LEGITIMATE BY DESIGN: TOWARDS TRUSTED SOCIO-TECHNICAL SYSTEMS[1]

Brian Whitworth (bwhitworth@acm.org) New Jersey Institute of Technology, New Jersey

Aldo de Moor (ademoor@kub.nl) Tilburg University, The Netherlands

Abstract. *Legitimacy or "fairness" seems a key requirement for trust in computer-mediated social environments. Trust in turn seems necessary for productive community interactions like e-commerce. But unless legitimacy is built into social software, achieving trust may not be possible. This means expressing apparently vague social "rights" as specific information system (IS) requirements, i.e. carrying out a legitimacy analysis. We suggest a framework for the systematic analysis of who "owns" what in IS design, assuming basic object types and actions. This analysis not only allows social legitimacy concepts to be expressed in IS design terms, but could also reveal socio-technical system design choices for public review. The technique is illustrated by case examples. Legitimacy analysis can apply to wide variety of social software, from chat rooms to virtual realities. It could lead to future global standards for virtual social environment design, perhaps necessary for the emergence of a global online community.*

1. Introduction

1.1. *Information systems.*

Traditionally, information systems (IS) were *tools* that people *used* to solve technical problems external to themselves (e.g. using a spreadsheet to calculate a budget). People "used" the IS, whose functionality operated upon something outside the user. By contrast many modern multi-user applications, including the Internet, are *social environments,* i.e. they mediate social interaction between people. Now the user is necessarily represented within the software, rather than standing entirely outside it, and the functionality of the IS also operates on that representation. Users, or their representations, *are now used* by the software, as well as them using it. Adding human-human interaction effects to software functionality brings new requirements to software design – social requirements. Each decade, computing has reinvented itself, from commercial computers in the 1970's, to personal computers in the 1980's, to computers as communication tools in the 1990's. This decade seems destined to be decade of social computing where all software is if not groupware, then at least "group aware". IS human factors research has likewise expanded from computer usability (individual), to computer-mediated communication (largely dyads), and now to virtual communities (social groups). While the underlying infrastructure remains an information system, the overall system is now a social one.

---

[1] Published as Whitworth, B., & deMoor, A. (2003). Legitimate by design: Towards socio-technical systems. *Behaviour & Information Technology*, Vol 22, No 1, p31-51.

Although software activity derives from hardware activity, software design (data and program structures) stands apart from hardware design (chips and circuits). While the "higher" system (software) *depends* on the lower system (hardware) to operate, it naturally *directs* hardware activity to meet higher requirements. These software requirements (e.g. for database or network protocols), are of a different order from hardware requirements (e.g. power or heat restrictions). Indeed, the design of hardware, like chips or network components, is often to support desired software information needs. In 1978 Hiltz and Turoff noted that a computer system can also be a social system: "Computerized conferencing systems allow for the design of a complete human social system …" (Hiltz & Turoff, 1993. p418). By social system is here meant the social interaction of people to generate human meaning which directs their activity. This meaning is considered to arise
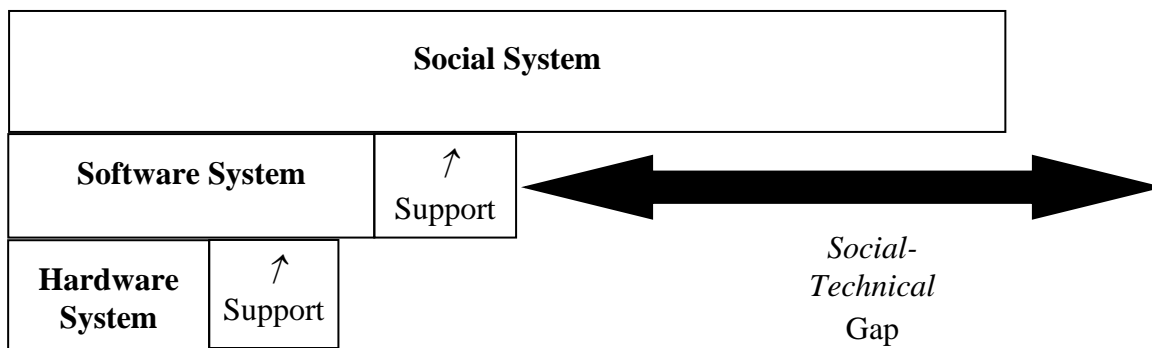

Figure 1. The social-technical gap

from basic human cognitive processes, including those based on the exchange of factual, interpersonal and group information (Whitworth, Gallupe, & McQueen, 2000). In other words, social meaning derives from information exchange, but is not equivalent to it. In computer-mediated social interaction, the social system relates to the software system as the software system relates to the hardware system (see Figure 1). Thus while the social system depends upon the software to operate, it naturally directs software to meet its additional requirements, which are distinct from those of both hardware and software. These requirements must be expressed in terms of social interaction. Talk of a "virtual" community should not be taken to imply that a computer-mediated community is somehow less "real" than a physically mediated one. Rather, we hold that human sociality may extend across both cyber and physical space. Figure 1 suggests that a social system built upon a technical system (rather than the physical world), is still a social system. While the means of interaction, a computer network, is virtual, the people involved are real, as are the social effects.

*1.3.	Social-technical gap.*

A review of a decade of groupware research suggests that the main problem social software faces is the *social-technical gap* - the difference between social needs/expectations and computer system capability (Ackerman, 2000). It is the degree software fails to meet social requirements (see Figure 1). For example Internet privacy concerns seem essentially a conflict between a social requirement

(privacy) and current Internet system design. Solving technical problems, like bandwidth and protocols, will not solve social requirements. We must begin with the social need, and translate it into system design requirements, i.e. make more "sociable" software (Preece, 2000).

1.4. *Legitimacy.*

We propose the general social requirement of *legitimacy* – that dealings between people in a common environment are fair. Other examples are privacy, copyright, censorship, trespass, intellectual property, libel and even virtual "rape". As software becomes more social, legitimacy issues will likely increase, a trend already visible. Almost every issue of the Communications of the ACM over the last five years has had at least one article on privacy, copyright, libel, piracy, trust, trespass, digital signatures, online rights or some other legitimacy issue. Although difficult, such issues are now a fact of life for software designers, and likely to remain so (Lessig, 1999). The problem facing IS designers is a new one, it is not what *can* be done (technical), but what *should* be done (social), what is legitimate rather than what is feasible.

2. Legitimacy

2.1. *Virtual community.*

While current concepts vary (Lee, Vogel, & Limayem, 1992), we take a virtual community to exist when a socially self-sustaining group, with persisting social practices, acts in a common computer-mediated space. Groups are socially self-sustaining when the benefits of social interaction, such as gaining knowledge, making friends, or collective action, make members want to remain in the group for social reasons. A community is not just a task group, that stops when the task is done, or a "rent-a-crowd", that requires payment to meet. Temporary gatherings, like people gathered to view a spectacle, are aggregates rather than communities. A community arises from social benefits, not just economic benefits. A business (or e-business) that sells bargains can't consider its customers a community if they only come for the bargains (economic benefits). The community is not even the people in it at a moment in time, but their ongoing system of social interaction. People may come and go, but the community persists through the group's passed on cultural knowledge, ongoing networks of relationships, and persisting community norms and structures. We see a community as not only the network of interpersonal ties between individuals (Wellman, 2001), but also their collective social structures and norms. While technology may reduce physical constraints, it cannot "liberate" individuals from the social demands of shared living. Every relationship constrains, and every group binds – these bonds are the price of friendship and belonging respectively. A community then, is not just a set of individuals, but *a form of self-sustaining social interaction that endures*.

2.2. *Virtual community environment (VCE).*

A VCE is an information system that supports a virtual community. A VCE is not a virtual community, nor does a VCE's existence guarantee a virtual community will arise. For a bulletin board to support a virtual community requires that lasting and common social practices develop. More general terms are groupware and cyberspace. *Groupware* is any software that supports computer-mediated social interaction, whether communities or not. *Cyberspace* enables computer-mediated interaction of any sort, not necessarily between people.

2.3. Social value.

To be self-sustaining, a community must generate social value for its members (Preece, 2000). This value is the benefits of social interaction, less the costs. A recent model of computer-mediated social interaction proposes it has three cognitive purposes - gaining factual knowledge from others, relating to others (forming friendships), and belonging to action groups (collective action) (Whitworth et al., 2000). This produces, on a group level, informational, personal and normative influences which act upon each group member. The latter allows communities to apply implicit and explicit social norms, which prescribe constraints on communal behaviour that can reduce the social costs of interaction (Stamper, 1994).

2.4. Social cost.

In a common environment, the actions of one person may deny those of another, i.e. there is the potential for *action conflict*. For example when people share the same roads, for one driver to proceed at an intersection, perhaps another must wait. If both parties proceed (seeking an individual advantage) they will collide or "crash" - the environment cannot satisfy both demands a once. Such conflict may damage (or cost) not only the acting parties, but others as well, and may escalate to long term personal vendettas. For the group, the effect is usually a net loss, so social groups internally weakened by action conflicts tend not to last. A stable community requires some way to reduce member conflict.

2.5. Legitimacy.

Developed human societies reduce "lose-lose" conflicts via a complex set of "rights" - common expectations of who can do what, when and where. What is "right" varies over time and between cultures, i.e. legitimacy is a situated, not universal, practice. Even so, if conflicts are resolved by "right" not might, the social group is strengthened. Legitimacy then is a *social perception, common to a group, used to resolve situations of internal action conflict*, usually expressed in terms of what is "right" or "fair", or what is "wrong" or "unfair". Groupware meta-functions like visibility and negotiability seem beneficial because they allow legitimacy to operate and reduce conflict (Wulf & Rohde, 1996). For example visibility allows social queuing, where people agree that arrival order (fairly) determines service order, even though physically one could "jump the queue". Where no reason exists to prefer one party over another, communities create one, e.g. by placing a "Stop" sign at intersecting roads. The case has been well made that justice is an implementation of the concept of fairness, and fairness benefits the group as whole (Rawls, 2001). Psychology studies show people avoid unfair situations (Adams, 1965), and often prefer procedural justice to personal benefits (Lind & Tyler, 1988). Legitimacy seems a social adaptation to a group problem (conflict), in the process of human social evolution towards the formation of large, stable and prosperous communities (Diamond, 1998).

2.6. Law.

Law is the formal expression of legitimacy as group rules designed to resolve action conflicts. Hence law cases always involve two parties in conflict (prosecution and defense). Legitimacy is the social perception that precedes the law, and the sense used to form laws. Judges and juries use it to make precedence decisions, where the law is unclear. Legality (whether actions accord to law) is not equivalent to legitimacy. An action seen as illegitimate may not be illegal if no law exists (e.g. virtual rape). And a legitimate action could be illegal (e.g. a bad law).

2.7. Sanctions.

Sanctions are actions taken on behalf of communities to enforce laws (e.g. by police). A common sanction is group rejection, by banishment or imprisonment. Though legitimacy perceptions precede law, sanctions (and police) are still required, as knowing what is right is not doing it. Individuals may *act* illegitimately, through need or greed, but still *know* their actions are wrong. Those who knowingly commit illegal acts are criminals. They risk their community membership and liberty. But when people feel the law is illegitimate (e.g. suffragettes for women's rights), they are revolutionaries. What is at stake now is the community itself, which may be destabilized if legitimacy is denied.

2.8. Trusted systems.

A trusted social system is one where legitimate rights are implemented (Stefik, 1997). For example, Locke first made explicit the concept that people have a natural right to the fruits of their labor (Locke, 1690). For example, if some people take the trouble to grow flowers, most agree it is not right that passers-by pick them for their benefit without asking. This is not considered "fair". This fairness can be implemented in various ways - by laws and police, by a fence, by norms, or any combination. If successful, and flowers are not stolen, the environment becomes *trusted*, and gardeners find it worthwhile to grow flowers. However if people *unfairly* pick other people's flowers, without permission, gardeners may not to bother to grow them (for others to steal). In this case, neither the gardeners nor the thieves have flowers, i.e. the group as a whole loses. The same logic can be applied to any constructive endeavor in a social setting, suggesting that trust of "the system" is a necessary condition for social productivity.

2.9. Legitimacy as public good.

The above argument, that author ownership benefits the group, has also been well made for privacy, i.e. that *privacy is a public good*, as without privacy, individuals cannot be themselves, and groups need individuals to act freely (Regan, 1995). As Tim Berners-Lee puts it: "*No-one will take part in the new web-like way of working if they do not feel certain that private information will stay private.*" (Berners-Lee, 2000). Hence it has been argued that businesses should welcome legitimate rights like privacy, because they are good for business (Lester, 2001). Progress in legitimacy, (e.g. slavery, human rights, women's rights) seems to go hand in hand with economic prosperity, perhaps because in trusted systems, people contribute, purchase, provide, and generally participate more. Also when systems are legitimate, people self-regulate, and do not have to be forced (by police) to do things (Tyler, 1999), reducing regulatory costs. The core value of legitimacy seems to be that it benefits the community as a whole (Davis, 2001), i.e. *legitimacy increases prosperity*. If trusted online communities are equally critical to generating online community benefits, like "e-business" (Schubert, 2000; Weltry & Becerra-Fernandez, 2001), it follows that legitimacy is equally important to online social interaction.

2.10. Security.

Trusted systems require legitimacy and security. If security ensures that a system is used as intended, legitimacy defines that intent. For example, whether a user is who they say they are (authentication), is a security issue. What rights they *should* have (authority) is a legitimacy issue. In generating trust and business, no amount of security can compensate for a lack of legitimacy, as police states illustrate.

2.11. Summary.

Figure 2 summarizes these concepts. Multiple actors in a common environment increase the
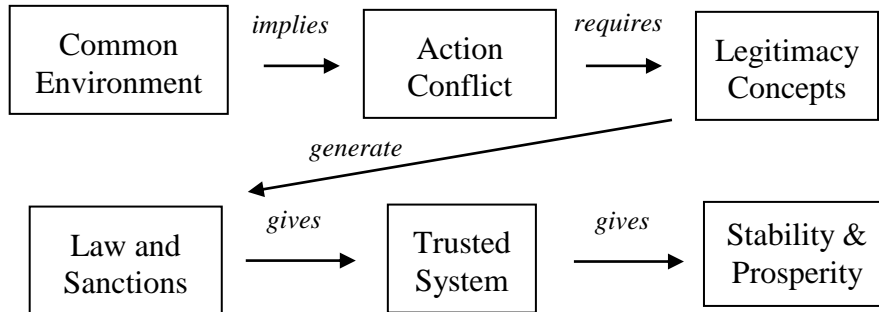

Figure 2. Legitimacy as means of resolving community action conflicts

likelihood of action conflicts. Legitimacy perceptions, implemented by laws, norms, restraints, ethics or sanctions, reduce the likelihood of such conflicts. This gives a more trusted social system, which increases stability and prosperity. Legitimacy is thus a foundation stone of any prosperous and enduring community, and communities that ignore it do so at their peril (Fukuyama, 1992). We propose that the community, not the individual, is the "user" of social software, and legitimacy is one of its "user requirements".

3. The problem

If legitimacy is as critical to virtual as physical communities, *how can a VCE support it?*

3.1. Current situation.

It has been suggested that we are merely "hunter-gatherers in an information age", and that the technology revolution has returned us to socially primitive times (Meyrowitz, 1985, p315). In any environment, people naturally move to valued resources, like food or gold. But unless some *social* value evolves, when the *economic* value departs, the people depart, giving a "ghost town". In cyber-space the economic resource is information, and information "gold-diggers" seem just as fickle. Many online environments, expecting communities to automatically form from social aggregation (Rheingold, 1993), found that after an initial honeymoon, members often drift away, and most struggle to last beyond a year (Rosson, 1999). Virtual community web sites are as often community obituaries ("ghost sites") as active communities.

3.2. Legitimacy is inherent to social interaction.

When communities form, group rules tend to follow. The original Internet design supported a community without rules, a "free" society with no laws, where everyone did as they wished, e.g. authorship was not supported. While traditional painters can embed their signature in their work, cyber-artists cannot. Who would have thought when the Internet started that ownership would become the issue it is today (e.g. Napster)? Copyright was supposed to die, as technology defined social interaction, but the reverse has happened. The Commerce Department's 1995 White Paper suggests we are entering an era when copyright may be more protected than ever before (Lessig,

1999). Rather than social practices following technology, copyright is being reinstated in the online environment (Stefik, 1997). The lesson is clear – if legitimacy requirements prevail, it is better to incorporate them from the beginning. For example if the original Internet architecture had included a public/private data field, authors could have chosen to give or keep copy rights, and browsers could respect that choice (Lau, Etzioni, & Weld, 1999). Acrobat 5.0 now allows a digital signature to be placed in documents, satisfying a legitimacy requirement initially thought unimportant (creator ownership). Legitimacy problems seem inherent to social situations, and when ignored do not go away, but continue to resurface until resolved.

3.3.   Legitimacy issues are widespread.

Today's concern is that software companies will reinstate their own rights (of property) but ignore their customer's rights (of privacy). The Orwellian monitoring and control systems possible in the "brave new worlds" of virtuality could be a "great leap backwards" socially, and privacy groups have sprung up in response. Stephen Manes puts the conflict well: "*'You have zero privacy anyway.' Sun Microsystems' CEO Scot McNealy said last year. 'Get over it.' He's right on the facts, wrong on the attitude*." (Manes, 2000). "The attitude" is the view of some corporates that "Your information belongs to us.". Rather than getting over it, people are more likely to do, eventually, what they always do: demand legitimate rights. An example is the Intel's inclusion of a Processor Serial Number (PSN) in its Pentium III in 1999. Users felt strongly that it was not right for application software to access a unique identity number on their computer without their consent. Privacy, the right to personal information, is a legitimacy issue, and legitimacy prevailed when Intel disabled the feature. Our current performance implementing legitimacy in virtual environments seems at best weak (Privacy-International, 2002), and we now consider why.

3.4.   Virtual environments are unlike the physical world.

Laws in a physical community are expressed in terms of physical actions and concrete objects. They govern what people do, not what they think or feel. Historical law assumes a physical world, constrained by time and space. But virtual environments have significantly different functionality. People cannot physically be in two places at once, but in the electronic world this is as easy as opening two chat room windows. To physically take is to dispossess, but electronically, to take is merely to copy. The virtual world is *a functionally different world*, so it may not be appropriate or even possible to transfer laws from the physical world to an electronic one (Burk, 2001). They must be, and are being, *re-invented*, by re-applying legitimacy concepts to virtual contexts. For example, in the physical world as one flips TV channels or browses a shopping mall, no-one is recording the commercials you watch or the clothes you try on, but Net browser cookie functionality means the equivalent happens to online surfers. Why this makes some people, as PC World describes it, "mad as hell", is that *they do not feel it is right* (Tynan, 2000). Setting a cookie on a user's hard drive seems like a shop slipping their business card in a visitor's pocket, then surreptitiously checking the pockets of everyone who visits. No doubt a law could be passed on cookies. But after cookies, what will be the next new software "feature" requiring legislation? Historical law has evolved over hundreds of years, but cyber-law probably does not have that luxury. Given the rate of invention of new, unprecedented functionality, technology seems likely to outstrip its social assimilation into law, if it has not done so already, and legislation seems destined to be always behind the play.

3.5.   Virtual environments are not like each other.

Virtual worlds differ not only from the physical world, but also from each other, so the virtual environment is not one but many. By analogy, in computer simulation games, where players manage virtual worlds, different functionality requires different rules. For example, flight requires rules of flight, and games with time travel need rules for that. Likewise where virtual environments differ in functionality, they may require different laws. If the law must be reinvented because virtual worlds differ from the physical world, the same applies when virtual worlds differ from each other.

3.6.   Virtual environments can be the law.

In a virtual community, since the software defines what objects and actions are possible, it can pre-empt legitimacy. Software architecture can regulate people's online lives more than any tyrant, and while originally the Net was thought innately ungovernable, and beyond controls, it is now clear it can easily be "turned" the other way - to a system of perfect regulation and control (Lessig, 1999). As Mitchell so clearly puts it, in cyberspace *code is law* (Mitchell, 1995, p111), and virtual environments can be and do internally whatever their designers wish. Currently each VCE designer fashions their virtual rules according to their social world view of what is best or right (Turoff, 1991). But while each designs their virtual society software "in their own image", the images are not the same. By acts of commission and omission, the creators of today's cyber-worlds can implement or nullify any legitimacy concept, e.g. if everyone is anonymous, there is no accountability. If software designers are social designers, they are judge, jury, and executioner on issues of legitimacy. In sum, while implementing legitimacy is essential for stable and prosperous virtual communities:

1. *Virtual worlds are unlike the physical world* - they may require new laws, re-invented from base concepts.

2. *Virtual worlds are unlike each other* - they may each require different new laws.

3. *Virtual worlds can be the law* – community laws may be effectively decided once the system is written, yet designers have no agreed guidelines.

The power of code is a double-edged sword. In socio-technical systems, technical development must be matched by social development to make real progress. Facing this challenge, traditional mechanisms for implementing trusted social systems seem ill-suited. Traditional law is in danger of becoming mostly inapplicable, usually out of date, and easily ignored in practice, given the rate of software innovation and the power of code to define the online social environment. To develop stable online communities, for trade or social interaction, requires a new approach.

4.   Legitimate by design

To treat legitimacy as an ethical problem places the onus on the individual. To treat it as a legal problem places the onus on law makers. To make it a social problem places the onus on social action groups. However once the software is written, and the social environment created, traditional ethical, legal and social forces may be ineffective. Legitimacy has thus also become a system design problem. While system designers may prefer to avoid ethical considerations, for a VCE to be morally neutral may not be an option (Brey, 1999). If social software design affects the *social system it supports*, then social requirements must be recognized *before* the software is created, as well as after (see Figure 3). The software must be designed to support (and not prevent) the desired

goal of stable and productive social interaction, by supporting legitimacy. This seems the only way to avoid social "errors" that can be costly to correct, and difficult to repair.
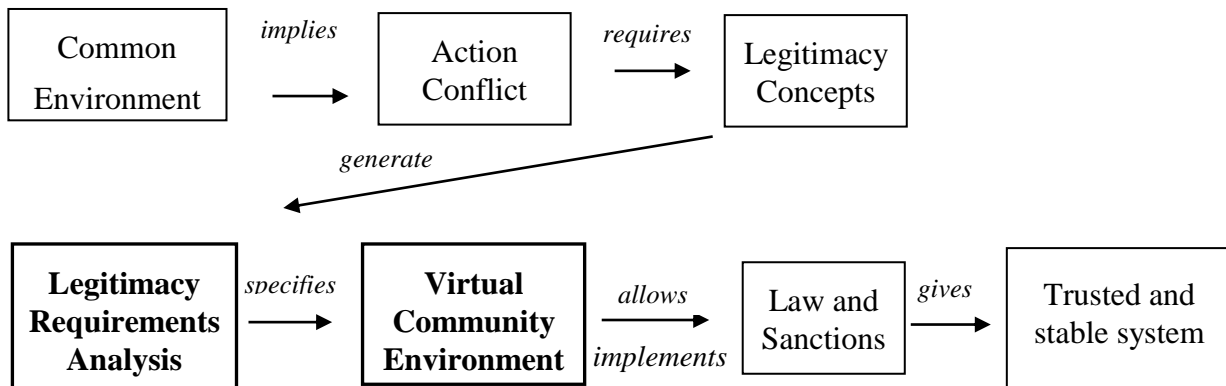

Figure 3. Legitimacy and virtual community design

4.1. *Legitimate by design.*

Making computer systems legitimate by design addresses the problems of the last section. Firstly, socio-technical systems are more likely to be legitimate and trusted if they are designed with legitimacy requirements in mind. Code can support any social right, provided that right is expressed clearly. Secondly, legitimacy requirements, if definable, can set a design pattern for new and future systems, as well as current ones (Stefik, 1997), via generic concepts of what is fair. Expressing basic social rights in software logic terms gives solutions that are not limited to a particular environment. It is then not necessary to "re-invent the social wheel" for each new software product. For example, while for the law cookies are a new problem, in legitimacy terms the issue is a relatively old one, namely: When is it acceptable to record people's activity? Finally, if code can be the law, then applying the source of law (legitimacy) at the source of code (design) seems a good way to obtain socially acceptable electronic systems. Digital audio technology (DAT) gives an example of this approach. DAT allowed unlimited perfect copying of audio tapes, whereas previously every physical tape-to-tape copy was degraded. This was a major threat to copyright. Rather than use police or increasing copying fines, a better solution was to alter the code within DAT machines so that once again every copy is slightly degraded (Reidenberg, 1996). When someone copies a CD too many times, quality degrades significantly, *and this seems fair*. Legitimate solutions to problems created by technology can be built into the "laws" of the technology itself. However by the same logic, they can also be "built out". Some founding principles are required for such operations, and we suggest that legitimacy concepts provide this.

4.2. *Legitimacy requirements.*

IS functionality is essentially program actions upon information objects. All possible actions in a computer system must be defined in the system's logic. In a social system, it is the actions by people that are important. It follows that any electronic social system must fully define who can do what, and hence what is (and is not) legitimate. Currently this is based on common sense and trial and error, which in the expensive business of software development seems inadequate. In social systems, small differences, like who can see what, can have big effects. Social "errors" are cost not

only in rework, but also in reputation. What company wishes to be branded as the creator of unfair software, or as against public good? It seems better to analyze legitimacy requirements beforehand, not just for copy or privacy rights, but for all rights. This requires some way to translate the apparently vague human sense of legitimacy into the precise specifications of an information system. Conversely it means expressing VCE design choices in terms understandable by people in communities. Before suggesting how this can be done, let us consider some cases.

5. Example cases

The following cases illustrate how VCE design choices can create legitimacy problems. The next section suggests how legitimacy analysis could solve such problems.

5.1.   Case 1. Right to own oneself.

In LamdaMOO, a text based virtual reality, a character called Mr. Bungle, (actually a group of NYU undergraduates), acquired the power of "voodoo", the ability to take over the voices and actions of other players (Dibbell, 1993). One night, Mr. Bungle used this power in public to control and violently "rape" several female characters, making them respond as if they enjoyed it. No physical law was broken - there was no physical contact so there was no legal rape. But clearly a fundamental right of social environments was violated - the right to "own" oneself. The LambdaMOO community, committed to non-regulation, was nonetheless outraged. Finally one of the "Wizards" unilaterally deleted the Mr. Bungle character. Many months and 11 ballots later, the system functionality was altered to prevent such outcomes, making "voodoo" an illegal power. The system information processing logic was changed to satisfy a social requirement.

5.2.   Case 2. Right to anonymity.

Lessig describes a thriving online anonymous class discussion that encountered a vicious personal attack on a classmate by a character called IBEX (Lessig, 1999):

"Almost immediately, conversation in the group died. It just stopped. ... Until, that is, the victim responded, with an answer that evinced the wounds of the attack. IBEX's words had cut. The victim was angry and hurt and he attacked back. But his salvo only inspired another round of viciousness ... [and] other members of the class could not resist joining in."

This single person changed a thriving online community into a dying one, as people drifted away, disgusted with what had happened. Most simply left the space. System anonymity effectively gave IBEX the right to hurt others without accountability (and words can hurt as much as deeds). Unable to resolve its internal conflict, the community died, as unlike the LambdaMOO programmers, they could not redesign their environment to solve the problem. In this case the system logic was not able to be changed to fit community social requirements.

5.3.   Case 3. Rights to display.

Who controls objects created within a "space" owned by another? Consider a bulletin board (or chat-room, list server, video-conference etc) created by person P1 for some purpose. Now suppose person P2, who has contributed to the board for some time, adds an item that P1 finds offensive, and P2 refuses to retract it. Does P1 have the right to delete the item? Programming this capability is easy, but is it legitimate? If P1 can destroy the item (and P2's effort in creating it), is not every item potentially deletable, by P1 simply declaring it "offensive"? Surely individuals have some

rights to the items they create? But if so, then can anyone add offensive material to any board? And simply deleting the item may not be enough, as P2 could re-post it. Does P1 then also have the right to delete the persona of P2, preventing them from entering the system? If so, who now owns any items P2 added that were not offensive? Are these now "orphans", or does P1 now own them? And if one persona can be "removed", can anyone be ejected from the system by the board owner? For most current boards, the controller has all these rights and more, i.e. they are effectively designed as dictatorships, albeit presumed benevolent ones. This is unlikely to be a permanent solution.

5.4. Case 4. Rights to view and record.

A VCE can easily "spy" on its inhabitants (see where they go and what they do) because *it mediates everything they do*. Whoever has such rights can be all-seeing and all-knowing in that context, and information is power. In 1998 Comet Systems offered a browser plug-in, the Comet Cursor, that changed the user's cursor on "cursorized" web sites. A media storm arose when a user discovered his browser was sending information packets to Comet Systems, so called "click-stream" data. Users objected to the lack of disclosure, the storing of a unique identity number, and the possible on-selling of private data. Comet changed its site to clearly disclose the information it collects, and now guarantees its customer's privacy. They argued collecting click-stream data is like counting footprints on a beach, and so was legitimate. Opponents argued that assigning each user a unique identity number meant they were potentially individually recognizable. It was recently suggested in a New York Times editorial that Microsoft's Media Player, bundled with Windows XP, quietly records the DVD's it plays and sends the information back to Microsoft (Editor, 2002). For systems like Passport or Hotmail to record user activity, and report it to a centralized database recording millions of people is not technically hard in today's wired world. The issue is whether a private company *should* record what customers do in "its" software, and record this information in a marketing database, *simply because it can*. Cookies seem to add insult to injury. By storing their information on the user's own hard drive the company does not even need a database. There seems something fundamentally wrong with companies using *your* hard drive to store *their* information on *you* without your knowledge or permission. While most browsers now allow users to reject cookies, the default is accept not choose, and users cannot manage their cookies, or review their content, without third party software. Yet equally users like to be personally greeted when they log on to web sites like Amazon. Who can record what on the Internet is currently a confused and complex issue, not technically, but in terms of what it is legitimate.

5.5. Case 5. Property rights.

Most people consider their hard drive belongs to them, so were upset when they discovered that Microsoft's Windows 98 registration secretly gathered all system hardware IDs, as well as the user's name, address etc, and sent the information back to Microsoft. It seemed the electronic equivalent of a company entering your physical house without your knowledge to check the serial numbers of equipment you own. If information can be owned, then this seemed to be information theft. It allowed Microsoft to identify stolen copies of Windows, but also to theoretically track users across open systems like the world wide web, like an electronic "eye in the sky". After this became public, Microsoft agreed to immediately stop the activity, issued a patch to remove hardware registration information from the registry, purged their database of information collected without user permission, and agreed not to include this functionality in any future releases of Windows. The capitulation of technology to social requirements was complete, but at what cost to social trust?

5.6. Case 6. Rights to sub-spaces.

 Legitimacy issues arise if a "super" bulletin board allows members to create new boards within the main board. If P1 owns the main board, and P2 creates a board within it, who has rights to items in the sub-board? Do *both* P1 (the board owner) and P2 (the sub-board owner) have the right to say remove offensive items from P2's board? What if they disagree? And what if P3 creates a board within P2's board which is within P1's board? This is a serious design issue for multi-level boards which attempt to represent hierarchical social structures.

5.7. Case 7. Commenting rights.

Where a source item is commented upon, changing the source may change a comment's meaning. What then happens to the comments? For example an item proposing a bid for $1,000,000 might receive the critical comment: "This is too expensive". If the source item is now changed to a bid for $1,000, the comment seems inappropriate. Yet many systems currently allow such "unfair" changes (e.g. Web-Board).

5.8. Case 8. Conversation rights.

With e-mail, a "private" message addressed to a co-worker can be responded to with a carbon copy to any number of others, and a copy of the original message included. This is the FTF equivalent of every conversant having a tape-recorder and the ability to replay (and broadcast) any prior conversation. Any "private" email can easily become a public broadcast. Yet most people would be upset to see their private e-mails shown publicly, on television for example. Is this legitimacy concern merely a convention that new media will overturn? Probably not. More likely problem cases will resurface the issue until e-mail becomes private to addressees. Until then, e-mail use may be reduced. President Bush's 2001 decision not to use e-mail because he doesn't trust it seems an example.

5.9. Case 9. Group rights.

When groups can decide on courses of action, for example by voting, who owns the vote? Do the voters have rights to the vote outcome they created? For example are they entitled to see the vote outcome? Or can it just go to a controller who may or may not tell the group? If they can see it, can they change it or delete it? If re-voting is allowed, can individuals change their vote individually, or must the group re-vote as a group? And when groups vote, should the first voters be visible to those who have not yet voted? Would this not introduce unfortunate sequence effects (McGuire, Kiesler, & Siegel, 1987)? For example if the first three voted one way, the others might tend to follow that lead. Voting software systems must assign a unique identification to each voter, to check they are entitled to vote, and they don't vote twice. But if the system knows the identity of anonymous voters, can a vote controller find out who voted which way? Can a group decide to make a secret ballot public? Even more unsettling is evidence that common group decision processes can be illegitimate. Arrow, following Borda, showed that simply taking the highest votes, the standard *plurality vote*, can lead to a group choosing the option they least prefer, and rejecting one they more prefer (Arrow, 1963). For example suppose two candidates with similar popular views share 60% of a vote, and a minority view candidate has 40%. The 60% majority will be split, giving the minority view the most votes. The candidate elected will be the one 60% of the group don't want. The general form of this problem is Arrow's Theorem, which states that given more than two choices, it is impossible to guarantee the group's highest vote represents the decisions of its members, i.e. there

is no logical basis for social action. Yet we base the democracy in which we live on the idea that a group action can represent the choices of its members.

6. Legitimacy analysis

Legitimacy problems will not be solved by technology alone. Indeed advances, such as mobile computing, seem more likely to create new legitimacy issues, and compound current ones (Ghosh & Swaminatha, 2001). We now propose that only by formally analysing who owns what, from the beginning of system design, can problems such as those presented in the last section be addressed. The concept of ownership allows legitimacy requirements to be expressed as system design specifications, and system design choices to be expressed in social terms.

6.1. Spaces.

A "social space" (S) is a "mini-environment" which constrains the objects of social interaction. It could be a list, a document, a bulletin board, a drawing area, etc. It is a complex object (one that contains other objects) such that deleting S deletes the objects in it. The system environment (E) is the first S. It seems axiomatic that *all objects must be contained by, and exist within, a space*.

6.2. Persona.

A social environment must contain IS objects called *personae* (P) that *represent* people, who also exist outside the virtual social environment. These could be Hotmail IDs or virtual avatars. People "exist" within the VCE via these personae. Social systems assume *people have choice*, and so are responsible/accountable for their actions to others in the community. Further, people are assumed to have a self-concept, and to value themselves. For example a car that collides with a person, is not accountable, as it has no self-concept, attaches no value to itself, and has no choice - but the car's driver is accountable. Full accountability requires that all actions be traceable back to those that did them. This suggests there is no absolute right to anonymity in a community. A person may be anonymous to others, but must be known to the VCE, to the degree there is accountability.

6.3. Objects.

Since a VCE is an IS, all objects within it are information objects (O). The base processes of social interaction suggest generic social object types (Whitworth et al., 2000). Task information exchange suggests *item* (with displayed factual content), *comment* (semantically dependent item), and *space* objects. Exchanging interpersonal information suggests *persona, mail* (addressed item), *reply* and *conversation* objects. Exchanging group action pattern information suggests *group, membership, voting* and *vote* objects. The basic actions of any information system are create, delete and change. Given a space, one can also move (objects between spaces). Given personae, one can also view, display and travel (e.g. enter a space). From a theoretical model of social interaction, one can derive expected social objects and actions.

6.4. Rights.

A right is *a system permission for a person to carry out some action* upon or with an information object, or any part of it. Declaration of rights, such as the Magna Carta (1297), the French declaration of the Rights of Man and of the Citizen (1789), and the United Nations Universal Declaration of Human Rights (1948), proclaim certain actions to be legitimate, i.e. permitted and/or

supported by the community or social system. The goal of legitimacy analysis is to specify which IS rights are legitimate, and how the software should support them. It means, in system design terms, specifying *who is allowed to do what to what, where, and when*.

6.5. Ownership.

Ownership in IS terms is having rights to actions *on or with an information object*. To "own" an object yet have no rights to act upon it, or use it in any way, is a contradiction in terms. Over twenty years ago Hiltz and Turoff recognized that "The first and foremost issue is that of ownership. Who owns the material entered in a group communication space?" (Hiltz & Turoff, 1993, p505). Legitimacy then, is about who owns what, which in an IS must be specified, or the computer, which mediates all actions, would not know what to do. It follows that *all changeable VCE objects must be owned by one or more people*. Legitimacy then is not about what actions are "right" (ethics), but about who has what choices of action.

6.6. No rules?

An easy system design approach is to allocate all rights to everyone - the so called "no rules" solution. But if everyone has the right to destroy an object, the first to use that right denies it to others, who no longer have the right to say view the object, because it no longer exists. To give rights to all is to deny the rights of most. While no rules is an easy programmer solution, it makes all action conflict resolution an issue of "first in first served", which like "might is right" in the physical world, is not a good solution socially. The inherent logical problem of the no rules solution is avoided if, when an action changes an object, the VCE grants unique rights to a single actor, and legitimacy provides a basis for doing this.

6.7. Dictatorship.

Another easy system design approach is to give all (or most) rights to a "controller" and no or few rights to anyone else. But a godlike controller is not only easily overloaded, but often has rights others consider unfair. The result can be "a community of one". To give all rights to a single controller, or to give all rights to everyone, seem simple solutions that no longer suffice. The alternative we now explore is to *share ownership in legitimate ways*.

6.8. Freedom.

Who should own a persona? In the physical world for one person to own another is slavery, and this is now considered illegitimate. Freedom in a virtual context means an active persona should belong entirely to the person it represents. This implies that one person should not delete, change, use or even view another's persona without their permission. The concept seems simple, yet many information systems ignore it. Implementing this right would preclude Mr. Bungle's "voodoo" power and actions in Case 1. This definition of freedom is not "the right to do whatever you want", but rather the right to choice in self-action. This, as noted earlier, is critical to accountability, and could perhaps better be described as *autonomy*.

6.9. Space ownership.

If one can view inside a space without entering it, it is *transparent,* otherwise it is *opaque*. A space may be *open* or have *restricted* entry, by name or password, as set by the space owner. A space owner with the right to refuse entry could have resolved Case 2 by excluding IBEX. While a

nickname means a person is not accountable in the real world, they are still accountable within the virtual environment, and the sanction of community exclusion remains a powerful one (MacKinnon, 1997). The right to exclude is not equivalent to the right to delete a persona. Indeed to exclude means *not* to delete, as the system must record who was excluded. If only individuals could own objects, every space would be a dictatorship, but groups can also own objects and hence spaces. In Case 2, if the group had owned the space, it could have voted to eject IBEX and done so (had such functionality been available).

6.10.   Delegation and transfer.

Some rights may be voluntarily transferred. Contracts between people are usually records of the transfer of rights. Where a right involves no responsibility for existing information, it can be freely transferred, e.g. the right to create. Otherwise, both parties must agree to the transfer, i.e. the original owner first relinquishes ownership, then the new owner takes it up. Full transfer of ownership (all action rights to an object) is not undoable, as the right to transfer ownership itself is given away. It is reversible in the sense that the new owner can give it back, but no undo button is possible, as once transferred, the original owner has no more rights to the object. However owners may just *delegate* an object, transferring all rights except the right to transfer ownership. In this case, they can take back the object at any time, and the delegatee cannot further transfer ownership, only return it. However some rights, such as freedom, derive from one's existence as a human being, and so seem *inalienable rights*, i.e. they always exist and cannot be transferred. One cannot sell (or buy) freedom, but retain it as long as one is a human being with choice.

6.11.   States.

A state is the set of actions that can be performed upon an object. Let the normal state be "active". In ownership transfer, after an owner relinquishes ownership they cannot change the object because they have relinquished ownership. Nor can the person the item is being given to, as they have not yet taken up ownership. The item is in an "in transfer" state. The only action possible is "take ownership", by either the original or new owner(s). States define *when* actions can occur.

6.12.   Creation.

A created object becomes part of the complex object that contains it, which changes that complex object. Hence creation is *an action upon the space the object is created in,* so the right to create objects in a space belongs initially to the owner of that space. Others can only create in a space only if the space owner delegates that right to them. Since delegated rights can always be withdrawn, this provides another solution to Case 2 – the space owner(s) could withdraw IBEX's creation rights, again given software functionality to do so.

6.13.   Ownership of created objects.

Who owns an object created in a space – its creator or the space owner? Locke held that its creator legitimately owns it. They may give it away or sell it if they choose, but they have first rights to it because they made it. A space owner cannot both delegate and not delegate the right to create, and so *does not automatically own objects created by others within their space*. They can withdraw creation rights, but cannot direct acts of creation by free others who own themselves. While this idea seems simple, many current systems ignore it, allowing system controllers to delete, even edit, items in "their" spaces. Yet most disallow changes of authorship, so space owners cannot claim

responsibility for another's creation, i.e. they recognize attribution rights if not copy rights. It may seem strange that the owner of a space has no necessary rights to the items in it. But consider a building into which people bring personal items. The building owner has no rights to the property of its occupants, such as books, purses or pens.

6.14.   Partial delegation.

If the delegation of creation rights can be partial, space owners can restrict creations in a given space. For example one could delegate the right to create an item's content but not the right to change item properties such as whether the item is anonymous or signed. Hence a space could require that all items added be signed (or that all added items be anonymous). In this case the space owner has delegated creation rights except for the property anonymous/signed, which is not delegated. Equally a space could require created text content to be limited to a certain length, or have a certain format. These limits should be obvious before the item is created, so the creator has informed choice.

6.15.   Display rights.

Displaying an object seems an action of the displayed object on its surrounding space. Any person entering the space can view the object, but it is displayed to the space, whether people are present or not. Where an action involves one object acting on another, it seems sensible that the right resides with both owners jointly. Hence the right to display within a space depends on both item and space owners. An equivalent physical situation is where a notice is placed on a notice board. The notice owner has chosen to display the item in public, but the notice-board owner must also agree. Either the notice owner, or the board owner, may remove the notice. But this does not imply that the board owner owns the notice. He or she may prevent its display, but not destroy or change it, or alter the author. Distinguishing display from edit and delete rights balances the rights of space owners and item contributors, and resolves the problems of Case 3. On a bulletin board, it means an offensive item can be rejected but not actually deleted. Its owner must still have access to it (though others wont).

6.16.   Copy rights.

To grant rights to *view* does not implicitly grant rights to *display* or to *copy*. For example publishers sell the right to view certain information, but not to copy and re-display it, which is "piracy" (with attribution) or plagiarism (without attribution). Many creators on the Web willingly grant others the right to copy and redisplay, but not to re-sell, or to plagiarize. They wish to give view rights, but not edit, display, copy or attribution rights. A computer system that supported this would have to first record the document owner, and secondly what document rights were delegated. The software could block say copying, but allow messages to the owner requesting permission to copy. The goal is not to restrict use, but to provide choice - willing authors could consciously declare documents public, freeing honest users from copyright concerns. Copyright is not about restricting use, but about creator choice.

6.17.   Privacy.

We propose that *viewing an object is an action.* Perhaps because the viewer is outside the IS, viewing is often overlooked as an action. The action of viewing is unique in *changing nothing but the viewer*. Other actions like edit or delete alter objects within the information system. But

knowing one is being viewed energizes the viewed party, an effect called social facilitation (Geen & Gange, 1983). This is expected for social beings whose daily existence depends upon how the group sees them. Naturally people wish to control how they are seen. In communities, the legitimacy concept of "privacy" defines socially acceptable viewing. It implies that for information to be private, it must be about a person. Information that does not identify the individual is not subject to privacy. For example a sports match turnstile that counts each person's entry raises no privacy issues, nor does a web counter doing the same at a web site. However information that uniquely identifies the individual becomes "personal", and so could be private. The collection of data by Comet in Case 4 could have been avoided had they simply not recorded data that identified individuals (as personae or real world identities).

6.18.   Informed consent.

If privacy means that a person has the right to *display* information about themselves, it seems an extension of freedom (Johnson, 2001, p120), i.e. since one owns the object that is oneself, one has the right to display that object (oneself) - or not. US Supreme Court Justice Louis Brandeis described privacy as the "right to be left alone", i.e. *the right of a person to not be viewed*. This implies that people should only be viewed by their choice. Stated another way, *individuals grant others the right to view them*, i.e. they have an element of choice. Clearly if one does not know that one is being viewed, there cannot be any choice. To *consent* to be viewed, one must be *informed* that one is being viewed. Thus informed consent (to be viewed) is a prerequisite of privacy. Hence what annoyed people the most about Comet's activities was they didn't know that they were being viewed (Case 4). Cookies provide the same problem if people do not know what is being recorded about them. By "value-sensitive" design, browsers can implement legitimacy requirements like informed consent in cookie deposits (Friedman, Howe, & Felten, 2002).

6.19.   Private property.

If the object and space owner are one and the same, for example a person in their home, this is private property. To secretly view someone in their home is to be a "peeping Tom", and is an invasion of privacy. To take things one does not own from a place one does not own is to be a thief. In VCE design, private places should be declared as such to the information system. Any actions in a private place are then by owner permission only. Common publicly owned spaces, like public showers, may also be *assumed private* to whoever temporarily uses them. For example an e-mail, before sending is presumed private, even if the unsent e-mail is saved to a common disk area. A trustworthy online e-mail system would ensure that an unsent e-mail that is saved remained private. In Case 5, a legitimacy analysis could have avoided the problem in the system design stage, by establishing that Microsoft was taking without permission information that it did not own, from an area it did not own. Microsoft's social error cost them not just software repair time, but also lost reputation, which, in the long run, may prove by far the greater cost.

6.20.   Public spaces.

When a person, by choice, enters a shared public area, such as a public park, they implicitly consent to be viewed because they view others. Privacy is not violated, as there is implied informed consent. A principle of *visibility equity* seems to operate – that if one party can see another, the other should, in fairness, also be able to see them. In the physical world this is generally so, as lines of sight work both ways. In a shared online space, this means that if anyone is visible, everyone should be visible.

If one wishes to be invisible, then one has no right to see anyone else. To be like the "invisible man", able to see others without being seen oneself, is not a legitimate social arrangement. On e-mail list servers, such as ISWorld, participants are often invisible – one does not know who the viewers are. But equally others cannot see you, so the situation is fair. An exception is the space owner, who has the right to see whoever enters, because they own the space.

6.21. *Public space monitoring.*

Interestingly, people may accept being viewed anonymously, even in public spaces, if the purpose is public good, the viewing is done openly, and how the information is used is known – for example building security cameras in plain view. Simply being viewed, even in a one-way fashion, may not be a privacy concern if there is informed consent. Case studies of work place computer-based monitoring find that acceptance or not varies with the social context (George, 1996). However people tend to object to hidden cameras, not for public good, without information use limits. They object to being "spied on" without their knowledge, as they wish the right to control the display of personal information. In VCE design, any public space monitoring should be declared before entry, as should how the information collected will be used, so people can consent (or not) before entering the space.

6.22. *Right to record.*

Any action in an information system can be recorded, which record is then an object in the system, raising the issue of who owns that object (the record). Where the record is of a person's actions, this seems to involve two prior principles. Firstly, the record creator could own it, as they created it (creation rights). But if the record is of personal actions, the person recorded also has rights to it (privacy rights). In a private space, privacy seems to have priority. Thus one cannot record a person in a private place, just as one cannot view them, without consent. But in public place one has already consented to be viewed. Thus one can photograph a person in a public place without their permission. The record simply allows the recorder to "re-view" the original (legitimate) view at a later date. Where one has no right to view, equally there is no right to record. But a record allows the recorder to display as well as view it. Since granting *view* rights does not grant *display* rights, even for a photograph taken in a public place, magazine publishers generally ask permission of the person photographed before printing the photo. We conclude that a recorder does not unilaterally own any private information they gather. Celebrities and "public" figures seem an exception - perhaps they have granted generic display rights by their declared business of being viewed. For online recording then, the same principles apply as for online viewing, with one addition – private information cannot be re-used, in the sense of being re-displayed or transmitted to other parties, without informed consent. This implies people should know about databases that hold their personal information, and be able to withdraw it from them. For example "spam" lists that use computer power generate unsolicited messages to large numbers of people who mostly do not want to receive them. Spam is currently a major productivity problem for people on the web, as important messages are diluted by an ever increasing flow of solicitations. Applying the principle of informed consent would reduce spam, as people could remove themselves from such lists. Again the result would be a public benefit and increased productivity for the group.

6.23.   Spaces within spaces.

Who can create spaces within a space? Suppose P2 can create a space S2 within a space S1 owned by P1, and P1 ≠ P2. P1 owns the content of S1 but has delegated the right to create objects in it. P2 created and owns S2, so can create objects in it. Now suppose P1 takes back the delegated right to create objects in S1. The content of S1 should no longer be changeable by others. However since P2 still owns S2, P2 still has the right to create objects in S2, which alters the content of S1 (because S2 is a part of S1). *Hence only a space owner can create a space within their space* (which space they may *then* delegate to another). They cannot transfer ownership of a sub-space, or delegate the right to create sub-spaces, without in effect losing ownership of the space. P1 has no rights to objects within S2 while P2 owns it. But P1 can take back the delegated ownership of S2 at any time, which returns ownership of S2 to P1. P2 is responsible *for* S2, but responsible *to* P1. This resolves the problems of Case 6.

6.24.   Comments.

Items intended to be semantically dependent are *comments or responses.* The meaning of a comment object, O2, depends upon the meaning of a *source* object, O1. To process O2 correctly one must first process O1, e.g. the comment "Nonsense!" may depend on the statement "2+2=5". Full semantic dependency means if O1 is destroyed, O2 must be destroyed, and if O1 is changed, the meaning of O2 becomes indeterminate (possibly invalid). Non-comment items may also be semantically dependent, but are presumed able to stand alone. For comments, this presumption is reversed. The existence of comments suggests that the source object be retained after an edit, for example as a previous edit *version*. The comment remains attached to its original source, until the author *reconfirms* it also carries forward to the new, edited, version. Comments should not be deleted simply because their source is edited, though if a source item is deleted, any attached comments must also be deleted, by virtue of their dependency. Such contextual display rights resolve the issues of Case 7.

6.25.   Conversations.

While an idea may have one author, conversations have two or more, suggesting mutual information rights. However e-mail currently gives all rights to the receiver, ignoring sender rights to display only to the person addressed (Case 8). Software design could change this, e.g. if senders could permanently record "Personal for: xxxxxx" within an e-mail. Interestingly, other conversational conventions are strictly held to, e.g. that one cannot "edit" past messages. Yet in a spoken conversation, that what is said cannot be unsaid is just a property of the medium (airwaves). In the electronic medium it would be easy to allow message editing - simply overwrite the previously sent message. But this would contradict the current assumption that the receiver "owns" the sent message merely because the transmission system works a certain way. But imagine an e-mail system where the message remains on the sender machine, and only a link is sent to the receiver(s). Message ownership could then remain with the message sender, who would retain the ability to edit or delete "sent" messages at any time, and could also restrict who can access the link to addressees. In this case, as a recipient may reply to a message that is later edited, previous versions must be stored, as for the comments case discussed earlier. The current architecture of e-mail is not a given, it is just one possible alternative. Considering the rights of both senders and receivers suggests alternative designs that could resolve the problems of Case 8, and Zix Mail is an example (ZixMail, 2001).

6.26.   Group action.

Many current systems involve groups, but give them no rights as acting entities. Yet the democracy we live in is based on the legitimacy of the group as an actor. For example, if a group votes on an issue but does not see the results (which go to some controller), the controller owns the vote. Imagine a democracy where voters could not see the results of their vote! A group that owns itself can be said to be autonomous or free. Extending Locke's principle of author ownership to groups suggests that *autonomous groups own their own vote information because they created it.* Hence they have the right to see the results of a completed vote, which the computer system should provide automatically. Many current voting systems do not work this way. Group autonomy also means that if the group chooses to vote anonymously, then no-one, not even a system controller, should be able to know who voted which way (the VCE itself must know, to avoid repeat voting). If a group owns its voting activity, it can also change it, for example by removing the anonymity. This would reveal who voted which way, but to the entire group. In the physical world, voting is such an expensive process, in time, collecting and calculating the vote, that large groups generally elect representatives to act for them, but virtual groups could act directly, using computer power to calculate vote results for every decision (Whitworth & McQueen, 1999). Virtual groups could be more, not less, democratic than physical ones.

6.27.   Free speech.

The idea that the decision of the group should *represent* the decisions of its members, that voting represents the "will of the people", is a legitimacy concept. One aspect is that each group member's views are fairly included in the discourse process by which the group decides to act. The right to free speech seems the right to have a "voice" in group decision making. If so, free speech applies primarily to group action decisions, i.e. political discussion. It is not simply the right to say whatever you want, any more than freedom is the right to do whatever you want. Groups may censor or restrict communications deemed harmful to the community without denying free speech.

6.28.   Representation.

In a group decision, typically each group member gets one "vote". This seems based on the view that as social beings, people are equal. Some liberal and thoughtful people have proposed that the group would be better served if the better people had more votes. For example John Stuart Mill advocated multiple votes for the better educated (Mill, 1948). Yet communities have retained one person one vote, where people have any vote at all. The reason may be that group decisions are as much about accountability as capability. Social equality clearly does not mean that we are all equally capable, but rather equally accountable. When a group decides, we all get one vote because we all bear the consequences, and who can say that their state is of more value than another?

6.29.   Voting.

To decide, a group must first decide how it will decide, and this is certainly the stuff of revolutions. This includes defining the minimum number that can decide for the group (the *quorum),* the *decision percentage* (usually 51%), and how a vote is called (usually to put a *motion* requires two people, a proposer and a "seconder", but it can be more). But potentially the most contentious issue is how individual choices should be combined to give the group choice. Arrow's paradox shows that the plurality vote method, such as used in U.S. Senate elections, can be unrepresentative for decisions with more than two choices. Computer power now suggests solutions not feasible for

physical voting. For example voters can rank all choices, though this involves more voter effort, and the computer can use second or third etc choices to calculate the most representative choice. Another more efficient way is to *drop the least preferred choice(s) and vote again* if there is no group majority. Eventually this leads to a binary choice where Arrow's paradox would not occur, barring unlikely perfectly even splits. Even in the latter case, the group could revote until a majority occurred, e.g. when the Cardinals elect a new Pope, they re-ballot until a decision is made. Computer voting, by largely removing the work of distributing, collecting and calculating votes, could avoid the problems of Case 9, by either rank ordering candidates, or allowing re-voting. If there were errors, as in the recent U.S. presidential election, it could simply be done again. To allow groups and communities to act, decide, and own objects seems to be the challenge of the next generation of groupware (Whitworth, Gallupe, & McQueen, 2001).

6.30.   Unity of action.

Re-voting raises sequencing issues if members who have not yet voted can see the votes of others who voted before them. In physical elections, usually no results are released until all parties have voted (lest earlier voters influence later ones). The principle here seems to be one of unity of action, i.e. *an acting entity must operate as a unity when it decides to act*. It must act or not act, and a vote is an action by the group. Hence all individual votes must be completed before the results are revealed to the group. Individuals thus should not change their vote individually - the group should re-vote as a group or not at all. By contrast, in a group discussion people can speak in any sequence order, because they are exchanging information as individuals, and the group is not acting. For a legitimate online democratic vote to elect a President or Prime Minister, the computer voting system would have to be trusted not to reveal voting patterns to *any voting community member* until the vote was done. Voting systems that are transparent to their controllers do not fulfill this requirement, and we know of no electronic voting where the group owns its own vote in the democratic sense described here. If the next revolution after e-commerce is e-government (Bos, 1993; Symonds, 2000), there still seems some way to go before it is a social reality.

7.   An illustrative example

To illustrate these points, consider an imaginary case involving Attila, a kindly bulletin board owner, and Luke, an independent board contributor. Suppose Luke adds an item Attila finds offensive or inappropriate, but Luke disagrees. What can happen? More importantly, what *should* happen? Is Luke free to say whatever he wants? Or can Attila simply delete the item forthwith because it is his board? Can Attila edit the item to remove the offensive part? Can he withdraw Luke's right to create items? Can he require that all Luke's future postings be checked by him before being made public? Can he ban Luke from entering the board? Can Luke publicly withdraw his board membership? Can he withdraw the item? If he does, can Attila still make the posting public, to warn others against "this sort of thing"? Can he set the board to "watch" Luke, and keep a log of all his activities? If Luke mails him a private apology, can he publish it on the board? Can he alter Luke's name on present or future items to "Gross-Luke" until he learns a lesson? Can Luke "fix" the item and resubmit it? Can Luke change his name to Arthur and rejoin the board and do it again? Can Attila delete Luke from the entire system? If so does that delete all Luke's items? If so, does that delete all comments on his items by other people? Suppose the added item was anonymous, can Attila find out who wrote it? Can Attila require all items be signed in future? If so, can he change the item's anonymity so everyone sees that Luke is the author? Suppose Luke added

the "offensive" item to a sub-board of Attila's board, run by Ghengis, can Attila still delete the item, monitor Luke etc? What if Ghengis does not find the item offensive? If Luke is excluded from Attila's board, can he still enter Ghengis's board? Can Luke start a "revolution" to usurp Attila as board controller, by public vote? If successful, can Luke then delete Attila's items, ban Attila, etc, in revenge? To all these questions, and many others, for a programmer the answer is "whatever you want". Social software can be written to allow any functionality whatsoever. If design is driven solely by company requirements, operation will tend to company benefit, rather than community benefit. If the user community is to benefit, then the community requires legitimacy. Clearly not all the above options are legitimate. But which? Our previous, and admittedly initial, *legitimacy analysis* suggests that:

1. Luke owns the "offensive" item, so Attila should be unable to delete/edit it, or change its anonymity.

2. Luke owns his own persona, so Attila should be unable to delete/edit it, or change Luke's name.

3. Attila owns the board or space – so Attila should be able to withdraw display rights (reject), withdraw rights to create, check new items before display, and refuse entry rights.

4. Luke owns his item, so even if it is rejected he should be able to see it, edit it, and request it be reconsidered for display by Attila.

5. Attila owns the space, so can set as a creation condition that items be signed (not anonymous) – but if Luke's item is already created, this change should not affect it.

6. Luke did not grant Attila publishing rights, so Attila should not display Luke's item elsewhere.

7. Attila should not be able to record Luke's activity without his knowledge.

8. Luke has no right to community anonymity, so the system environment can allocate Luke's persona a unique internal ID. Even if he changes his name, the persona's ID is excluded, so he should not be able to rejoin under another name.

9. Luke should be able to withdraw from the board, and for a visible or public space, board members should be able to see he has withdrawn.

10. If Luke adds to the Ghengis sub-board, only Ghengis owns that space and rights to it. Attila has no rights inside Ghengis's board. However as Attila delegated the sub-board to Ghengis, he can take it back, and *then* reject the item or exclude Luke. But Ghengis may be unwilling to re-own sub-board again after being so "deposed".

11. If Luke is excluded from Attila's board, he should be automatically excluded from Ghengis's sub-board.

12. If Attila owns the board, Luke cannot depose him, but if Attila's ownership is delegated, Luke could appeal to the higher board owner. If the board community owned the board, and Attila was merely its representative, given support, Luke could propose no confidence in Attila, and could be elected to replace him. Conversely the group could reject Luke and his item.

13. If the group owns itself, Luke as a member should be able to contribute to any discussion on group action, and to vote.

Supporting legitimate rights allows more positive social interaction. For example if Luke still owns his rejected item, he should still be able to see it, and know it is rejected. So he can amend it, and

ask again that it be displayed in Attila's space. By contrast, if it were simply deleted, he might assume an error (on his part or the system's) and resubmit it, unaware it was "rejected", increasing the conflict with Attila. However if, knowing it was rejected, he resubmits it, he knows he may risk exclusion for example. Carefully following legitimacy concepts gives a better designed social interaction, but such functionality must be built in from the ground level.

7.1.   System ownership.

This analysis implies that all ownership is ultimately delegated from the owner of the entire system, the social environment people operate within. While it is logical that a system owner owns everything in it, such absolute power seems extreme. Human history suggests that absolute power corrupts, whether held by individuals or corporates. Legislation against monopolies reflects the view that private companies or individuals should not own community environments. Traditional competitive models fail when considering environments, because environments cannot "compete" (without existing in a higher environment). This issue is at the center of current anti-trust legislation against Microsoft, as it is clear that whoever monopolizes browser software, in effect, *owns the Internet*. That no-one owns or controls the Internet was at first seen as a weakness, but now many consider precisely that lack of centralized control its greatest strength. But since the World Wide Web is created by the software we write, the issue of ownership cannot be avoided. People must own it because people made it – but which people? History suggests that, like Tolkien's "one ring" in the Lord of the Rings story, individuals or companies that take power over environments are inevitably corrupted, and ultimately destroyed, by that power. *The solution our social history suggests is community ownership of environments*. Companies like Microsoft, who essentially create online environments, could "gift" them to the community, and operate them on behalf of the community, not in name but in fact. While this at first sight seems ridiculous, equally ridiculous is the idea that human communities will accept being run by a private corporate entity. The human desire for freedom or autonomy is probably too strong for that. Simply put, Microsoft, or any other corporate, is part of the community, rather than vice-versa. It follows that if software companies build community environments, they should be accountable to those communities for their design. To an extent this already occurs, but only via the intermediary of the press and media, not directly (Case 5). That an online social environment belongs to the community that uses it, rather than the private companies that created the software, seems a fundamental right, equivalent to an individuals right to freedom.

7.2.   Beyond online dictatorships.

Can people or groups really own environments? Surely there must be an overall controller to "fix" things? For example, if an offensive item is "rejected" but not deleted, would it not stay on the system forever? And if someone is excluded, or leaves a space and never returns, must they forever stay on the membership list in the name of "freedom"? Such problems can be resolved by the idea that all system objects have a "life", set at their moment of creation. This can be a space creation condition, and so be reset. This is currently the case for spaces like Microsoft's Hotmail, where created e-mail personae are automatically deleted after a certain time period if not used. There are no complaints because the life conditions are stated when the ID is created. A similar system can work for items, allowing for example people to add items to an online advertiser knowing that after one day say, their items will automatically expire (just as in effect occurs for newspaper advertisements). Systems can be designed to reduce the need for intervention. The idea that a space

must have an all powerful controller to run it equates to the idea that a community must have an all powerful King or Emperor to run it. Just as many people hundreds of years ago felt the latter was inevitable, so today many programmers assume an online bulletin board must be a dictatorship. But democracy is the idea that leaders represent the people, rather than the other way around. If physical communities can establish democratic social systems that operate without all powerful controllers, so can online communities.

7.3. *Terrorism.*

If all rights derive from the community, they can be withdrawn from those that oppose the community, either to destroy it (terrorists or community enemies), or take advantage of it (criminals). In physical communities, individual rights like privacy can be over-ridden if criminal or seditious activity is suspected. The state incarceration of criminals and enemies of the state shows that rights are not absolute, but delegated from the community, and so can be revoked. But the community must first *show to itself* (the group) the facts of the case, hence the idea of a fair and public trial. But with for example the current terrorist scare, can a community, or its representatives, legitimately revoke citizen's rights in the name of security? Again the answer seems to be yes, except that in a democratic group that purportedly owns itself, the same citizens whose rights are revoked should be represented in the decision to revoke those rights. To deny the privacy rights of a thousand good citizens to (possibly) find one terrorist reduces the public good those rights embody. The community is made less by such actions – less cooperation, less trust, and less community good. The degree of threat may justify this loss, but an alternative is to enlist people's cooperation, rather than to treat all citizens as potential enemies by removing their rights.

Concept	Owner	Action(s)	Virtual Right
Freedom	Person represented	Destroy, change	To control their persona
Privacy	Person represented	Display	To control personal information display
Property	Object owner	Change, view, destroy, display	To act upon the information object owned
Contract	Object owner	Transfer, delegate	To transfer/delegate all or some rights
Patent	Object creator	Create	To initially own a created object
Copyright	Item owner	Display	To display an owned item
Attribution	Object creator	Display author	To attach/display item authorship
Trespass	Space owner	Exclude	To control space entry
Sub-Letting	Space owner	Create sub-space	To own a sub-space
Publishing	Space owner	Display in a space	To display objects in a space
Context rights	Comment owner	Display in context	To display in an assumed context
Informed consent	Person represented	View	To know if being viewed or recorded
Representation	Group member	Vote	To contribute to group action
Free speech	Group member	Display content	To contribute to group discussion
Democracy	Group	Group actions	Of a group to own its actions

Table 1. Selected legitimacy concepts and IS rights

8.   A symbolic logic of rights

8.1.   Symbolic description of rights.

We propose that the apparently vague concepts of legitimacy can be expressed in the formal way necessary for computer systems, i.e. that *a symbolic description of legitimacy is possible*. Table 1 summarizes how some currently accepted legitimacy concepts can be translated into information system object and action design specifications. If legitimacy is formally describable, it can be supported through code. The specifications are surprisingly general, and cover a wide variety of cases, from e-mail to multi-user dungeons. However the issues involved are complex, as what is envisioned is not a linear, closed set of absolute rights. Rights can be delegated and taken back, objects may contain other objects, objects may be dependent, rights may interact with other rights, there may be rights to change rights, and groups may have rights. Legitimacy involves much more than can be expressed in this short paper, or perhaps even written down at all. The possibilities seem endless, and in their confluence and overlap, rights may require clarification and prioritization. In physical communities, this is done by selecting 12 citizens to represent the community – a jury. A judge exists not to control them, but advise of previous legitimacy decisions. In the online setting the same could be done, except there is no practical need to restrict the number to 12, nor even to a

single country. Issues of what is and is not legitimate online could be presented to community representatives for formal consideration, under guidance. Such community validation of social software requires that its design be expressed in a way that ordinary users *understand.* The idea of information object ownership seems to fulfill that requirement. It is something people find meaningful and interesting. But what, a skeptic might ask, are the chances of the online community agreeing even on a group decision process, let alone any legitimacy issue? To this one can only reply, what are the chances of the people of America, or Europe, or China agreeing on common systems of justice? To the degree that this is possible for countries, it is possible for the global online community.

8.2. Symbolic logic of rights.

We suggest that rights are not only describable to the public and software designers, but also display an internal consistency or logic. It seems important to people that parts of this logic do not contradict other parts, and further it seems surprisingly parsimonious. We therefore propose that it is possible to develop a *symbolic logic of rights – a theoretical description of legitimacy that is internally consistent.* But can social concepts like privacy and freedom be logically expressed? Legitimacy concepts vary between different local communities. They also seem to involve recursion, and so are probably not fully definable (for example how can a group decide how to decide?). But a useful logical system need be neither complete nor absolute. For example, geometry is neither complete (fully defined), nor absolute (e.g. different assumptions lead to non-Euclidian geometries) (Hofstadter, 1999), nor as Gödel showed, is Arithmetic (Gödel, 1962). Yet Arithmetic and Euclidian geometry are both logical and useful despite this. Hence that legitimacy is not fully definable, or that it varies between communities, are not arguments against it being logical or useful. Few would argue that society's laws are complete, and clearly laws vary between societies, yet laws are invariably both used and valued (Rawls, 2001). We should not assume that social concepts like legitimacy are illogical or unspecifiable because they are neither complete nor absolute. If legitimacy can be understood and applied by people in ways that are largely consistent, and juries and judges suggest this is so, then, in theory at least, it can be specified as a formal and internally consistent logical system. A legitimacy logic could draw conclusions, given basic assumptions or axioms, in a wide variety of cases. It could also clarify the base assumptions, and their implications, for consideration by communities.

8.3. Incompleteness.

But if legitimacy, by its complexity of interactions, is not completely definable, surely code, which is completely defined, cannot contain it? This is probably so, just as laws express rather than contain legitimacy. But a right may be *supported* without being enforced, e.g. by developing socially "translucent" systems, where what one does is (legitimately) visible to others (Erickson & Kellog, 2000). Making people in virtual communities visible and accountable to each other may, in the long run, be how legitimacy is established online, rather than by a complete definition of what is and is not legitimate.

9. Conclusions

While the issues of cooperative living are old, new technology has increased the stakes. It has increased the scale of information gathering possible, the types of information that can be gathered,

the permanence of that information, and the scope of its possible distribution (Johnson, 2001). For the first time in human history, we can change not only the world we work within, but its laws of operation. It is as if we could alter the laws of physics in the physical world. The normal restraints of social interaction have been drastically reduced, and incapability is no longer a basis for right action. Nuclear technology did the same for international relationships – countries either found social alternatives to war or destroyed each other. Computer technology is magnifying the need to address social issues, as increasingly the social systems of countries, communities and corporates become intimately bound to its operation, as virtual society becomes simply "our society".

The absolute power of code in virtual social environments presents us with a choice: to implement legitimacy concepts, acquired through an often painful physical history, or to ignore them at our peril. Legitimate action is not inherent to individuals, as the manifest injustices of our history suggest. But the development of legitimacy does seem part of our social evolution, of our learning on a community level. And these lessons have been hard won. If current software design returns us socially to a time before the American and French revolutions, to before the British Parliament, how long it will take to re-learn the same social lessons? And since these virtual worlds already intimately connect to our physical state, at what cost? New technologies may not solve social problems, but they do provide new opportunities to address them. It would be a tragedy to have to learn again in virtual environments social lessons already learned in the physical world.

If legitimate action is not instinctive to people, what drives it? The answer seems to be that legitimate communities prosper, and those based on unfair or corrupt practices do not (Transparency-International, 2001). Communities restrain illegitimate acts, because the benefit is on a community level, not an individual one. Justice is "blind" because it does not distinguish individuals, but treats us all the same. While individuals may seek an unfair advantage, the inescapable logic of a community is that its members cannot all have an "unfair advantage" over each other. Studies of prisoner's dilemma problems suggest that when two people each seek such an unfair advantage over the other, both lose and the overall utility (public good) reduces (Poundstone, 1992). The "tragedy of the commons" concept suggests the same (Hardin, 1968). While legitimacy seems at first an abstract individual issue of morals and ethics, the concept of social adaptiveness makes it a pragmatic one. For communities, legitimacy is what distinguishes those that endure and prosper from those that don't. In every case described here, legitimacy has either prevailed by the power of the community, or the community has failed. No company, however profit orientated, wishes a reputation as the company that profits at public expense (if only for the effect on their profit). The public good is a common thread to all the legitimacy issues discussed. Freedom is a public good, creator ownership is a public good, privacy is a public good, and for groups to represent the wishes of their members is a public good. This explains how views on legitimacy can change – what is in the public good may not always be apparent. For example it is still not obvious to many that free speech is a public good, nor was it obvious to many in the past that freedom of women or slaves was a public good. The legitimate delegation of rights, the granting of rightful autonomy, increases social performance. People, on average, gain by community membership rather than lose. But if a community is more than its current members, community benefit is not equivalent to "the greatest good for the greatest number". For example, to poison or contaminate land for profit is not legitimate, even if the profit is shared with everyone, because it deprives future community members of usable land.

The concept of community good can guide legitimacy decisions in new situations. For example it is now suggested in the U.S. that creator ownership rights be handed down through generations, rather than creations becoming public property after their creator dies. But while recognizing creator rights encourages creativity, extending ownership to their non-creative descendants would more likely stifle it (Lessig, 1999). If the works of Shakespeare or Beethoven were still copyright, many current books, shows and films would not be possible. Placing works in the public domain after the creator's death allows them to be creatively used by all. This suggests again that all online works should have a copyright "life" equal to that of their creator.

The issue of legitimacy is not any easy one, but it is not one software designers can ignore. As Berners-Lee says: "… technologists cannot simply leave the social and ethical questions to other people, because the technology directly affects these matters." (Berners-Lee, 2000, p124). In the social information age we now enter, social requirements like legitimacy will be fundamental to socio-technical system design. For social software, *the user is the community, and legitimacy is the user's requirement*. A good case can be made that legitimacy is a new dimension of software requirements engineering (de Moor & Jeusfeld, 2001). Those who create social environments have a responsibility to the communities they support that goes beyond that of a software tool creator. A social environment should not be a Pandora's box, whose contents only become apparent when opened. If the global reach of the Internet is to lead to a global community, then that community may need to agree on basic online social rights before it can flourish. If we wish trusted and prosperous online communities, we can no longer afford to design social environments in a social-theoretical vacuum, relying on common sense and good will. It is time for socio-technical systems to become *legitimate by design*

10. References

Ackerman, M. S. (2000). The intellectual challenge of CSCW: The gap between social requirements and technical feasibility. *Human Computer Interaction, 15*, 179-203.

Adams, J. S. (1965). Inequity in Social Exchange. In L. Berkowitz (Ed.), *Advances in Experimental Social Psychology* (Vol. 2, pp. 267-299): Academic Press, New York.

Arrow, K. (1963). *Social Choice and Individual Values* (2nd ed.): Yale University Press.

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

Bos, E. (1993). Can information technology improve the quality of democracy? *Behaviour and Information Technology, 12*(3), 194-195.

Brey, P. (1999). The ethics of representation and action in virtual reality. *Ethics and Information Technology, 1*(1), 5-14.

Burk, D. L. (2001). Copyrightable functions and patentable speech. *Communications of the ACM, 44, February*(2), 69-75.

Davis, R. (2001). The digital dilemma. *Communications of the ACM, February/44*(3), 77-83.

de Moor, A., & Jeusfeld, M. A. (2001). Making Workflow Change Acceptable. *Requirements Engineering, 6*(2), 75-96.

Diamond, J. (1998). *Guns, Germs and Steel*.: Vintage.

Dibbell, J. (1993). A rape in cyberspace. *Village Voice, 36, 37*(December 23).

Editor. (2002, Sunday, Feb 24, section 4). Technology threats to privacy. *New York Times,* pp. 12.

Erickson, T., & Kellog, W. (2000). Social translucence: An approach to designing systems that support social processes. *ACM Transactions on Computer-Human Interaction, 7*(1, March), 59-83.

Friedman, B., Howe, D. C., & Felten, E. (2002). *Informed Consent in the Mozilla Browser: Implementing Value-Sensitive Design.* Paper presented at the Hawaii International Conference on the System Sciences, Hawaii.

Fukuyama, F. (1992). *The End of History and the Last Man*. New York: Avon Books Inc.

Geen, R. G., & Gange, J. J. (1983). Social facilitation: Drive theory and beyond. In A. P. V. K. M. D. H. H. Blumberg; Hare (Ed.), *Small Groups and Social Interaction* (Vol. 1, pp. 141-153).

George, J. F. (1996). Computer-based monitoring: Common perceptions and empirical results. *MIS Quarterly, December*, 459-480.

Ghosh, A. K., & Swaminatha, T. M. (2001). Software security and privacy risks in mobile e-commerce. *Communications of the ACM, February/44*(2), 51-57.

Gödel, K. (1962). *On Formally Undecidable Propositions*. New York.

Hardin, G. (1968). The tragedy of the commons. *Science, 162*, 1243-1248.

Hiltz, S. R., & Turoff, M. (1993). *The Network Nation: Human communication via computer* (Revised edition (from 1978) ed.). Cambridge: MIT Press.

Hofstadter, D. R. (1999). *Godel, Escher, Bach: An eternal golden braid*. New York: Basic Books.

Johnson, D. G. (2001). *Computer Ethics*. Upper Saddle River, New Jersey: Prentice-Hall.

Lau, T., Etzioni, O., & Weld, D. (1999). Privacy interfaces for information management. *Communications of the ACM, 42*(10), 89-94.

Lee, F. S. L., Vogel, D., & Limayem, M. (1992). *Virtual Community Informatics: What We Know and What We Need to Know.* Paper presented at the Hawaii International Conference on the System Sciences, Hawaii.

Lessig, L. (1999). *Code and other laws of cyberspace*. New York: Basic Books.

Lester, T. (2001). The reinvention of privacy. *The Atlantic Monthly, March*, 27-37.

Lind, E. A., & Tyler, T. R. (1988). *The Social Psychology of Procedural Justice*.: Plenum Press, new York.

Locke, J. (1690). *Second treatise of civil government* (Vol. Chapter 5, section 27).

MacKinnon, R. C. (1997). Punishing the Persona. In S. G. Jones (Ed.), *Virtual Culture: Identity and Communication in Cyber Society* (pp. 262). Thousand Oaks, CA: Sage.

Manes, S. (2000). Private Lives? Not Ours! *PC World, June*, 312.

McGuire, T. W., Kiesler, S., & Siegel, J. (1987). Group and computer-mediated discussion effects in risk decision making. *Journal of Personality and Social Psychology, 52*(5), 917-930.

Meyrowitz, J. (1985). *No Sense of Place: The impact of electronic media on social behavior*. New York: Oxford University Press.

Mill, J. S. (1948). *Considerations on Representative Government*. Oxford: Basil Blackwell.

Mitchell, W. J. (1995). *City of Bits Space, Place and the Infobahn*. Cambridge, MA: MIT Press.

Poundstone, W. (1992). *Prisoner's Dilemma*. New York: Doubleday, Anchor.

Preece, J. (2000). *Online Communities: Designing Usability, Supporting Sociability*. Chichester, England: John Wiley &Sons.

Privacy-International. (2002). *Big Brother Awards International*. Available: http://www.bigbrother.awards.at/org/ [2002.

Rawls, J. (2001). *Justice as Fairness*. Cambridge, MA: Harvard University Press.

Regan, P. (1995). *Legislating privacy, technology, social values and public policy*. Chapel Hill, NC: University of North Carolina Press.

Reidenberg, J. R. (1996). Governing networks and rule making in cyberspace. *Emery Law Journal, 45*, 911.

Rheingold, H. (1993). *The Virtual Community: Homesteading on the Electronic Frontier*. Reading, MA: Addison-Wesley.

Rosson, M. B. (1999). *I Get By With a Little Help From My Cyber-Friends: Sharing Stories of Good and Bad Times on the Web.* Paper presented at the Proceedings of the 32nd Hawaii International Conference on System Sciences, Hawaii.

Schubert, P. (2000). *The pivotal role of community building in electronic commerce.* Paper presented at the Proceedings of the 33rd Hawaii International Conference on System Sciences, Hawaii.

Stamper, R. (1994). Social norms in requirements analysis: an outline of MEASUR,, *Requirements Engineering: Technical and Social Aspects* (pp. 107-139): Academic Press.

Stefik, M. (1997). Trusted systems. *Scientific American, March*, 78.

Symonds, M. (2000). The next revolution: After e-commerce, get ready for e-government. *Economist, 24*(June).

Transparency-International. (2001). *Corruption Perceptions*. Available: www.transparency.org [2002.

Turoff, M. (1991). Computer-mediated communication requirements for group support. *Journal of Organizational Computing, 1*, 85-113.

Tyler, T. (1999, October 14-16). *Deference to group authorities: Resource and identity motivations for legitimacy.* Paper presented at the Society of Experimental Social Psychology Annual Conference, St Louis, Missouri.

Tynan, D. (2000). Privacy 2000: In Web we Trust? *PC World, June*, 103-116.

Wellman, B. (2001). Physical place and cyberplace: The rise of personalized networking. *International Journal of Urban and Regional Research, 25*.

Weltry, B., & Becerra-Fernandez, I. (2001). Managing trust and commitment in collaborative supply chain relationships. *Communications of the ACM, June/44*(6), 67-73.

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

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

Whitworth, B., & McQueen, R. J. (1999). *Voting before discussing: Computer voting as social communication.* Paper presented at the Proceedings of the 32nd Hawaii International Conference on System Sciences, Hawaii.

Wulf, V., & Rohde, M. (1996). Reducing conflicts in groupware: Metafunctions and their empirical evaluation. *Behavior & Information Technology, 15*(6), 339-351.

ZixMail. (2001). *www.zixit.com*.