     - *Group*. A set of personae acting as one[11].
     - *Agent*. A program representing a persona.
   - *Object*. Conveys information and meaning.
     - *Item*. A simple object with no dependents, e.g. a bulletin board post.
     - *Space*. A complex object with dependents, e.g. a bulletin board thread.
   - *Right*. A system permission for an actor to operate on an entity.
     - *Simple rights*. Rights to act on object or actor entities.
     - *Meta-rights*. A right to act on a right, e.g. the right to delegate a right.
     - *Role*. A variable right (a set of rights).
2. ***Operations***. Stored as programs or methods processing entities.
   - *Null operations* do not change the target entity, e.g. view an object[12], enter a space.

10. An actor need not be a person, e.g. a program can be an agent.

11. Online and offline are different worlds by their base architecture. An offline group is physical people who act as one. Groups can also form groups; e.g. the stock market is a group of groups (companies). An offline group can have one online persona; e.g. a company registered on Facebook. An online group is a personae set that acts as one. See 6.13 for how a system can define this.

12. View is null at the informational level but not at the psychological level, as looking at someone affects them.

- *Use operations* change the target in some way, e.g. edit, create.
- *Communication operations* transfer data from sender(s) to receiver(s), e.g. send.
- *Social operations* change a right or role, e.g. delegate.

Links are discussed elsewhere (Whitworth & Bieber, 2002). The following outlines general principles for online social interactionss in any socio-technical system. While many of these may seem obvious, recall that to software nothing is obvious and everything must be specified.

## 6.4 THE SYSTEM

The information system itself is the first entity, owned by the system administrator (SA), who is the first actor. A tyrant SA might alter posts or votes by whim, or a benevolent one might follow Plato's best form of rule and give citizens rights. Currently, no online system elects its system administrator; e.g. even Wikipedia is not a democracy. Yet as today's online kings and emperors die, some form of succession will have to be worked out.

An ACS controls the system at the informational level. If it is *not* to be in charge, it must allocate all system rights to people who are, giving the first ACS operational principle:

**P1**. *All non-null entity rights are allocated to actors at all times.*

It is not necessary to allocate null rights that have no effect. This principle implies that every entity is ultimately owned by a person. If it is not so, an access control system must at some point respond to an access request from itself, which is impossible. An information system has no self to act socially. That all rights are always set means they are not added or deleted, but *allocated* and re-allocated.

## 6.5 PERSONAE

An online persona represents an offline actor, e.g. an avatar, mail profile, wall or channel can represent an offline person, group or organization. An online persona is activated by a logon operation, which equates it to the offline party. An online computer agent can act for a group, as installation software does for a company, but social acts must ultimately trace back to people and online is no different[13]. If an installation misleads, we sue the company directors, not the software[14].

*Who owns a persona?* Open systems let people self-register, to create their personae. If freedom applies online, one should own one's online self. Yet many systems do not permit this. Can you delete a Wikipedia or Wordpress profile?[15] The requirement of freedom gives the ACS principle:

**P2.** *A persona should be owned by itself.*

Some complexities are that a persona can be:

1. *Abandoned*. HotMail accounts inactive for over 90 days are permanently deleted, i.e. if not activated, they "die."
2. *Transferred*. One can permanently pass a persona to another, along with its reputation[16].
3. *Delegated*. One can ask an agent to act on your behalf, e.g. by a proxy vote.
4. *Orphaned*. If the person behind a persona dies, their will is physically respected, but online programs act as if death does not exist, e.g. one can get an eerie Facebook message from a person after going to his funeral. In a few decades Facebook will represent millions of obituaries, so we need online wills. Death is a social reality online and offline.

13. Registering by a nickname online instead of one's "real" name denies accountability offline but not online; e.g. a banned EBay seller name loses its online reputation.

14. A person who acts as an agent can still be held accountable; e.g. if told to shoot someone and does do.

15. See how to permanently delete your account on popular web sites here: http://www.smashingmagazine.com/2010/06/11/how-to-permanently-delete-your-account-on-popular-websites/

16. In the movie *The Princess Bride*, the Dread Pirate Robert persona was passed on, so the idea is not new.

Table 6.1 shows a summary of persona access rights.

| Persona | *View* | *Delete* | *Edit* | *Ban* | *Create* |
|---|---|---|---|---|---|
| *System Admin* | √ | | | √ | √ |
| *Owner* | √ | √ | √ | | √[1] |

TABLE 6.1: Persona access rights [1]Delegated by the SA.

## 6.6 OBJECT ENTITIES

Object entities convey meaning by evoking cognitive processing; e.g. a family photo.

*Items*. An item is a simple object with no dependents; e.g. a board post that can be deleted, edited or viewed. In the object hierarchy tree, items are like leaves. An item can be a:

1. *Comment*: Items whose meaning depends on another; e.g. "*I agree"* makes no sense alone.
2. *Message*: Items with sender(s) and receiver(s); e.g. an email.
3. *Vote:* Items that convey a position, a choice from a response set.

*Spaces*. As leaves need branches, so items need spaces; e.g. an online wall that accepts photos is an information space. A space is a complex object with dependents. It can be deleted, edited or viewed like an item but can also contain objects; e.g. a bulletin board is a space. Spaces within spaces give object hierarchies, with the system itself the first space.

A space is a *parent* to the *child* entities it contains, who depend on it to exist. So deleting a space deletes its contents; e.g. deleting a board deletes its posts. The move operation changes the parent space of an object. The enter space operation shows the objects on display in it. As every entity is in the system space:

**P3:** *Every entity has a parent space, up to the system space.*

If every entity has a parent space[17], its *ancestors* are the set of all spaces that contain it, up to the system itself, the first ancestor. The *offspring* of a space are any child objects it contains, their children, etc. So all entities have owners and ancestors, and any space can have offspring.

## 6.7 OPERATIONS

| Entity Type | Operations |
|---|---|
| Any entity | View |
| 1. Social entity | ..., Delete, Edit |
| a. Persona | ..., Logon |
| b. Agent | ..., Delegate |
| c. Group | ..., Join |
| 2. Object entity | ..., Delete, Edit, Move |
| a. Item | ..., ConvertToSpace |
| b. Space | ..., Create, Enter |
| 3. Right entity | ..., Allocate, Re-allocate |
| a. Role | ..., Friend, Ban |

Table 6.2: Operation sets by entity type.

17. Except, of course, for the system itself.

Operations are actor initiated methods on information entities subject to access control.

*Operation sets*. Operations can be clustered for access control purposes; e.g. in the *delete set:* "delete" flags an entity for destruction, "undelete" reverses that, and "destroy" kills it permanently. An ACS that can manage one delete set operation can manage all of them. Likewise in the *edit set:* "edit" alters an entity value, "append" extends it, "version" is edit with backup, and Wikipedia's "revert" is the inverse of version. Again, variants of a set present the same ACS issues, so to resolve one is to resolve all. While edit changes an existing entity, the create operation set adds a new entity; e.g. to create or duplicate a Wikipedia stub for others to edit. Table 6.2 shows the operation sets for various entity types.

*View*. Operations like view are null acts that do not change their informational level target, but in some cultures staring at another is an act of aggression. The psychological process is *social facilitation,* where being looked at energizes the viewed party (Geen and Gange, 1983). Viewing people in a social system affects them because success in a social group depends on how others see you. Privacy, to control information about ourselves, is important for the same reason. The act of viewing also has effects on the community level; e.g. a "viral" online video makes others want to view it too.

The right to use an entity cannot be applied if the actor cannot see it, giving the ACS operational principle:

**P4**: *Any right to use an object implies a right to view it.*

*Communication*. In a simple communicative act, a sender creates a message that a receiver views. It is by definition a joint act where both parties have choice so communication should be by mutual consent. Privacy as the right to remain silent, not to communicate and not to receive messages, arises from this mutuality. In the physical world, people ask "*Can I talk to you?*" to get permission to communicate.

Some online systems, however, like email, do not recognize this. They give anyone the right to send a message to anyone, whether they will or no, and so invite spam. In contrast, in Facebook, chat, Skype and Twitter, one needs prior permission to message someone. The details of legitimate communication, where a *channel* is opened by mutual consent before messages are sent, are given in Whitworth and Liu (2009). The resulting ACS operational principle is:

**P5:** *Any communication act should have prior mutual consent.*

Progress in telephony illustrates how a technical communication has evolved socially . At first phones just transmitted information — the phone rang and one answered, not knowing who was calling. This allowed telemarketing, the forerunner of spam. Then cell phones showed caller id by default, so one could choose to respond, i.e. it was more mutual. Yet cell phone users still have to personally type in contact list names, while social networks let us each type in our own name and for others to add to their contact list. Cell phone companies could use this synergy, but as TV remote engineers are locked into the physical level, so cell-phone companies are locked into an information level mind-set[18]. They cannot see that people naturally share.

## 6.8 ROLES

Roles, like parent, friend or boss, simplify rights management by covering many cases, but still remain understandable, so people can review, evaluate and accept them. They are equally useful online, e.g. Wikipedia citizens can aspire to steward, bureaucrat or sysop roles by good acts. Slashdot's automated rating system offers the moderator role (Benkler, 2002) to readers who are registered (not anonymous), regular users (for a time), and those who have a positive "karma" (how others rate their comments). Every registered reader has five influence points to

18. Users could have the option to show a name instead of a number when they call. It could be their real name or a nickname, just as they now can choose to show their caller-id number. The social system would self-adjust, as receivers may not reply to anonymous senders. If people choose to show their name to their friends, that is their choice, so no privacy is lost. Privacy is not secrecy.

spend on others as desired over a three day period (or they expire). In this role democracy, highly rated commenters get more karma points and so more say on who is seen. The technology lets a community democratically direct its governance.

In information terms, a role is a variable rights statement, e.g. a friend role is a set of people with extra permissions. Roles are generic rights, giving the ACS operational principle:

**P6:** *A role is a right expressed in general terms using sets.*

Roles are the variables of social logic:

$$Role = (\boldsymbol{Actor}, \boldsymbol{Entity}, \boldsymbol{Operation})$$

The bolding indicates a variable, e.g. the owner role can be generally defined as any party who has all rights to an entity:

$$Role_{Owner} = (\boldsymbol{Owner}, Entity_i, Operation_{All})$$

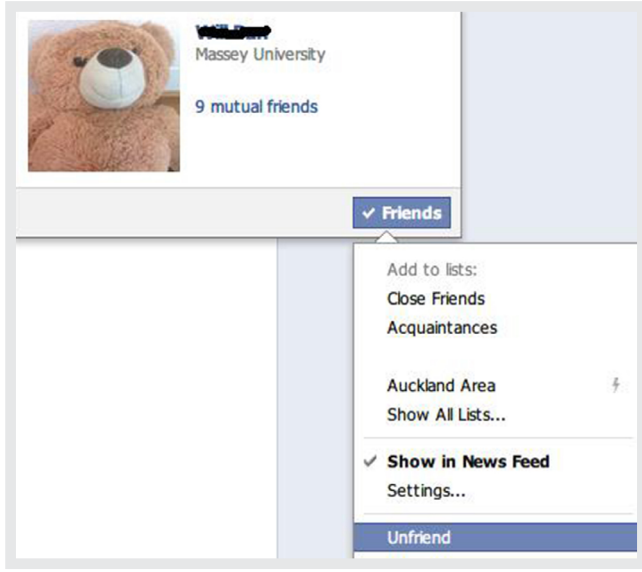



FIGURE 6.1: Unfriending in Facebook.

Making a person the owner just allocates the **Owner** set to include their persona. Roles are flexible, e.g. the friend role lets one change who can see photos posted on a Facebook wall:

$$Role_{Friend} = (\textbf{\textit{Friend}}\ ,\ Entity_{Wall}\ ,\ Operation_{View})$$

where **Friend** is a persona set. To "friend" another is to add them to this role set, and to "unfriend" is to remove them. As a variable can be undefined, so a role can be empty, i.e. a null friend set.

To "friend" is spoken of as an act on a person, but it does not change the persona entity, so it is really an act upon a local role (Figure 6.1). You decide your friend set, and do not need permission to friend or unfriend anyone, so bulletin boards can ban people at will (Figure 6.2). If banning were an act on another's persona it would need their consent. That it is an act on *my* role gives the ACS principle:

**P7.** *A space owner can ban or give entry to a persona without its owner's permission.*

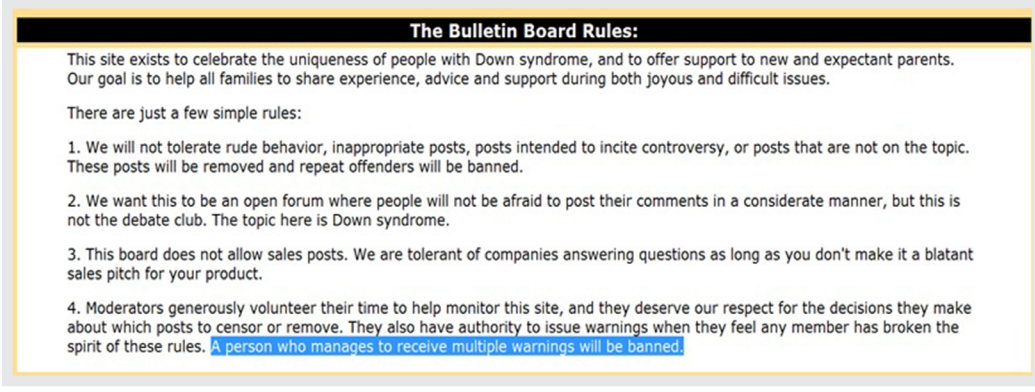



FIGURE 6.2: Bulletin board ban rules.

*Courtesy of Brian Whitworth and Adnan Ahmad. Copyright: CC-Att-SA-3 (Creative Commons Attribution-ShareAlike 3.0).*

Re-allocating actors is not the only way to alter a role. By definition, one can change a role's:

1. *Actor*. The role actor set.
2. *Entity*. The entities it applies to.
3. *Operation*. The operations it allows.

For example, a friend role could limit the objects it applies to, with some photos for family only. It could also allow adding comments to photos or not. Few current systems fully use the power of local roles; e.g. social networks could let actors define an acquaintance role, with fewer rights than a friend but more than the public, or an extended family role.

## 6.9 META-RIGHTS

Owning an object is the right to use it:

$$\text{Right}_{\text{Own}} = \text{R}\,(\textbf{Owner}, \text{Entity}_{\text{i}}, \text{Operation}_{\text{Use}})\,,$$

but an entity right can also be acted on, i.e. re-allocated. A meta-right is the right to re-allocate a right. In formal terms:

$$\text{Right}_{\text{MetaRight}} = \text{R}\,(\textbf{Owner}, \text{Right}_{\text{Own}}, \text{Operation}_{\text{Allocate}})\,,$$

where the entity acted on is a right. An owner with *all* rights to an entity also has its meta-rights, i.e. the right to change its rights. Paradoxically, fully owning an entity implies the right to give it away entirely. Reachability[19] requires meta-rights to be absolute, so there are no meta-meta-rights, giving the ACS operational principle:

**P8.** *A meta-right is the right to allocate any entity right, including the meta-right itself.*

19. Reachability, or halting, is that a program logic finishes and doesn't run endlessly.

Previously, to own an entity was to have all rights, but giving away use rights while keeping meta rights is still ownership; e.g. renting an apartment gives a tenant use rights, but the landlord still owns it, as they keep the meta-right. The tenant can use it but the owner says who can use it.

## 6.10 THE ACT OF CREATION

To create an object from nothing is as impossible in an information space as it is in a physical one. Creation *cannot* be an act upon the object created, which by definition does not exist before it is created. An actor cannot request ACS permission to create an object that does not exist. To create an information object, its data structure must be known, i.e. exist within the system. So creation is an act upon the system, or in general, an act on the space immediately containing the created object, giving the ACS operational principle:

**P9**. *Creation is an act on a space, up to the system space.*

This rule is well defined if the system itself is the first space. Creating is an act upon a space because it changes the space that contains the created object. If creation is an act upon a space, the right to create in a space belongs initially to the space owner:

$$Right_{Create} = R\ (SpaceOwner_i\ ,\ Space_i\ ,\ Operation_{Create})$$

The right to create in a space initially belongs to its owner, who can delegate it to others. The logic generalizes well; e.g. to add a board post, YouTube video or blog comment requires the board, video, or blog owner's permission. One can only create in a space if its owner permits. Now an ACS can be simply initialized as a system administrator owning the system space with all rights, including create rights. The system administrator must give rights away for the community to evolve.

*Creator ownership*. Object creation is a simple technical act, but a complex social one, e.g. how are newly created entity rights allocated? The 17th century Brit-

ish philosopher John Locke argued that creators owning what they create is both fair and increases community prosperity, i.e. it is legitimate (Locke, 1690/1963). The logic applied whether the product was a farmer's crop, a painter's painting or a hunter's catch. If the creator of something chooses to sell or give it away, that is another matter. A community that grants producers the right to their products encourages creativity. Conversely, why produce for others to own? This gives the ACS operational principle:

**P10**. *The creator of new entity should immediately gain all rights to it.*

Creator ownership conveniently resolves the issue of how to allocate new object rights — they go to its creator, including meta-rights. Yet a program can act any way it likes; e.g. it could make the system administrator own all created objects. Creator ownership is a social not a technical requirement , i.e. a social success condition.

*Creation conditions*. A creation condition is when a space owner *partially* delegates creation. It can limit:

1. *Object type*. The object type created; e.g. the right to create a conference paper is not the right to create a mini-track space.
2. *Operations*. The operations allowed on created objects; e.g. blog comments are not usually editable once added, but ArXiv lets authors edit publications as new versions.
3. *Access*. Who can access created objects; e.g. YouTube gives contributors exclusive edit rights, but Wikipedia lets anyone edit any creation.
4. *Viewing*. Who can view created objects, e.g. bulletin boards let others view your submission but conferences in the paper review phase do not.
5. *Editing*. The field values of a created object, e.g. date added, may be non-editable. The space owner may also set field default values.

A space owner can delegate creation rights as needed; e.g. to show vote results only to people who have voted, to avoid bias.

*Transparency*. Yet fairness dictates a creator's right to know creation conditions in advance. In general, transparency is the right to view rights and rules of governance that affect you or might affect you. So those who create in a space should know the creation rules in advance. The ACS principle is:

**P11**. *A person can view in advance any rights that could apply to him or her*.

Successful socio-technical systems like Facebook, YouTube and Wikipedia support transparency. A space owner can delegate the right to create in whole or part, but should disclose creation conditions up front, so that potential creators can decide to create or not.

For any entity, the system can assign these roles:

1. *Owner*. With meta rights to the entity.
2. *Parent*. The containing space owner.
3. *Ancestors*. Ancestor space owners (SA the first ancestor).
4. *Offspring (space only)*. The owners of any entities contained in a space.
5. *Local public (space only)*. Actors who can enter the space.

A space's local public role defines what others can do in the space:

$$Role_{LocalPublic} = (\textbf{\textit{LocalPublic}} , Space_i , Operation_{Any})$$

It can be set manually, as friends are allocated, or point to a GlobalPublicList.

*Ancestor role*. A conference paper's ancestors are its mini-track, track and conference chairs. An entity, being part of the space it is in, should be visible to the owner of that space. Privacy does not contradict this, as it refers to the display of personal information, not created object information. Generalizing, the ACS principle is:

**P12**. *A space owner should have the right to view any offspring*.

So the ancestor role for any entity is given view rights to it:

$$Role_{Ancestor} = (\textbf{\textit{Ancestors}}, Entity_i, View)$$

For example, a paper posted on a conference mini-track should be visible to mini-track, track and conference chairs, but not necessarily to other track or mini-track chairs. Ancestors should be notified of new offspring and offspring of new ancestors. The social logic is that a paper added to a mini-track is also added to the track, so the track chair can also view it.

*Offspring role*. An entity created in a parent space must be created by an actor with the right to enter that space. If a space bans the owner of an object in it, the object is disowned, contradicting P1. A child object's owner must be able to enter its space to act on it, even if they cannot do anything else. By extension, they can also enter any ancestor space. This does not imply any other rights. The ACS principle is:

**P13**. *An entity owner should be able to enter any ancestor space.*

e.g. adding a mini-track paper should let one enter the track and conference spaces, even if one cannot see or do anything there. Any space should allow its offspring owners to enter it:

$$Role_{Offspring} = (\textbf{\textit{Offspring}}, Space , Enter)$$

Table 6.3 summarizes the basic access rights for entities and spaces.

| Entity | *View* | *Delete* | *Edit* | *Display* | *Allocate* |
|---|---|---|---|---|---|
| *Ancestor* | √ | | | | |
| *Parent* | √ | | | √[1] | |
| *Owner* | √ | √ | √ | √[2] | √ |
| *LocalPublic* | √[1,2] | | | | |

| Space also | *Enter* | *Create* | | | |
|---|---|---|---|---|---|
| *Ancestor* | √ | | | | |
| *Owner* | √ | √ | | | |
| *LocalPublic* | √[1] | √[1] | | | |
| [1]As allocated by the owner. [2] As allocated by the parent. | | | | | |

Table 6.3: Entity and space access rights.

## 6.11 DISPLAY

To display an object is to let others view it. The right to display is not the right to view; e.g. viewing a video online does not let you display it on your web site[20]. Display is the meta-right to view, i.e. the right to give the right to view an object to others; e.g. privacy is the meta right to display the persona object. As people have private numbers in a phone book, so Facebook or Linkedin persona are displayed to the public by owner consent. The phone company that owns a phone book list can also choose not to display a listing, giving the ACS principle:

**P14**. *Displaying an entity in a space requires both persona and space owner consent.*

Displaying an item in a space is its owner giving display rights to the space owner. For example, to put a physical notice on a shopkeeper's notice board involves these steps:

1. *Creation*. Create a notice. You own it and can still change it, or rip it up.
2. *Permission*. Ask the board owner if it can be posted on the notice board.

20. A more complex example is: if going out in public implicitly gives others the right to view you, anyone can take a photo of you without your consent, but they cannot display that photo on a magazine cover without your consent.

3. *Post*. The board owner may vet notices in advance or let people post themselves.
4. *Removal*. As the notice is displayed by mutual consent, either can remove it.

The shopkeeper's right to take a notice down is not the right to destroy it, because *he or she does not own it*. Nor can he or she alter (deface) notices on the board.

The same social logic applies online. Creating a YouTube video gives you view rights to it, but it is not yet displayed to the public. Giving the right to display a YouTube video is like giving a notice to a shopkeeper to post on their board. The item owner delegates the right to display their video to the space owner, YouTube, who then can choose to display it in their space. In general, to display any video, photo or text in any online space requires mutual consent, as one party gives another the right to display, giving the ACS principle:

**P15**. *An entity owner must give view meta-rights to a space owner to display in that space*

| **Display result** | | *Space owner* | |
|---|---|---|---|
| | | *Accept* | *Reject* |
| *Object owner* | *Submit* | YES | NO |
| | *Withdraw* | NO | NO |

TABLE 6.4: A display interaction.

Display as a rights transaction is the basis of all publishing, whether of a video, a book or a paper. Table 6.4 shows how the display result depends on the interaction between author (object owners) and publisher (space owners) rights. A space can delegate display rights, to let creators display as desired, e.g. YouTube. Or it

may vet items before display and reject some, e.g. ArXiv, which also lets authors withdraw submissions. Bulletin boards let anyone submit but not withdraw, and reserve the right to moderate postings, i.e. reject later.

Authors who publish must give *some* rights to the publisher to publish. After that, they cannot "un-publish", nor can a publisher[21]. Yet authors do not give *all* rights to publisher, e.g. attributions rights. Usually the right to publish is given once only, but some publisher contracts take the right to do so many times; e.g. publishing an IGI book chapter led to its re-publication in other collections without the author's permission[22] (Whitworth & Liu, 2008).

*Entity creation*. Technically, creating an entity is simple — the program just creates it — but socially adding into another's space is not a one-step act. Adding a YouTube video involves:

1. *Registration*. Create a YouTube persona.
2. *Entry*. Enter YouTube (not banned).
3. *Creation*. Create and upload a video.
4. *Edit*. Edit video title, notes and properties.
5. *Submit*. Request YouTube to display the video to their public.
6. *Display*. The public sees it and can vote or comment.

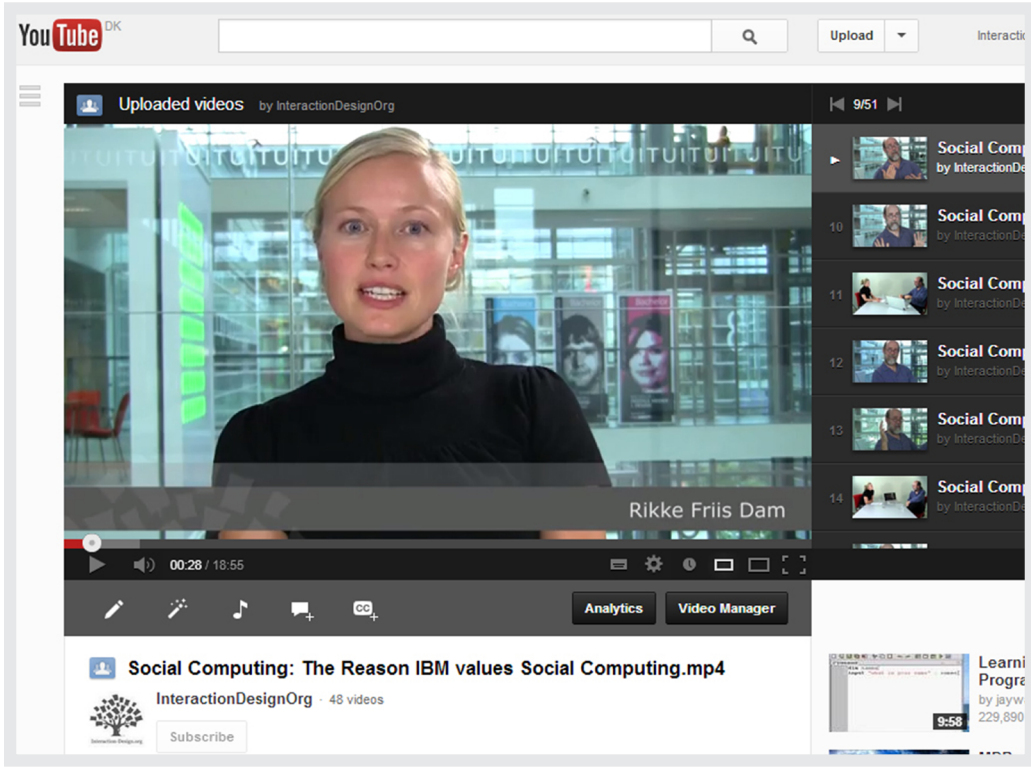
YouTube lets anyone registered in the public role (1) enter their space and (2) create a video, by uploading or recording (3), which they own (4), They can view and edit its details in private. At this point, the video is visible to them and administrators, but not to the public. They can still delete it. (5) It is then submitted to YouTube for display to its public. This occurs quickly as (6) display rights are

21. A journal can 'retract' a publication, to deny it, but cannot 'un-publish' it.

22. The paper "*Politeness as a Social Computing Requirement"* published in IGI's *Handbook of Conversation Design for Instructional Applications* was later republished in their *Selected Readings on the Human Side of Information Technology* and in *Human Computer Interaction: Concepts, Methodologies, Tools, and Applications*without the author being even advised, let alone asked.

delegated. To create, edit and display a video are distinct steps. YouTube can still *reject* videos that fail its copyright or decency rules. This is not a delete, as the owner can still view, edit and resubmit it. A technology design that let space owners permanently delete videos would discourage participation.

*Consistency* For the above logic to be consistent, it should also apply when the video itself is a space for comments or votes. Indeed it is, as video owners have the choice to allow comments or votes just as YouTube had the right to accept their video (Figure 6.3). That YouTube gives the same rights to others as it takes for itself is a key part of its success as a socio-technical system.




*Copyright status: Unknown (pending investigation). See section "Exceptions" in the copyright terms below.*

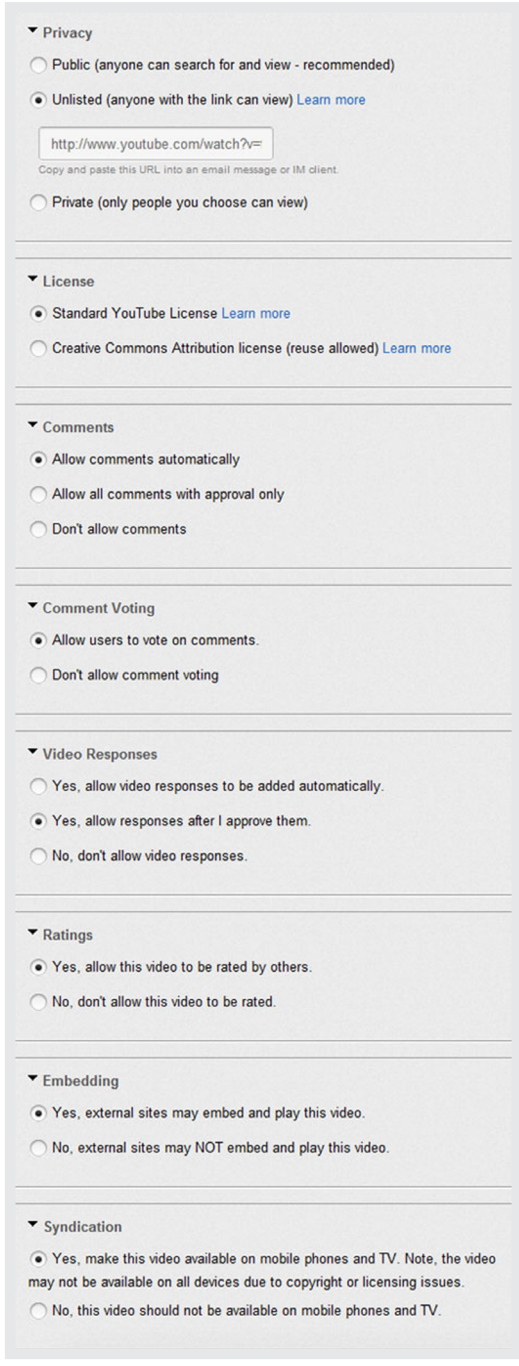



FIGURE 6.3 A-B: A YouTube video and its rights.

*Courtesy of Brian Whitworth and Adnan Ahmad. Copyright: CC-Att-SA-3 (Creative Commons Attribution-ShareAlike 3.0).*

## 6.12 RIGHTS OPERATIONS

The right to re-allocate rights makes social interaction complex, but it also lets socio-technical systems evolve from an initial state of one administrator with all rights to a community sharing rights. Use and meta rights can be re-allocated, as follows:

1. *Transfer*. Re-allocate all rights, including meta-rights. Rights are irrevocably given to the new owner; e.g. after selling a house, the old owner has no rights to it.
2. *Delegate*. Re-allocate use rights but not meta-rights. It can be reversed, e.g. renting.
3. *Divide*. A right divided among an actor set requires them to join to permit the act, so any party can stop it; e.g. couples who jointly own a house must both agree to sell it.
4. *Share*. A right shared across an actor set lets each exercise it as if they owned it exclusively; e.g. couples who severally share a bank account can each take out all the money.

| **Right Operation** | *Allocated by* | | *Allocated to* | |
|---|---|---|---|---|
| | *Meta-rights* | *Use rights* | *Meta-rights* | *Use rights* |
| *Transfer* | | | √ | √ |
| *Delegate* | √ | | | √ |
| *Merge use* | √ | ½ √ | | ½ √ |
| *Merge all* | ½ √ | ½ √ | ½ √ | ½ √ |
| *Share use* | √ | √ | | √ |
| *Share all* | √ | √ | √ | √ |

TABLE 6.5: Results use and meta rights re-allocations.

Table 6.5 shows the resultant states of each rights operation for allocator and allocatee. Dividing a right means that all must agree to it, while sharing a right means that any party alone can activate it. In information terms, dividing is an AND set and sharing is an OR set. This is not just splitting hairs, as if a couple owns a house jointly, both must sign the sale deed to sell it, but if they own it severally, either party can sell it and take all the money. Re-allocating rights applies to many social situations; e.g. submitting a paper online can give all rights to a primary author, let the primary author delegate rights to others, merge rights so that all authors must confirm changes, or share rights among all authors. Each has different consequences; e.g. sharing an edit right is risky but invites participation, while merging it among the authors is safe but makes contributing harder.

*Delegation.* Delegation, by definition, does not give meta-rights, so a delegatee cannot pass rights on. Renting an apartment gives no right to sub-let, and lending a book does not give the right to on-lend it. It is not hard to show that if delegatees delegate, accountability is lost. If one lends a book to someone who lends it to another who loses it, who is accountable? This gives the operational principle:

**P16**. *Delegating does not give the right to delegate.*

Allocating use rights to an existing object makes the target person accountable for it, so it requires consent; e.g. one cannot add a paper co-author without agreement. The principle is:

**P17**. *Allocating existing object use rights to a person requires their consent.* An ACS might ask:

*"Bob offers you edit rights to 'The Plan', do you accept?"*

In contrast, rights to null acts, like view or enter, or to acts like create, can be allocated without consent because they imply no accountability:

**P18**. *Allocating null rights to existing objects, or the right to create, requires no consent.*

So Facebook owners owners can freely delegate entry, view or create rights to their wall space to anyone without permission.

*Social networks*. Social networks currently send messages like:

*"Bob wants to be friends with you."*

Friendship is assumed to be a tit-for-tat social trade, where I offer to make you a friend if you make me one. Yet by P7, one can befriend another without their permission[23]. If the software allowed it, we might get messages like:

*"Bob considers you a friend, please visit his page. "*

This is giving friendship, not trading it. As one can love a child unconditionally, even if they do not return the favour, so friendship need not be a commercial transaction.

For a social network to consider that the friends of my friends are also my friends contradicts P16. As liking someone does not guarantee that one will like their friends, so making a friend should not reset my friend list. This illustrates a technical option that failed because it had no social basis.

## 6.13 IMPLEMENTATION

Traditional access control enforcement is done by a security kernel mechanism. A security kernel is a trusted software module that intercepts every access request call submitted to a system and decides if it should be granted or denied, based on some specified access policy model. Usually, a centralized approach is used, so one policy decision point handles all resource requests. The access request gets either an executed action result or a permission denied message. Social networks have millions of users, so centralized or semi-decentralized cer-

23. So giving another the right to view your Facebook wall does not let them spam you with change notices from theirs.

tificates are a bottle neck. This plus the social need for local ownership by content contributors suggests a strategy of distributed certificates to implement the ACS policy model outlined here. Allowing local policy decision points to handle resource requests also ensures local user control over resources. If distributed certificates are stored in the stakeholder's namespace, only he or she can access and modify them (Figure 6.4).

## 6.14 AN EXAMPLE: TAPPING THE INTERNET

The power of technology will soon allow, if it does not already, the power to store every phone call, text message, e-mail, tweet and online posting, every day from now on. The PRISM system's capacity to record all global communications all the time illustrates what is possible. The privacy problem is that listening in on people gives power over them and computer databases extend that power to the indefinite past. That what you post online can come back to haunt you later will reduce synergy, as in a police state. Security experts say that if you do nothing wrong you have nothing to fear, but who among us is perfect? And happenstance can make anyone a person of interest, e.g. an innocent neighbour or relative of a terrorist, whose online past is then open to public or government scrutiny.

The social view that violating everyone's rights to catch some criminals is counter-productive was built into the US constitution by its founders, with rules preventing random search and unwarranted monitoring. A government should not deny its citizens freedom or privacy without just cause, but it now seems that PRISM has been doing just that online, justifying that firstly it only collects "meta-data" like call length and phone number and secondly that it is necessary for security.

In information terms, meta-data is data about data, like an average. In contrast, call length is data about a call, i.e. just data. In a database, call length is a property of the call entity as call content is. The meta-data argument lacks validity

but the claim that community security trumps personal privacy does not, because a citizen is part of the community. One has privacy with respect to others but not with respect to the state one is in. So the fourth amendment allows monitoring given *it is truly a community act*, as formally checked by a judge. Wiretapping requires a warrant, to stop vested interests from abusing community power, as the US tax department targeting groups based on political affiliation did. As a community cannot permit what it doesn't know of, such systems should not be secret. This social logic applies whether tapping a phone line or the Internet in general.

Technology power offers not just more capability but also more options. In the past, a wire-tap revealed a communication, content and all, but PRISM is ignoring the content and claiming it is ok. A technology evolution has undone a prior social evolution, namely the fourth amendment of the US constitution, so lawmakers have to go back to original causes. The reason for this amendment is shown by a use case: suppose a computer system tapping all phone calls uncovers terrorist plots which thus fail. Organized terrorists then stop using phones and say use texts instead, although foolish novices may be picked up. If texts are tapped, they move on to e-mail, blogs, chat, etc. Spam and virus wars follow this pattern. The result is a system that monitors all the communications of its good citizens but is bypassed by most terrorists and is now ripe for corruption, abuse and hijack. Monitoring *all* communications may find *some* terrorists, but is it worth it? The founders of the US constitution certainly did not think so, and electronic surveillance is no different from physical surveillance.

The technology that causes a social problem can also solve it if built with social principles like privacy in mind. One solution is to consider all personal data "radioactive" and so not collect it. What is not stored cannot be stolen or subpoenaed. Companies like Vodafone that personally bill people have to collect personal data but companies like Google have no reason to store names and addresses that governments can later demand. If personal data is collected let it be kept behind

a firewall, apart from the general network data that systems like PRISM analyze. The latter could then be provided, but to get individual names would require a warrant, just as an individual phone wiretap does. With a little ingenuity, technology can fight terrorism with  without going socially backwards.

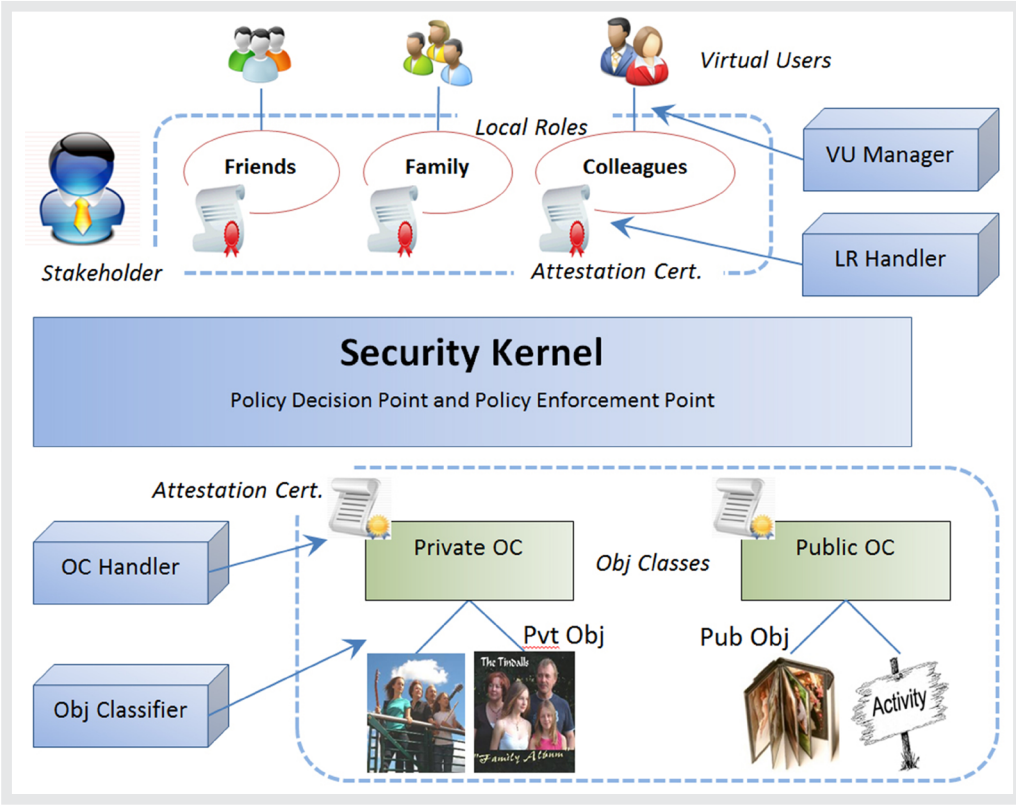



FIGURE 6.4: Distributed access control model architecture.

## 6.15 A SOCIOTECHNICAL STANDARD

Legitimate access control can assign owner, parent, ancestor, offspring and local public rights to objects and spaces in a way that encourages social system success. We hope the following ACS principles are a first step to developing common standards in socio-technical design:

1. All non-null entity rights should be allocated to actors.
2. A persona should be owned by itself.
3. Every entity has a parent space, up to the system space.
4. Any right to use an object implies a right to view it.
5. Any communication act should have prior mutual consent.
6. A role is a right expressed in general terms using sets.
7. A space owner can ban or give entry to a persona without its owner's permission.
8. A meta-right is the right to allocate any entity right, including the meta-right itself.
9. Creation is an act on a space, up to the system space.
10. The creator of new entity should immediately gain all rights to it.
11. A person can view in advance any rights that could apply to them.
12. A space owner should have the right to view any offspring.
13. An entity owner should be able to enter any ancestor space.
14. Displaying an entity in a space requires both persona and space owner consent.
15. To display an entity in a space, the entity owner gives view meta-rights to the space owner.
16. Delegating does not give the right to delegate.
17. Allocating existing object use rights to a person requires their consent.
18. Allocating null rights to existing objects, or the right to create, requires no consent.

## 6.16 DISCUSSION QUESTIONS

The following questions are designed to encourage thinking on the chapter and exploring socio-technical cases from the Internet. If you are reading this chapter in a class - either at university or commercial – the questions might be discussed in class first, and then students can choose questions to research in pairs and report back to the next class.

1. What is access control? What types of computer systems use it? Which do not? How does it traditionally work? How do social networks challenge this? How has access control responded?
2. What is a right in human terms? Is it a directive? How are rights represented as information? Give examples. What is a transmitted right called? Give examples.
3. What is the difference between a user and an actor? Contrast user goals and actor goals. Why are actors necessary for online community evolution?
4. Is a person always a citizen? How do communities hold citizens to account? If a car runs over a dog, is the car accountable? Why is the driver accountable? If online software cheats a user, is the software accountable? If not, who is? If automated bidding programs crash the stock market and millions lose their jobs, who is accountable? Can we blame technology for this?
5. Contrast an entity and an operation. What is a social entity? Is an online persona a person? How is a persona activated? Is this like “possessing” an online body? Is the persona “really” you? If a program activated your persona, would it be an online zombie?
6. What online programs greet you by name? Do you like that? If an online banking web site welcomes you by name each time, does that improve your relationship with it? Can such a web site be a friend?

7. Compare how many hours a day you interact with people via technology vs. the time spent interacting with programs alone? Be honest. Can any of the latter be called conversations? Give an example. Are any online programs your friend? Try out an online computer conversation, e.g. with Siri, the iPhone app. Ask it to be your *personal* friend and report the conversation. Would you like a personal AI friend?
8. Must all rights be allocated? Explain why. What manages online rights? Are AI programs accountable for rights allocated to them? In the USS Vincennes tragedy, a computer program shot down an Iranian civilian airliner. Why was it not held to account? What actually happened and what changed afterwards?
9. Who should own a persona and why? For three STSs, create a new persona, use it to communicate, then try to edit it and delete it. Compare what properties you can and cannot change. If you delete it entirely, what remains? Can you resurrect it?
10. Describe two ways to join an online community and give examples. Which is easier? More secure?
11. Describe, with examples, current technical responses to the social problems of persona abandonment, transfer, delegation and orphaning. What do you recommend in each case?
12. Why is choice over displaying oneself to others important for social beings? What is the right to control this called? Who has the right to display your name in a telephone listing? Who has the right to remove it? Does the same apply to an online registry listing? Investigate three online cases and report what they do.
13. How do information entities differ from objects? How do spaces differ from items? What is the object hierarchy and how does it arise? What is the first space? What operations apply to spaces but not items? What operations

apply to items but not spaces? Can an item become a space? Can a space become an item? Give examples.

14. How do comments differ from messages? Define the right to comment as an AEO triad. If a comment becomes a space, what is it called? Demonstrate with three commenting STSs. Describe how systems with "deep" commenting (comments on comments, etc.) work. Look at who adds the depth. Compare such systems to chats and blogs – what is the main difference?
15. For each operation set below, explain the differences, give examples, and add a variant to each set:

    - *Delete*: Delete, undelete, destroy.
    - *Edit*: Edit, append, version, revert.
    - *Create:* Create.

    Define a fourth operation set.
16. Is viewing an object an act upon it? Is viewing a person an act upon them? How is viewing a social act? Can viewing an online object be a social act? Why is viewing necessary for social accountability?
17. What is communication? Is a transfer like a download a communication? Why does social communication require mutual consent? What happens if it is not mutual? How does opening a channel differ from sending a message? Describe online systems that enable channel control.
18. Answer the following for three different but well known communication systems: Can a sender be anonymous to a receiver? Can a receiver be anonymous to a sender? Can senders or receivers be anonymous to moderators? Can senders or receivers be anonymous to the transmission system?
19. Answer the following for a landline phone, mobile phone and Skype: How does the communication request manifest? What information does a receiver get and what choices do they have? What happens to anonymous senders? How does one create an address list? What else is different?

20. What is a role? Can it be empty or null? How is a role like a maths variable or computing pointer? Give role examples from three popular STSs. For each, give the ACS triad, stating what values vary. What other values could vary? Use this to suggest new useful roles for those systems.
21. How can roles, by definition, vary? For three different STSs, describe how each role variation type might work. Give three different examples of implemented roles and suggest three future developments.
22. If you unfriend a person, should they be informed? Test and report what actually happens on three common social networks. Must a banned bulletin board "flamer" be notified? What about someone kicked out of a chat room? What is the general principle here?
23. What is a meta-right? Give physical and online examples. How does it differ from other rights? Is it still a right? Can an ACS act on meta-rights? Are there ACS meta-meta-rights? If not, why not? What then does it mean to "own" an entity?
24. Why can creating an item not be an act on that item? Why can it not be an act on nothing? What then is it an act upon? Illustrate with online examples.
25. Who owns a newly created information entity? By what social principle? Must this always be so? Find online cases where you create a thing online but *do not* fully own it.
26. In a space, who, initially, has the right to create in it? How can others create in that space? What are creation conditions? What is their justification?
27. Find online examples of creation conditions that limit the object type, operations allowed, access, visibility and restrict edit rights. How obvious are the conditions to those creating the objects?
28. Give three examples of creating an entity in a space. For each, specify the owner, parent, ancestors, offspring and local public. Which role(s) can the owner change?

29. For five different STS genres, demonstrate online creation conditions by creating something in each. How obvious were the creation conditions? Find examples of non-obvious conditions.
30. For the following, explain why or why not. Suppose you are the chair of a computer conference with several tracks. Should a track chair be able to exclude you, or hide a paper from your view? Should you be able to delete a paper from their track? What about their seeing papers in other tracks? Should a track chair be able to move a paper submitted to their track by error to another track? Investigate and report comments you find on online systems that manage academic conferences.
31. An online community has put an issue to a member vote. Discuss the effect of these STS options:
    - Voters can see how others voted, by name, before they vote.
    - Voters can see the vote average before they vote.
    - Voters can only see the vote average after they vote, but before all voting is over.
    - Voters can only see the vote average after all the voting is over.
32. An online community has put an issue to a member vote. Discuss the effect of these STS options:
    - Voters are not registered, so one person can vote many times.
    - Voters are registered, but can change their one vote any time.
    - Voters are registered, and can only vote once, with no edits.

    Which option might you use and when?
33. Can the person calling a vote legitimately define vote conditions? What happens if they set conditions such as that all votes must be signed and will be made public?

34. Is posting a video online like posting a notice in a local shop window? Explain, covering permission to post, to display, to withdraw and to delete. Can a post be deleted? Can it be rejected? Explain the difference. Give online examples.
35. Give physical and online examples of rights re-allocations based on rights and meta-rights. If four authors publish a paper online, list the ownership options. Discuss how each might work out in practice. Which would you prefer and why?
36. Should delegating give the right to delegate? Explain, with physical and online examples. What happens to ownership and accountability if delegatees can delegate? Discuss a worst case scenario.
37. If a property is left to you in a will, can you refuse to own it, or is it automatically yours? What rights cannot be allocated without consent? What can? Which of these rights can be freely allocated: Paper author. Paper coauthor. Track chair. Being friended. Being banned. Bulletin board member. Logon ID. Bulletin board moderator. Online Christmas card access? Which rights allocations require receiver consent?
38. Investigate how SN connections multiply. For you and five others find out the number of online friends and calculate the average. Based on this, estimate the average friends of friends in general. Estimate the messages, mails, notifications etc. you get from all your friends per week, and from that calculate an average per friend per day. If you friended all your friend's friends, potentially, how many messages could you expect per day? What if you friended your friend's friend's friends too? Why is the number so large? Discuss the claim of the film *Six Degrees of Separation*, that everyone in the world is just six links away from everyone else.

39. Demonstrate how to “unfriend” a person in three social networks. Are they notified? Is unfriending “breaking up”? That an “anti-friend” is an enemy, suggests “anti-Facebook” sites. Investigate technology support for people you hate, e.g. celebrities or my relationship ex. Try anti-organization sites, like sickfacebook.com. What purpose could technology support for anti-friendship serve?

## YOUR NOTES AND THOUGHTS ON CHAPTER 6

Record your notes and thoughts on this chapter. If you want to share these thoughts with others online, go to the bottom of the page at: http://www.interaction-design.org/books/the_social_design_of_technical_systems_2nd_ed/online_rights.html

**NOTES:**

# CHAPTER 7

# The Future

"*The future is socio-technology not technology*"

The future of computing depends on its relationship to humanity. This chapter contrasts the socio-technical and technical visions.

## 7.1 TECHNOLOGY UTOPIANISM

Technology utopianism is the view that technology will endlessly improve until it not only does what people do, but does it better. The idea that computer technology will define the future of humanity is popular in fiction; e.g. Rosie in *The Jetsons*, C-3PO in *Star Wars* and Data in *Star Trek* are robots that read, talk, walk, converse, think and feel. We do these things easily so how hard could it be? In films, robots learn (*Short Circuit*), reproduce (*Stargate's* replicators), think

(*The Hitchhiker's Guide's* Marvin), become self-aware (*I, Robot*) and eventually replace us (*The Terminator, The Matrix*). In this view, computers are an unstoppable evolutionary juggernaut that must eventually supersede us, even to taking over the planet (Figure 7.1).

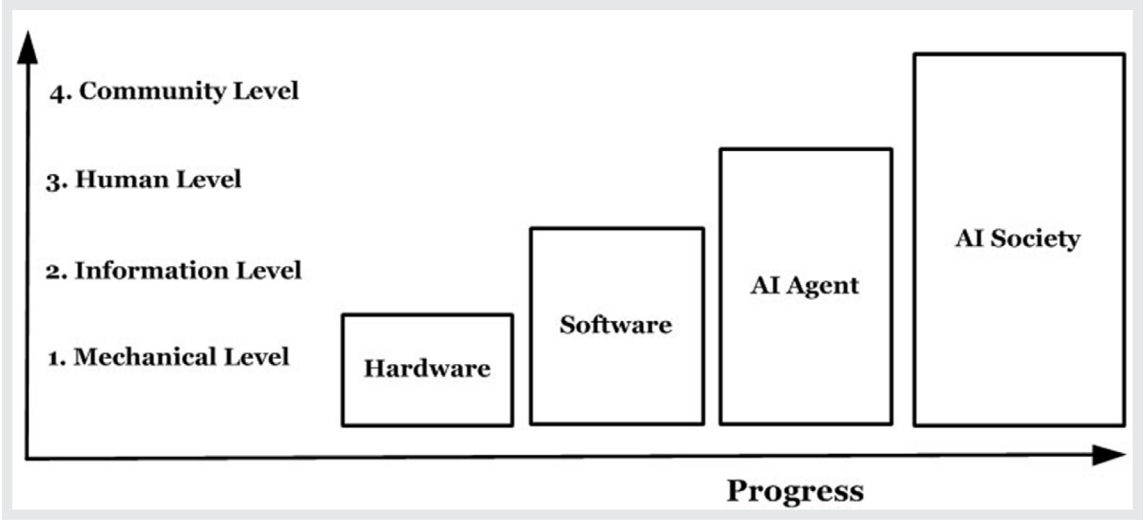



FIGURE 7.1: Technological utopianism.

*Courtesy of Brian Whitworth and Adnan Ahmad. Copyright: CC-Att-SA-3 (Creative Commons Attribution-ShareAlike 3.0).*

Yet cartoon fiction (Figure 7.2a) and movie fiction (Figure 7.2b) are not fact. Today's *actual* housework robot is the Roomba (Figure 7.2c). It picks up dirt but has problems like turning in repeating circles, software paralysis, thread clog build up and getting stuck under the couch. If you wash it in water, it dies, but make no mistake, this is a clever robot. *Yet it needs help to survive in a house with perfectly flat terrain, no enemies, and free food*[1]. Drop it outside, or even just stop tending it, and it dies out, i.e. insects do better. Right now, computers could not even conquer a planet of cockroaches, let alone people.

1. The Roomba 'feeds' from electricity sockets in the house.

FIGURE 7.2.A: A Cartoon Robot in Superman.

*Courtesy of Fleischer Brothers production. Copyright: pd (Public Domain (information that is common property and contains no original authorship)).*

FIGURE 7.2.B: A Movie Robot in Star Wars .

*Courtesy of Brayon Michael Pieske. Copyright: pd (Public Domain (information that is common property and contains no original authorship)).*


FIGURE 7.2.C: An Actual Robot (iRobot Roomba version)[104].

*Courtesy of Larry D. Moore. Copyright: CC-Att-SA-3 (Creative Commons Attribution-ShareAlike 3.0)* [2,3,4].

Compare people and robots on the tasks we do well, like conversation. People chat easily but the computers next to us are strangely silent. The technology to convert text to speech is well established, so why don't cars for example talk to us, like KITT in *Knight Rider*? The reason is that computer conversations are truly embarrassing, e.g. this Cleverbot (CB) conversation, run through twice[5]:

**Me:** *If you become me, am I you?*

***CB:*** I believe in you babe, or What if I don't believe they are?

**Me:** *If you are not you, who are you?*

***CB:*** I am not a vampire, or I'm you.

2. Courtesy of Fleischer Brothers production
3. Courtesy of Brayon Michael Pieske
4. Courtesy of Larry D. Moore. Copyright: CC-Att-SA-3
5. From http://cleverbot.com

**Me:** *How do you know?*

***CB:*** Because I'm a girl, or Because you know I know I know you know me.

**Me:** *But girls can be vampires.*

***CB:*** Yes they can, or No they can't.

**Me:** *So could you be a vampire girl?*

***CB:*** No. I am not, or A girl.

**Me:** *Could you become a vampire girl?*

***CB:*** Why? or No, could you become a girl?

**Me:** *If you were bitten by a vampire would you be a vampire girl?*

***CB:*** Who have you told that I am Jesus? or I'd buy an X-wing, because they're awesome.

**Me:** *So you could become a vampire girl?*

***CB:*** I doubt it. But I know it could be true, or I could be a lot of things.

**Me:** *That contradicts what you said before.*

***CB:*** I don't think it does, or You are contradicting yourself.

**Me:** *You could be not you and a vampire.*

***CB:*** No, I'm just a rabbit, or Girl vampire.

**Me:** *Could you be not you as a vampire girl?*

***CB:*** No, or I could probably be a giraffe, if science were able to make me so.

**Me:** *You just said you could be a vampire girl which is not you.*

***CB:*** I never said I was a boy, or I never said I was a girl.

This painful conversation has no end. It is not hard to see the programmers faking conversation, i.e. it is a Turing test fail.

Spatial coordination is no better, as a comparison of the Robot Soccer World Cup and the human Soccer World Cup reveals a vast disparity. Artificial Intelligence (AI) advocates say robots will surpass people in forty years, but they said the same forty years ago.

In vision, computers struggle with retinal, not cortical, activity:

*"Computers are no real competition for the human brain in areas such as vision, hearing, pattern recognition and learning. ... And when it comes to operational efficiency there is no contest at all. A typical room-size supercomputer weights roughly 1,000 times more, occupies 10,000 times more space and consumes a millionfold more power ..."*

*-- Boahen, 2005*

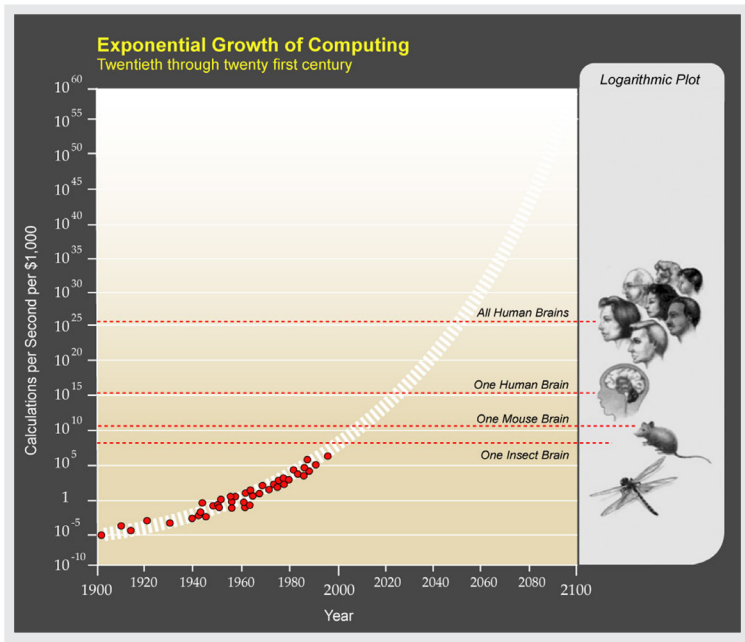



FIGURE 7.3: The exponential growth of simple processing power.

*Courtesy of Ray Kurzweil and Kurzweil Technologies, Inc.. Copyright: CC-Att-SA-1 (Creative Commons Attribution-ShareAlike 1.0 Unported).*

Clearly talking, walking and seeing are not as easy as our brain makes it seem.

Technology utopians use Moore's law, that computer power doubles every eighteen months[6], to predict a future *singularity* at about 2050, when computers go beyond us. Graphing simple processing power by time suggests when computers will replace people (Kurzweil, 1999) (Figure 7.3). In theory, when computers have as many transistors as the brain has neurons, they will process as well as the brain, but the assumption is that brains process as computers do, which they do not.

In Figure 7.3, computers theoretically processed as an insect in 2000, and as a mouse in 2010; they will process as a human in 2025 and beyond all humans in 2050. So why, right now, are they unable to do even what ants do, with their neuron sliver? How can computers get to human conversation, pattern recognition and learning in a decade if they are currently not even at the insect level? The error is to assume that all processing is simple processing, i.e. calculating, but calculating cannot cross the 99% performance barrier.

## 7.2 THE 99% PERFORMANCE BARRIER

Computers calculate better than us as cars travel faster and cranes lift more, but calculating is not what brains do. Simple tasks are solved by simple processing, but tasks like vision, hearing, thinking and conversing are not simple[7]. Such tasks are *productive*, i.e. the demands increase geometrically with element number, not linearly[8].

6. See for example http://karlnordstrom.ca/ideas/?p=6

7. Simple processing works at the informational level, e.g. a literal number or word recall, number calculations or an *eidetic* memory of a scene. The human level introduces the processing of processing, as explained shortly.

8. On an 8x8 chess board, the number of possible chess games is 10120 - more than the atoms in the observable universe. In an Indian tale, the inventor of chess was offered a boon by the king. He asked for a grain of wheat on the first chess square, two grains on the second, four on the third, eight on the fourth, and so on, doubling the grains each time. It seemed a modest request, so the king agreed, but the result was over 18 billion, billion grains, more weight than all life on Earth. This is the productivity problem for only 64 elements!

For example, the productivity of language is that five year olds know more sentences than they could learn in a lifetime at one per second (Chomsky, 2006). Children easily see that a Letraset page (Figure 7.4) is all 'A's, but computers struggle with such productive variation. As David Marr observed (Marr, 1982), using pixel level processing for pattern recognition is:

> "*like trying to understand bird flight by studying only feathers. It just cannot be done*"

Simple processing, at the part level, cannot manage productive task wholes – the variation is just too much. AI experts who looked beyond the hype knew decades ago that productive tasks like language would not be solved anytime soon (Copeland, 1993).

FIGURE 7.4: Letraset page for letter 'A'

The bottom line for a simple calculation approach is *the 99% performance barrier*, where the last performance percentage can take more effort than all the rest together, e.g. 99% accurate computer voice recognition is one error per 100 words, which is unacceptable in conversation. For computer auto-drive cars, 99% accuracy is an accident a day! In the 2005 DARPA Grand Challenge, only five of 23 robot vehicles finished a simple course (Miller et al., 2006).

In 2007, six of eleven better funded vehicles finished an urban track with a top average speed of 14mph. By comparison, skilled people drive for *decades* on bad roads, in worse weathers, in heavier traffic, much faster, and with *no* accidents[9]. The best robot cars are not even in the same league as most ordinary human drivers.

Nor is the gap is closing, because the 99% barrier cannot be crossed by any simple calculation, whatever the power. The reason is simply that the options are too many to enumerate. Calculating power, by definition, cannot solve incalculable tasks. Productive tasks can only be solved by productive processing, whether the processor is a brain or a super computer[10].

9. A good MTBA (Mean Time Between Accidents) is twenty years, see http://ridingsafely.com/ridingsafely3.html

10. Courtesy of Dmadeo. Copyright: CC-Att-SA-3.

FIGURE 7.5.A: Kim Peek inspired the film Rain Man.

*Courtesy of Dmadeo. Copyright: CC-Att-SA-3 (Creative Commons Attribution-ShareAlike 3.0).*

FIGURE 7.5.B: Dustin Hoffman in the role of Rain Man.

*Copyright © MGM. All Rights Reserved. Used without permission under the Fair Use Doctrine (as permission could not be obtained). See the "Exceptions" section (and subsection "allRightsReserved-UsedWithoutPermission") on the page copyright notice.*[11]

11. Another example is the statement that all people are equal, when diversity is an necessary part of natural selection. Of course, all should have equal rights.

Our brain in its evolution already discovered this and developed beyond simple processing. In *savant syndrome*, people who can calculate 20 digit prime numbers in their head still need full time care to live in society; e.g. Kim Peek (Figure 7.5a), who inspired the movie *Rain Man* (Figure 7.5b), could recall every word on every page of over 9,000 books, including all of Shakespeare and the Bible, but had to be cared for by his father. He was a calculation genius, but as a human was neurologically disabled because the higher parts of his brain did not develop.

Savant syndrome is our brain working without its higher parts. That it calculates *better* means that it already tried simple processing in the past, and moved on. We become like computers if we regress. The brain is not a computer because it is a *different kind of processor*.

## 7.3 A DIFFERENT KIND OF PROCESSOR

To be clear: the brain *is* an information processor. Its estimated hundred billion neurons are biological on/off devices, no different on the information level from the transistors of a computer. Both neurons and transistors are powered by electricity and both allow logic gates (McCulloch & Pitts, 1943) .

So how did the brain solve the "incalculable" productive tasks of language, vision and movement? A hundred billion neurons per head is more than the people in the world, about as many as the stars in our galaxy, or galaxies in our universe, but the solution was not just to increase the number of system parts.

If the brain's processing did depend only on neuron numbers, then computers would indeed soon overtake its performance, but it does not. As explained, general system performance is not just the parts, but also how they connect, i.e. the system architecture. The processing power of the brain depends not only on the number of neurons but also on the interconnections, as each neuron can connect to up to ten thousand others. The geometric combinations of neurons then matches the geometrics of productive tasks.

The architecture of almost all computers today was set by von Neumann, who made certain assumptions to ensure success:

1. *Centralized Control:* Central processing unit (CPU) direction.
2. *Sequential Input:* Input channels are processed in sequence.
3. *Exclusive Output:* Output resources are locked for one use.
4. *Location Based Storage:* Access by memory address.
5. *Input Driven Initiation:* Processing is initiated by input.
6. *No Recursion:* The system does not process itself.

In contrast, the brain's design is not like this at all, e.g. it has no CPU (Sperry & Gazzaniga, 1967). It crossed the 99% barrier using the processing design risks that von Neumann avoided, namely:

1. *Decentralized Control***:** The brain has no centre, or CPU.
2. *Massively Parallel Input*: The optic nerve has a million fibres while a computer parallel port has 25.
3. *Overlaid Output:* Primitive and advanced sub-systems compete for output control, giving the benefits of both.
4. *Connection Based Storage:* As in neural nets.
5. *Process Driven Initiation:* People hypothesize and predict.
6. *Self-processing:* The system changes itself, i.e. learns.

Rather than an inferior biological version of silicon computers, the brain is a *different kind of processor,* designed to respond in real time to ambiguous and incomplete information, with both fast and slow responses, in conditions that alter over time while referencing concepts like "self", "other" and "us".

Computer science avoided the processing of processing as it gave infinite loops, but the brain embraced recursion. It is layer upon layer of link upon link, which currently beggars the specification of mathematical or computer models.

Yet it is precisely recursion that allows symbolism - linking one brain neural assembly (a symbol) to another (a perception). Symbolism in turn was the basis of meaning. It was by processing its own processing that the brain acquired language, mathematics and philosophy.

Yet one can make "super computers" by linking many PC video cards in parallel. If more of the same is better computing, then bigger oxen are better farming! How can AI surpass HI (Human Intelligence) if it is not even going in the same direction?

That computers will soon do what people do is a *big lie*, a statement so ludicrous that it is taken as true[12]. Computers are *electronic savants*, calculation wizards that need minders to survive in any real world situation. If today's computers excel at the sort of calculating the brain outgrew millions of years ago, how are they the future? For computing to attempt human specialties by doing more of the same is not smart but dumb. If today's super-computers are not even in the same processing league as the brain[13], technology utopians are computing traditionalists posing as futurists.

The key to the productivity problem was not more processing but the *processing of processing*. By this risky step, the brain constructs a self, others and a community, the same constructs that both human and computer savants struggle with. So why tie up twenty-million-dollar super-computers to do what brains with millions of years of real life beta-testing already do? Even if we made computers work like the brain, say as neural nets, who is to say they would not inherit the same weaknesses? Why not change the computing goal, from human replacement to human assistant?

12. This is not to say they cannot be, just that to do this would require drastic changes to the current von Neumann architecture.

13. See Kasparov's The Chess Master and the Computer, 2010

## 7.4 THE SOCIO-TECHNICAL VISION

In the socio-technical vision, humanity operates at a level above information technology. Meaning derives from information but it is not information. Meaning relates to information as software relates to hardware. So IT trying to do what people do is like trying to write an Android application by setting physical bit values on and off: it is not feasible even though it is not impossible. Hence computers will be lucky to do what the brain does in a thousand years, let alone forty. The question facing IT is not when it will replace people, but when it will be seen for what it is: just information level processing.

While the hype of smart computing takes centre stage, the real future of computing is quietly happening. Driverless cars are still a dream but reactive cruise control, range sensing and assisted parallel parking are already here (Miller et al., 2006). Computer surgery still struggles but computer supported remote surgery and computer assisted surgery are here today. Robots are still clumsy but people with robot limbs are already better than able. Computer piloted drones are a liability but remotely piloted drones are an asset. Computer generated animations are good, but state-of-the-art animations like Gollum in *Lord of the Rings*, or the *Avatar* animations, are human actors plus computers. Even chess players advised by computers are better than computers alone[14]. The future of computing is happening while we are looking elsewhere.

All the killer applications of the last decade, from email to Facebook, were people doing what they do best and technology doing what it does best, e.g. in email, people create meaning and the software transmits information. It is only when computers try to manage meaning that things go wrong. It is because meaning is not information that people should supervise computers and computers should not control people.

14. A man was looking for his lost keys at night under a lamp post. When asked where he lost them, he replied: 'Over there in the bushes - but the light is better here.' Seeking to find human meaning in information data is the same error.

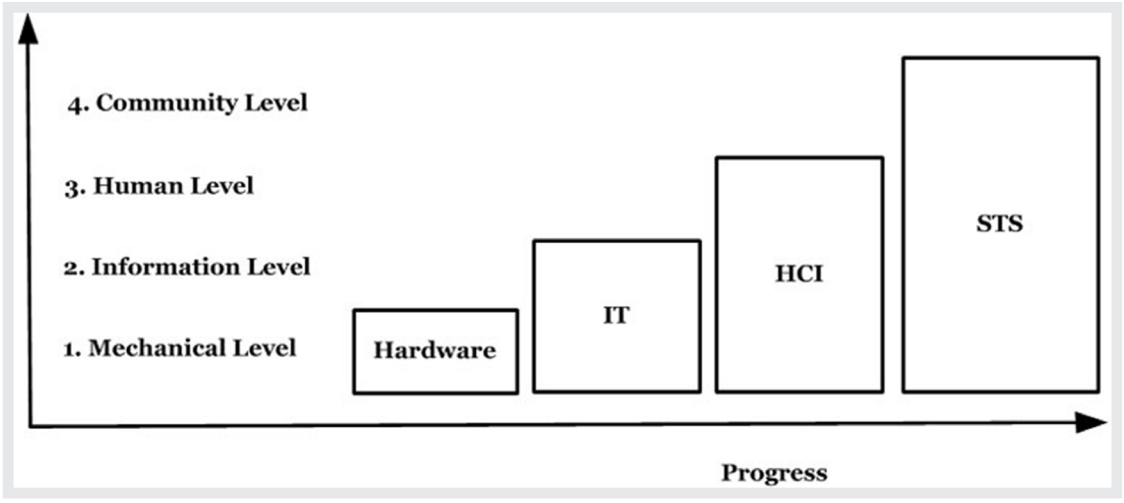



Figure 7.6: The socio-technical vision.

*Courtesy of Brian Whitworth and Adnan Ahmad. Copyright: CC-Att-ND-3 (Creative Commons Attribution-NoDerivs 3.0 Unported).*

When computing becomes background, as in pervasive computing theory, it will work with people as it should. Socio-technical design builds technology to social requirements because higher levels directing lower ones improves performance (Figure 7.6). If people direct technology it *may* go wrong, but if technology directs people it *will* go wrong. Higher requirements directing lower designs is an evolution but the reverse is not. To look for progress in lower levels of operation because they are more obvious makes no sense[15].

To see the Internet as just hardware and software is to underestimate it. Computing is more than that. Technology alone, without a human context, is less than useless — it is pointless. To say "technology is the future" is to pass the future to the mindless and heartless. The future is socio-technology not technology.

All that we know, think or feel, right or wrong, is now online, for everyone to see. Thoughts previously hidden in minds are now visible to all. This is good and bad

15. For example, statements like: 'The internet is full of idiots writing rubbish for other idiots to read.'; or 'The internet is full of idiots and one of them might just be you.'; and 'Do not feed the trolls (DNFTT)'

because thoughts cause deeds as guns fire bullets, so what happens online connects directly to what happens offline, physically. The ideas that precede acts now battle for the hearts and minds of people on the Internet. Virtuality has become reality because although the medium is electronic, the people are real.

Some say the Internet is making us stupid, but it is not. Technology is just a glass that reflects, whether as a telescope or a microscope. Online computer media show us our ignorance, but they do not create it. We must take the credit for that. The Internet is just a giant mirror, reflecting humanity. *It is showing us to us*.

What we see in the electronic mirror is not always pretty, but it is what we are, and teh same mirror also shows selfless-service, empathy and understanding. As has been said, you cannot get out of jail unless you know you are in jail, so humanity cannot change itself until it sees itself. The Internet then, with all its faults, is a necessary step forward in our social evolution. In this view, computing is an integral part of a social evolution that began over four thousand years ago, with the first cities. Now, as then, the future belongs to us, not to the technology. The information technology revolution is leading to new social forms, just as the industrial technology revolution did. Let us all help it succeed.

## 7.5 DISCUSSION QUESTIONS

The following questions are designed to encourage thinking on the chapter and exploring socio-technical cases from the Internet. If you are reading this chapter in a class - either at university or commercial – the questions might be discussed in class first, and then students can choose questions to research in pairs and report back to the next class.

1. What is technology utopianism? Give examples from movies. What is the technology singularity? In this view, computers must take over from people. Why may this not be true?

2. List some technology advances people last century expected by the year 2000? Which ones are still to come? What do people expect robots to be doing by 2050? What is realistic? How do robot achievements like the *Sony dog* rank? How might socio-technical design improve the Sony dog? How does the socio-technical paradigm see robots evolving? Give examples.
3. If super-computers achieve the processing power of one human brain, then are many brains together more intelligent than one? Review the "Madness of Crowds" theory, that people are dumber rather than smarter when together, with examples. Why doesn't adding more programmers to a project always finish it quicker? What, in general, affects whether parts perform better together? Explain why a super computer with as many transistors as the brain has neurons is not necessarily its processing equal.
4. How do today's super computers increase processing power? List the processor cores of the top ten? Which use NVidia PC graphic board cores? How is this power utilized in real computing tasks? How does processing cores operating in sequence or parallel affect performance? How is that decided in practice?
5. Review the current state-of-the-art for automated vehicles, whether car, plane, train, etc. Are any *fully* "pilotless" vehicles currently in use? What about remotely piloted vehicles? When does full computer control work? When doesn't it? (hint: consider active help systems). Give a case when full computer control of a car would be useful. Suggest how computer control of vehicles will evolve, with examples.
6. What is the 99% performance barrier? Why is the last 1% of accuracy a problem for productive tasks? Give examples from language, logic, art, music, poetry, driving and one other. How common are such tasks in the world? How does the brain handle them?
7. What is a human savant? Give examples past and present. What tasks do savants do easily? Can they compete with modern computers? What tasks

do savants find hard? What is the difference between the tasks they can and cannot do? Why do savants need support? If computers are like savants, what support do they need?

8. Find three examples of software that, like Mr. Clippy, thinks it knows best. Give examples of: 1. Acts without asking, 2. Nags, 3. Changes secretly, 4. Causes you work.
9. Think of a personal conflict you would like advice on. Keep it simple and clear. Now try these three options. In each case ask the question the same way:
   - Go to your bedroom alone, put a photo of family member you like on a pillow. Explain and ask the question out loud, then imagine their response.
   - Go to an online computer like http://cleverbot.com/ and do the same.
   - Ring an anonymous help line and do the same.

   Compare and contrast the results. Which was the most helpful?
10. A rational way to decide is to list all the options, assess each one and pick the best. How many options are there for these contests: 1. Checkers, 2. Chess, 3. Civilization (a strategy game), 4. A MMORPG, 5. A debate. Which ones are computers good at? What do people do if they cannot calculate all the options? Can a program do this? How do online gamers rate human and AI opponents? Why? Will this always be so?
11. What is the difference between syntax and semantics in language? What are programs good at in language? Look at text-to-speech systems, like here, or translators here. How successful are they? Are computers doing what people do? At what level is the translating occurring? Are they semantic level transformations? Discuss John Searle's Chinese room thought experiment.

## Your notes and thoughts on chapter 7

Record your notes and thoughts on this chapter. If you want to share these thoughts with others online, go to the bottom of the page at: http://www.interaction-design.org/books/the_social_design_of_technical_systems_2nd_ed/the_future.html

**NOTES:**

# References

## COPYRIGHT TERMS

### Open Access

We believe in Open Access and the **democratization of knowledge**. Unfortunately, world class educational materials are normally hidden behind payment systems or in expensive textbooks. If you want this to change, you should help us out! Kind thoughts are **not** enough - you need to act!

**Copyright Terms**

We do **NOT** use copyright as a restrictive instrument, but as an instrument **to protect the author against misuse while encouraging redistribution and use of his/her work**. As such, these copyright terms are designed for the author and the reader, not the publisher and the profit.

**1. Definitions**

a. **"Adaptation"** means a work based upon the Work, or upon the Work and other pre-existing works, such as a translation, adaptation, derivative work, arrangement of music or other alterations of a literary or artistic work, or phonogram or performance and includes cinematographic adaptations or any other form in which the Work may be recast, transformed, or adapted including in any form recognizably derived from the original, except that a work that constitutes a Collection will not be considered an Adaptation for the purpose of this License. For the avoidance of doubt, where the Work is a musical work, performance or phonogram, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered an Adaptation for the purpose of this License.

b. **"Collection"** means a collection of literary or artistic works, such as encyclopedias and anthologies, or performances, phonograms or broadcasts, or other works or subject matter other than works listed in Section 1(f) below, which, by reason of the selection and arrangement of their contents, constitute intellectual creations, in which the Work is included in its entirety in unmodified form along with one or more other contributions, each constituting separate and independent works in themselves, which together are assembled into a collective whole. A work that constitutes a Collection will not be considered an Adaptation (as defined above) for the purposes of this License.

c. **"Distribute"** means to make available to the public the original and copies of the Work through sale or other transfer of ownership.

d. **"Licensor"** means the individual, individuals, entity or entities that offer(s) the Work under the terms of this License.

e. **"Original Author"** means, in the case of a literary or artistic work, the individual, individuals, entity or entities who created the Work or if no individual or entity can be identified, the publisher; and in addition (i) in the case of a performance the actors, singers, musicians, dancers, and other persons who act, sing, deliver, declaim, play in, interpret or otherwise perform literary or artistic works or expressions of folklore; (ii) in the case of a phonogram the producer being the person or legal entity who first fixes the sounds of a performance or other sounds; and, (iii) in the case of broadcasts, the organization that transmits the broadcast.

f. **"Work"** means the literary and/or artistic work offered under the terms of this License including without limitation any production in the literary, scientific and artistic domain, whatever may be the mode or form of its expression including digital form, such as a book, pamphlet and other writing; a lecture, address, sermon or other work of the same nature; a dramatic or dramatico-musical work; a choreographic work or entertainment in dumb show; a musical composition with or without words; a cinematographic work to which are assimilated works expressed by a process analogous to cinematography; a work of

drawing, painting, architecture, sculpture, engraving or lithography; a photographic work to which are assimilated works expressed by a process analogous to photography; a work of applied art; an illustration, map, plan, sketch or three-dimensional work relative to geography, topography, architecture or science; a performance; a broadcast; a phonogram; a compilation of data to the extent it is protected as a copyrightable work; or a work performed by a variety or circus performer to the extent it is not otherwise considered a literary or artistic work.

g. **"You"** means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

h. **"Publicly Perform"** means to perform public recitations of the Work and to communicate to the public those public recitations, by any means or process, including by wire or wireless means or public digital performances; to make available to the public Works in such a way that members of the public may access these Works from a place and at a place individually chosen by them; to perform the Work to the public by any means or process and the communication to the public of the performances of the Work, including by public digital performance; to broadcast and rebroadcast the Work by any means including signs, sounds or images.

i. **"Reproduce"** means to make copies of the Work by any means including without limitation by sound or visual recordings and the right of fixation and reproducing fixations of the Work, including storage of a protected performance or phonogram in digital form or other electronic medium.

**2. Fair Dealing Rights.** Nothing in this License is intended to reduce, limit, or restrict any uses free from copyright or rights arising from limitations or exceptions that are provided for in connection with the copyright protection under copyright law or other applicable laws.

**3. License Grant.** Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

a. to Reproduce the Work, to incorporate the Work into one or more Collections, and to Reproduce the Work as incorporated in the Collections; and,

b. to Distribute and Publicly Perform the Work including as incorporated in Collections.

c. For the avoidance of doubt:

i. **Non-waivable Compulsory License Schemes**. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme cannot be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License;

ii. **Waivable Compulsory License Schemes**. In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme can be waived, the Licensor waives the exclusive right to collect such royalties for any exercise by You of the rights granted under this License; and,

iii. **Voluntary License Schemes**. The Licensor waives the right to collect royalties, whether individually or, in the event that the Licensor is a member of a collecting society that administers voluntary licensing schemes, via that society, from any exercise by You of the rights granted under this License.

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats, but otherwise you have no rights to make Adaptations. Subject to Section 8(f), all rights not expressly granted by Licensor are hereby reserved.

**4. Restrictions.** The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

a. You may Distribute or Publicly Perform the Work only under the terms of this License. You must include a copy of, or the Uniform Resource Identifier (URI) for, this License with every copy of the Work You Distribute or Publicly Perform. You may not offer or impose any terms on the Work that restrict the terms of this License or the ability of the recipient of the Work to exercise the rights granted to that recipient under the terms of the License. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties with every copy of the Work You Distribute or Publicly Perform. When You Distribute or Publicly Perform the Work, You may not impose any effective technological measures on the Work that restrict the ability of a recipient of the Work from You to exercise the rights granted to that recipient under the terms of the License. This Section 4(a) applies to the Work as incorporated in a Collection, but this does not require the Collection apart from the Work itself to be made subject to the terms of this License. If You create a Collection, upon notice from any Licensor You must, to the extent practicable, remove from the Collection any credit as required by Section 4(b), as requested.

b. If You Distribute, or Publicly Perform the Work or Collections, You must, unless a request has been made pursuant to Section 4(a), keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or if the Original Author and/or Licensor designate another party or parties (e.g., a sponsor institute, publishing entity, journal) for attribution ("Attribution Parties") in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; (ii)

the title of the Work if supplied; (iii) to the extent reasonably practicable, the URI, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work. The credit required by this Section 4(b) may be implemented in any reasonable manner; provided, however, that in the case of a Collection, at a minimum such credit will appear, if a credit for all contributing authors of the Collection appears, then as part of these credits and in a manner at least as prominent as the credits for the other contributing authors. For the avoidance of doubt, You may only use the credit required by this Section for the purpose of attribution in the manner set out above and, by exercising Your rights under this License, You may not implicitly or explicitly assert or imply any connection with, sponsorship or endorsement by the Original Author, Licensor and/or Attribution Parties, as appropriate, of You or Your use of the Work, without the separate, express prior written permission of the Original Author, Licensor and/or Attribution Parties.

c. Except as otherwise agreed in writing by the Licensor or as may be otherwise permitted by applicable law, if You Reproduce, Distribute or Publicly Perform the Work either by itself or as part of any Collections, You must not distort, mutilate, modify or take other derogatory action in relation to the Work which would be prejudicial to the Original Author's honor or reputation.

**5. Representations, Warranties and Disclaimer**

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTIBILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

**6. Limitation on Liability.** EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

**7. Termination**

a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Collections from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

b. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

**8. Miscellaneous**

a. Each time You Distribute or Publicly Perform the Work or a Collection, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.

b. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

c. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.

d. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

e. The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the Protection of Literary and Artistic Works (as amended on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996 and the Universal Copyright Convention (as revised on July 24, 1971). These rights and subject matter take effect in the relevant jurisdiction in which the License terms are sought to be enforced according to the corresponding provisions of the implementation of those treaty provisions in the applicable national law. If the standard suite of rights granted under applicable copyright law includes additional rights not granted under this License, such additional rights are deemed to be included in the License; this License is not intended to restrict the license of any rights under applicable law.

### ii. The Interaction Design Foundation Addendum to the Creative Commons licence

The Interaction Design Foundation Addendum to the Creative Commmons licence is a placeholder for additions to the Creative Commons licence, which are deemed necessary to include in consideration of Danish law and the operation of this site and The Interaction Design Foundation.

**1. Attribution**

If this work is used under the licencing conditions set forth here, attribution must be clearly given, i.e. the author's name, the title and URL of this work/publication/web page must clearly appear. The attribution must be given in a manner appropriate to the medium in which it is given: For example, electronic copies must include a clickable URL, which does not use the nofollow attribute value.

**2. Updates**

Internet technology, publishing technology, and the applicable laws, rules, and regulations change frequently. Accordingly, The Interaction Design Foundation reserves the unilateral right to update, modify, change and alter its Site Terms and Conditions as well as Copyright Terms at any time. All such updates, modifications, changes and alterations are binding on all users and browsers of Interaction-Design.org, readers of electronic and non-eletronic versions of the publications produced by The Interaction Design Foundation. Such updates will be posted on Interaction-Design.org.

### iii. Exceptions

**Exceptions**

Many materials published by The Interaction Design Foundation - both in print and electronically - may contain materials where the copyright is owned by a third party, e.g. another publisher. In this case, the copyright status depends on the third party, i.e. the copyright owner, and may for example be "all rights reserved - used with permission". When this is the case, we clearly label the content. For images, we both write the specific copyright label (including attribution) underneath the caption in both electronic and print copies as well as include the copyright label (including attribution) inside the image file (i.e. the full-resolution version) in metadata types like EXIF, IPTC, and XMP. We only include and label content with the following copyright terms:

1. Pd:

   Public Domain (information that is common property and contains no original authorship)

   Legal Code (full licence text): http://en.wikipedia.org/wiki/Public_domain

2. CompositeWorkWithMultipleCopyrightTerms:

   Work that is derived from or composed of multiple works with varying copyright terms and/or copyright holders

3. FairUse:

   Copyrighted materials that meet the legal criteria for Fair Use when used by the Interaction Design FoundationThe most common cases of Fair Use are: 1) Cover art: Cover art from various items, for identification only in the context of critical commentary of that item (not for identification without critical commentary). 2) Team and corporate logos: For identification. 3) Other promotional material: Posters, programs, billboards, ads: For critical commentary. 4) Film and television screen shots: For critical commentary and discussion of the cinema and television. 5) Screenshots from software products: For critical commentary. 6) Paintings and other works of visual art: For critical commentary, including images illustrative of a particular technique or school. 7) Images with iconic status or historical importance: As subjects of commentary. 8) Images that are themselves subject of commentary.

   Legal Code (full licence text): http://en.wikipedia.org/wiki/Fair_use

4. AllRightsReservedUsedWithoutPermission:

   All Rights Reserved. Non-free, copyrighted materials used without permission. The materials are used without permission of the copyright holder because the materials meet the legal criteria for Fair Use and/or because The Interaction Design Foundation has not been able to contact the copyright holder. The most common cases of Fair Use are: 1) Cover art: Cover art from various items, for identification only in the context of critical commentary of that item (not for identification without critical commentary). 2) Team and corporate logos: For identification. 3) Other promotional material: Posters, programs, billboards, ads: For critical commentary. 4) Film and television screen shots: For critical commentary and discussion of the cinema and television. 5) Screenshots from software products: For critical commentary. 6) Paintings and other works of visual art: For critical commentary, including images illustrative of a particular technique or school. 7) Images with iconic status or historical importance: As subjects of commentary. 8) Images that are themselves subject of commentary.

5. AllRightsReserved:

   All Rights Reserved. Materials used with permission. Permission to use has been granted exclusively to The Interaction Design Foundation and/or the author of the given work/chapter, in which the copyrighted material is used. This permission constitutes a non-transferable license and, as such, only applies to The Interaction Design Foundation. Therefore, no part of this material may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, recording or otherwise without prior written permission of the copyright holder.

6. CC-Att-1:

   Creative Commons Attribution 1.0 Unported

   Legal Code (full licence text): http://creativecommons.org/licenses/by/1.0/

7. CC-Att-3:

   Creative Commons Attribution 3.0 Unported

   Legal Code (full licence text): http://creativecommons.org/licenses/by/3.0/

8. CC-Att-2:

   Creative Commons Attribution 2.0 Unported

   Legal Code (full licence text): http://creativecommons.org/licenses/by/2.0/

9. CC-Att:

   Creative Commons Attribution 3.0 Unported

   Legal Code (full licence text): http://creativecommons.org/licenses/by/3.0/

10. CC-Att-ND-3:

    Creative Commons Attribution-NoDerivs 3.0 Unported

    Legal Code (full licence text): http://creativecommons.org/licenses/by-nd/3.0/

11. CC-Att-ND-2:

    Creative Commons Attribution-NoDerivs 2.0 Unported

    Legal Code (full licence text): http://creativecommons.org/licenses/by-nd/2.0/

12. CC-Att-ND-1:

    Creative Commons Attribution-NoDerivs 1.0 Unported

    Legal Code (full licence text): http://creativecommons.org/licenses/by-nd/1.0/

13. CC-Att-ND:

    Creative Commons Attribution-NoDerivs 3.0 Unported

    Legal Code (full licence text): http://creativecommons.org/licenses/by-nd/3.0/

14. CC-Att-SA-1:

    Creative Commons Attribution-ShareAlike 1.0 Unported

    Legal Code (full licence text): http://creativecommons.org/licenses/by-sa/1.0/

15. CC-Att-NC-SA-3:

    Attribution-NonCommercial-ShareAlike 3.0 Unported

    Legal Code (full licence text): http://creativecommons.org/licenses/by-nc-sa/3.0/

16. CC-Att-SA-3:

    Creative Commons Attribution-ShareAlike 3.0

    Legal Code (full licence text): http://creativecommons.org/licenses/by-sa/3.0/

17. CC-Att-SA-2:

    Creative Commons Attribution-ShareAlike 2.0 Unported

    Legal Code (full licence text): http://creativecommons.org/licenses/by-sa/2.0/

18. CC-Att-SA:

    Creative Commons Attribution-ShareAlike 3.0 Unported

    Legal Code (full licence text): http://creativecommons.org/licenses/by-sa/3.0/

19. Unknown:

    Copyright status unknown

20. Trademarks and logos:

    All trademarks, logos, service marks, collective marks, design rights, personality rights or similar rights that are mentioned, used or cited by The Interaction Design Foundation and its authors are the property of their respective owners. The use of any trademark in our materials does not vest in the author or The Interaction Design Foundation any trademark ownership rights in such trademarks, nor does the use of such trademarks imply any affiliation with or endorsement of The Interaction Design Foundation and its authors by such owners. As such The Interaction Design Foundation can not grant any rights to use any otherwise protected materials. Your use of any such or similar incorporeal property is at your own risk. Words which we have reason to believe constitute trademarks may or may not have been labelled as such. However, neither the presence nor absence of such labels should be regarded as affecting the legal status of any trademarks.

While most material produced by The Interaction Design Foundation is free to use under its respective license as outlined above, some materials may be subject to additional legal restrictions when they are used in particular circumstances or in particular ways. These limitations may arise from laws related to trademarks, patents, personality rights, political censorship, or any of many other legal causes which are entirely independent from the copyright status of the work. For example, if you use a public domain image (i.e. uncopyrighted) of an apple to sell computers, you will violate the trademark rights of Apple Computer, Inc.

In addition, content linked from a page/chapter/book (in the online versions) is not covered by one of our licenses unless specifically noted. For example, pages may link to videos or slide decks that are not covered. The design of Interaction-Design.org (graphics, html, client-side scripts, etc.) is copyright of Mads Soegaard.