Towards a Theory of Online Social Rights[1]

Brian Whitworth[1], Aldo de Moor[2], and Tong Liu[1]

[1]Massey University (Albany), Auckland, New Zealand

[2]STARLab,Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussels, Belgium

Abstract. Legitimacy, defined as fairness plus public good, is a proposed necessary online and physical community requirement. As Fukuyama notes, legitimate societies tend to prosper, while others ignore legitimacy at their peril. Online communities are social-technical systems (STS), built upon social requirements as well as technical ones like bandwidth. As technical problems are increasingly solved, social problems like spam rise in relevance. If software can do almost anything in cyberspace, there is still the challenge of what should it do? Guidelines are needed. We suggest that online communities could decide information rights as communities decide physical action rights, by a legitimacy analysis. This requires a framework to specify social rights in information terms. To bridge the social-technical gap, between what communities want and technology does, rights must be translated into information terms. Our framework has four elements: information actors (people, groups, agents), information objects (persona, containers, items, comments, mail, votes), information methods (create, delete, edit, view, move, display, transfer and delegate), and the information context. The conclusions apply to any social-technical community, and we apply the framework to the case of Wikipedia.

Introduction

It has been proposed that legitimacy is necessary for social productivity whether based on electronic or physical media [10], where legitimacy is defin0ed as fairness plus social good. In this view email, chat, bulletin boards and groupware are *social-technical systems* (STS), i.e. social systems that overlay technical ones, where “technical” includes both software and hardware aspects. IS theory suggests information systems have different levels: Grudin suggests three: hardware, software and cognitive [4], Kuutti adds a work processes level [5], and Alter suggests hardware, software, people, and business processes [2]. Table 1 shows four STS levels, each a “view” of the same system, not different systems. Each level “emerges” from the previous, as information/data derives from mechanics, cognitive meaning has an information base, and community norms arise from human cognitions [11]. Information here is used as Shannon and Weaver originally defined the term [9], and equates to what business calls “data”. Higher levels assume lower levels, so a level

[1] Whitworth, B., Aldo de Moor, A. and Liu, T., 2006, *Towards a Theory of Online Social Rights*, International Workshop on Community Informatics (COMINF'06) Nov 2 - Nov 3, Montpellier, France

Appears in R. Meersman, Z. Tari, P. Herrero et al. (Eds.): OTM Workshops 2006, LNCS 4277, pp. 247 – 256, 2006. Springer-Verlag Berlin Heidelberg 2006

failure implies failure at levels above it, e.g. if hardware fails, software does too, as does the user interface, but a program can fail as software level but still function as hardware. Why not reduce everything to hardware? Describing computing by chip and line events is as inefficient as describing World War II in terms of atoms and electrons. Higher "holistic" levels increase performance, so communities can generate enormous productivity. STS design must reflect social requirements, lest online "civilization" become a stone-age culture built on space-age technology.

Translating acceptable principles of social interaction into STS specifications can bridge the social-technical gap, between what communities want and what technology does. STS designers must know what to do in information terms, based on legitimacy requirements, expressed as "rights". Just as a physical community uses ownership to

Table 1. Information system levels

Level	Examples	Error	Discipline
Social/ Cultural	Norms, culture, laws, sanctions, social roles	Unfairness	Sociology
Personal/ Cognitive	Semantics, attitudes, beliefs, opinions, ideas, ethics	Misunderstanding	Psychology
Information/ Data	Software, data, bandwidth, memory, processing	Infinite Loop	Computing
Mechanical/ Physical	Hardware, computer, telephone, FAX, voltages, heat	Overheating	Engineering

express physical rights, so an online community can use information ownership to express information rights [10]. Legitimate rights like freedom increase social productivity but can do so at the cost of social stability. This social problem has not one but many "solutions", e.g. there are forms of democracy. Yet while social issues have no perfect answer, not all social forms have equal outcomes, as in general more legitimacy is better than less. We propose a general framework to specify online "rights", as these are the design building blocks of a healthy online community. We hope others find our framework useful to build upon, modify or even contradict.

Criteria

For a software designer to implement social rules they must be:

1. *Complete:* The computer knows what to do in every case.
2. *Consistent*: The rules are easy to program and apply to new cases.

An online community in contrast desires interaction rules that are legitimate:

1. *Fair*: That the STS is impartial to individual actors, treating them as it were from behind a veil [8]. Justice is blind to individuals, i.e. does not favor specific actors. The social goal is social productivity, not personal gain (which is corruption).

2. *Socially beneficial*: A legitimate STS improves the public good. People tend to cooperate with legitimate communities and oppose illegitimate ones.

Rather than argue what past communities agree, we assume the following:

I. Social entities should be accountable for their acts to the community, which may apply sanctions like banishment.

II. To be accountable, social entities must be identified (not anonymous) to that community. Privacy gives the right to be anonymous *to others*, not to the community itself. A society can record data on birth, marriage and death etc, as necessary to identify its members.

Specification

How software architecture allocates information rights defines the social options of a virtual community [6]. We now explore STS rights given the earlier criteria of completeness and consistency (for the computer) and legitimacy (for the community).

STS primitives

We define the primitives of an STS as social actors, objects and methods (Table 2).

Social Actors. Social actors are people or groups who are accountable to society for their acts. We use "actors" rather than "users" to stress their ability to be accountable. Social agents are actors who act on behalf of another social entity, and can be people or automata, e.g. application installation programs are automated agents for a software company. Independently acting automated entities, however intelligent, are actors but not social actors. They are not accountable, and have no "self" to feel social sanctions. The STS objects that represent social actors can be called *persona*. A persona can be an online "handle", not a real name, but must be unique. As physical identities make people accountable in that world, persona identities make people accountable in virtual worlds, e.g. one can be banished from an online game. A persona represents when a person "exists" or is active within the STS. When a person "logs on" with userid and password, the STS then sets an active session for them until they "leave". We call personas "people", though they just represent them. The basic social actors in Table 2 are people, groups and agents.

Information Objects. Since an STS is an IS, it is an information object (O), as are the objects within it. It can contain *items* (I) whose main purpose is to convey meaning, defined as the cognitive processing evoked in people. If an item's meaning is dependent upon another source item, it is a *comment item* (IC). Items transmitted between people communicating are *mail* items (IM), and items whose meaning is only choice information can be called *votes* (IV). A *container* (C) is a complex object that can contain other objects, like an item list. Objects exist within containers, and the

STS environment is the first C. If a container is destroyed any objects within it are also destroyed, as the existence of O's in C depend on C existing. A C may contain another C1, in which case C1's objects are also part of C. The core information objects proposed are persona, containers, items, comments, mail and votes (Table 2).

Table 2. Social-Technical System Components

Actors	Objects	Methods
People (exist outside the STS)	*Persona* (represent people)	*Create/Delete/Undelete*
Groups (composed of people)	*Containers* (contain objects)	*Edit/Revert*
Agents (for people/groups)	*Items* (convey meaning)	*Archive/Unarchive*
	Comments (dependent meaning)	*View/Hide*
	Mail (transmit meaning)	*Move/Undo*
	Votes (choice meaning)	*Display/Reject*
		Join/Resign
		Include/Exclude
	Rights	*Transfer*
		Delegate/Undelegate

Social Methods. The actions social actors can apply to information objects include create, delete and edit, and most have an inverse. Some involve two objects, e.g. move changes an object's container, and can be enter or exit.

STS Rights

Using the components of Table 2, one can define a "right" as follows:

$$\textbf{Right} = \textbf{R}\ (\text{Actor}_i, \text{Object}_i\ , \text{Method}_i), \text{ or } \textbf{R}_i = \textbf{R}(A_i, O_i, M_i)$$

Roles, such as "owner", are sets of rights, e.g. to "own" an object is the right to all actions on or with that object:

$$\textbf{Right}_{\textbf{Owner}} = \textbf{R}(\text{Actor}_{\text{Owner}}, \text{Object}_{\text{Owned}}, \text{Method}_{\text{All}})$$

Rights Errors. A rights "error" occurs when a party allocated a valid right is unable to exercise it, e.g. when multiple actors have same object rights one party's rights can deny another's. If many actors can delete an item, if one person genuinely deletes it then it is gone, so the others have lost their choice to delete or not. The first actor abrogates the rights of others. A simple way to avoid rights errors is for one person to own everything (dictatorship), but this is not fair. In contrast, that everyone owns everything (anarchy) is fair, but invites rights errors and conflict. The middle path between anarchy and dictatorship, that most modern societies pursue, seems based on distributing ownership, and that is the approach we take here. Multiple ownership is complex, as people can act severally (where any can act for all), e.g. husband and wife who trust each other, or act jointly (where all must consent to act), or act democratically (where a majority prevail), or a combination, e.g. democratically elect a leader to act for the group. Ownership can also be passed back and forth, as in joint document writing.

Transfer. Rights built from Table 2's actors, methods and objects are also information objects in themselves, and shown as second tier objects, subject to meta-actions like transfer and delegation. Transfer changes an object's owner. A right that incurs no existing information responsibility can be transferred in one step, e.g. a right to view that incurs no viewer obligations may just be given. In contrast, if a transfer incurs information responsibility, both parties must agree in a two-step process:

a. The owner relinquishes ownership (and may designate the next owner)

b. The new owner takes up the ownership.

The new owner must agree if the new ownership involves accountability.

Delegation. Transfer gives all action rights to the target, so is non-reversible, but delegation transfers all object rights except the right to change rights, so is reversible, i.e. the owner can take back ownership at any time. The delegatee cannot further transfer ownership, as they have no right to transfer rights, e.g. loaning a book to another gives no right to loan it to a third party. Note that while people can (and do) do this, the issue of whether they should do it is a separate issue.

Object state. An object's state defines the action set that can be performed upon it, with the normal state as "active". In ownership transfer, when owners relinquish ownership they cannot still change the item. In the "given away" state the only action possible is "take ownership", by either the original or new owner(s). When in transfer or delegation, no acts are allowed except take ownership, which returns the object to the "active" state. Another state example is "archived", where no edits can occur but view is still possible, e.g. journal publication is a copyright transfer of ownership followed by an archive state, with viewing but no further edit changes allowed.

Creation Rights

The STS components of Table 2 can define many classes of STS rights, including creation, display, view, comment and group rights, and [10] provides many examples. Space does not permit us to review all these classes here. However, we can briefly discuss creation rights, given the others can be defined analogously.

Object Creation. It is reasonable to assume the initial owner of a created object is its creator, as without them the object would not exist [7]. However in an online setting, where does the right to create come from? An information object comprises various fixed data attributes that must be known before it is created, i.e. it is an instance of a prior general form. If to create an object its general form must be known, the form information must be stored somewhere prior to creation. It cannot be in the object, which is not yet created, so must be in the object's container (or its container, etc, up to the STS). All objects are thus created using information from the object(s) that contain them, Also creating an object also changes the container it becomes part of.

Hence it is reasonable to see object creation as an act upon the container the object is created in, which implies the container owner has the right to create objects in it:

$\mathbf{Right}_{\mathrm{CreateObject}} = \mathbf{R}(\mathrm{Actor}_{\mathrm{ContainerOwner}}, \mathrm{Object}_{\mathrm{Container}}, \mathrm{Method}_{\mathrm{Create}})$

Persona Creation. If the STS itself is a container, its owner has the right to create all objects within it. Since persona can act throughout the STS, they seem objects created upon the STS itself, i.e.:

$\mathbf{Right}_{\mathrm{CreatePersona}} = \mathbf{R}(\mathrm{Actor}_{\mathrm{STSOwner}}, \mathrm{Object}_{\mathrm{STS}}, \mathrm{Method}_{\mathrm{CreatePersona}})$

In this case, the STS owner would also own the persona created. However the right to freedom suggests that the person a persona represents should own it.

$\mathbf{Right}_{\mathrm{Freedom}} = \mathbf{R}(\mathrm{Actor}_{\mathrm{PersonRepresented}}, \mathrm{Object}_{\mathrm{Persona}}, \mathrm{Method}_{\mathrm{All}})$

While people normally own what they create (property rights), the right to freedom suggests people should own their online persona selves, which should not be owned by others (slavery). This rights conflict is resolved if the STS owner creates a persona, then transfers its ownership to the person concerned, as many mailing lists do. Systems like Hotmail delegate persona creation, letting entrants self-create persona, as the following right involves no responsibility for existing information:

$\mathbf{Right}_{\mathrm{CreatePersona}} = \mathbf{R}(\mathrm{Actor}_{\mathrm{STSEntrant}}, \mathrm{Object}_{\mathrm{STS}}, \mathrm{Method}_{\mathrm{CreatePersona}})$

If an object owner has all rights to it, this includes the right to destroy it, so one should be able to delete one's persona, e.g. a hotmail-id. However this assumes all transactions are complete, else one could uses an online persona to commit say credit card fraud, then "vanish" into thin air by deleting ones persona.

Item Creation. A container owner may delegate their right to create items to people who enter their container, as bulletin board owners let members create items in them. The right can be given freely as no accountability is implied. To create objects within C one may need to "enter" the container, which may be open entry or restricted by a password, equivalent to a door passkey. The delegated right is:

$\mathbf{Right}_{\mathrm{CreateItem}} = \mathbf{R}(\mathrm{Actor}_{\mathrm{ContainerEntrant}}, \mathrm{Object}_{\mathrm{Container}}, \mathrm{Method}_{\mathrm{CreateItem}})$

When an item is added to a list, is it owned by its creator or the list (container) owner, given they are not the same? In the first case, only its creator can delete it, and in the second, only the bulletin board owner can. We propose the initial object owner is always its creator, however the container owner can prevent its display to others (except of course its creator who can always see it) [10].

Creation Constraints. If creation within a container is delegated from the container's owner, the latter can delegate in degrees, i.e. a container may constrain object creation in any way, e.g. a list may require all items be signed. Such constraints should be evident at creation time, giving item creators informed choice. Creation constraints apply only at the moment of creation, so are not retrospective, e.g. if a list allowed anonymous contributions, then required that all contributions be signed, existing anonymous items need not have signatures. The changed creation condition applies only to creations after the change, but if an anonymous item owner wanted to edit it, the new edited item must then be signed (or the edit cancelled).

Creation constraints illustrate a *rights context*, where a right is limited by a contextual right. The delegated right to, say, add an item to a bulletin board can be written:

$\mathbf{Right}_{\text{CreateItem}} = \mathbf{R}(\text{Actor}_{\text{C-Entrant}}, \text{Object}_{\text{Container}}, \text{Method}_{\text{Create}}, \text{Context}_{\text{Constraint}})$

Wikipedia Ownership: A Rights Analysis

To illustrate a rights analysis we consider Wikipedia, since it is fairly successful, and its philosophy that no-one need own anything is a good test case. Wikipedia holds that all content in Wikipedia is owned by all Wikipedians, apparently currently numbering over 1.9 million. While this is public ownership rather than no ownership, it suggests all rights specifications reduce to a single statement:

$$\mathbf{Right}_{\text{All}} = \mathbf{R}(\text{Actor}_{\text{All}}, \text{Object}_{\text{All}}, \text{Method}_{\text{All}})$$

However this utopian specification is not the Wikipedia we actually see today. From its inception, Wikipedia has been under attack by "vandals", trying to destroy its content integrity with graffiti, pornography, insults or deletions. In response, it has evolved many social rules, which currently involve literally hundreds of pages, detailing rights to edit, to delete, to resign, to join, to create new topics, to revert an item, to change signature etc., e.g. while anyone can edit any item, to create a new item one must first register. Also a social hierarchy has evolved, of stewards, bureaucrats, sysops and other levels including that of Jim Wales, who listens to others but as he notes "…at some ultimate fundamental level, this is how Wikipedia will be run, period". Even in Wikipedia, the STS owner has absolute rights.

We approach Wikipedia in two ways. First, as a successful online social system, to define generally how Wikipedia allocates various rights, so other applications can implement some or all of them in a different context. Second, a rights analysis may suggest alternative options to those chosen by Wikipedia.

Wikipedian rights. The Wikipedia model has several interesting rights features:

- *Public editing.* A wikipedia creation condition is that the item created is editable by all. When one publishes in a journal one gives them public display rights via a copyright form, i.e. all can view it. Wikipedia simply goes a step further, in that to publish in it, one must give public edit rights.
- *Accountability.* While in Wikipedia anyone can edit anything, it records the IP address, which IP can be banished for community offences.
- *Pseudonymity.* Registering an online pseudonym makes one real world anonymous but still accountable online, as one's pseudonym reputation affects promotions, and banishment loses reputation gains. All Wikipedia acts are traceable, so all an actor's acts can be reviewed. If someone vandalizes one item, their other item edits can be checked. Each actor's "talk page" allows public comments, which they cannot delete or edit.

- *Transparency.* Administrative processes, like steward promotions, are public, i.e. everyone has the right to comment, and everyone can see all comments on position applicants. Final decisions are based on democratic votes.
- *Versions.* After every edit a version copy is kept. Hence nothing on Wikipedia is really deleted, as a "revert" can undo an edit. This reduces rights errors, as no-one ever really loses their rights by permanent deletions.
- *Attribution.* Wikipedia records who made each contribution and so gives unique attribution rights if not unique edit rights.

Wikipedia is an encyclopedia by the people for the people. It engages the power of the community, but must still protect itself against unfair actions like vandalism. In Wikipedia, one "troll" can destroy the good work of many others. Part of that protection is its software base, which implements a rights specification that defines who can do what to what information. Misplaced computer power means a small minority can increasingly damage the majority, e.g. email spam [12].

Wikipedian alternatives. Two alternatives to the Wikipedia rights choices regard ownership of account name and ownership of new item contributions.

Account name. In Wikipedia one's account name is attached to every online edit. A Wikipedian who initially registers under their real name, like John Doe, then after some edits wishes to change to a pseudonym must ask an administrator to do this, as it affects the Wikipedia database. This creates usurpation problems as one can overwrite an inactive username, i.e. pretend to be a previous contributor. It also means Wikipedian actors have no right to resign, except as permitted. A rights analysis suggests one should own one's display name entirely. While Wikipedia can create and own unique accountable system ID for each actor, privacy gives actors the right to display themselves to others or not. This suggests two data entities, a "SystemID" known only to, and owned by, the system, and used for community sanctions like banishment, and an "ActorID" or signature, used for public displays. The SystemID never ever changes, so usurpation is impossible. The ActorID is entirely changeable, via an editable profile, so actors need no administrator to change it, and Wikipedias's change username policy is unnecessary. Wikipedians could genuinely resign, which is not currently allowed without administrative permission, and keep their signature, which no other could then use, or delete it and let another take that signature, making their edits attributed to "Resigned". The latter illustrates that while physical publishing attributions cannot be changed once done, online publishing authorship allows retrospective reattribution. Changing a Wikipedia signature from Rising Devil to Fallen Angel, gives the system two options. It can retrospectively change all your past edits to the Fallen Angel signature, or it can leave them as Rising Devil but allocate any new edits to Fallen Angel. In the latter case, your signature has effectively two versions arising from your edit of it.

Ownership choice. While Wikipedia favors public ownership it still supports copyright, perhaps because if Wikipedia expects members to follow its rules, it would be inconsistent for it to ignore national and international rules like copyright. Wikipedia Foundation could be held responsible the larger community for flouting copyright, as music copying web portals were shut down by legal action after copyright violations. While Wikipedia seems an island, it connects to a social mainland that values ownership. Hence rather than force people to give edit rights away, Wikipedia could give item creators choices like:

1. *Public Edit*: Anyone can edit the item to improve it (default).
2. *Public Comment/Private Edit*: The item is open to comment by anyone, but only you can edit it.
3. *Private*: The item is viewable by others but only you can edit or comment.

Option 1 is currently Wikipedia's only choice. Private rights (option 3) do not prevent a Wikipedia administrator from rejecting its display rights, i.e. "deleting" it. Giving authors choice could open Wikipedia to many experts currently wary of it. In Wikipedia's criticism section Legio XX notes: "Dr MC Bishop, an archeologist and world-renowned authority on Roman armor, wrote an article on the Roman lorica segmentata, only to see it mangled beyond recognition." and so refused to contribute. Andrew Orlikowski notes that well written articles are being "pecked" by amateurs until excessively long and frequently wrong. Wikipedia articles it seems can decay as well as grow. Choice lets a contributor topic expert give away some but not all control, and so not be overwhelmed by the majority, as the Wikipedia product is the last edit, and "revert wars" are won by the most persistent. Already within Wikipedia many suggest that "deserving" articles be "semi-protected", to limit allowed edits. "Private" contributions could be marked as such, to let readers evaluate credibility. Delegation gives even more complex options, as items could be open to the public for a while, then return to private editing. The general principle is that if Wikipedia wants to invite all authors, why not give authors freedom of choice, with public ownership as just one option?

Conclusions

Online societies like Wikipedia challenge humanity, asking what have we learned in several thousand years of society? If social knowledge can be put in information terms, computing could enable a global online society in the near future. If not, and if concepts like legitimacy have no computer meaning, then we must re-learn what social value means online. Wikipedia illustrates the struggle, as it began open and optimistic, then developed social structures and rules in response to vandalism. Its social rights model began simple but quickly became complex, and it could still fail as an online experiment by "social error". The difference between online and physical societies is that online "architecture" is defined by computer code, which in turn is defined by analysis and design. Rights analysis must become part of social-technical system design, to carry forward social knowledge into computer code, to close the "socio-technical gap" [1], and to help online communities succeed by increasing social health. Since the social level supersedes the technical one (Table 1), what a community says ought to happen actually should happen, not as an optional ethical "frill", but as a necessary requirement for social productivity. Wikipedia is new but its social problems are old, and communities over thousands of years have evolved social structures and rules as Wikipedia has done in just a few years. Social knowledge need not be relearned if we can define and discuss social rights in information terms.

We began with the premise that every information system object is owned, including the system itself. This seemed to make every online system a dictatorship, as indeed most bulletin boards are, albeit benevolent ones. However the social innovation of democracy suggests that in certain conditions, a group can "own itself" in general, i.e. the owner of a social-technical system can be its member community. Democracy, like privacy, can be seen as an extension of freedom, an individual's right to own him or herself. While the complexities of democratic voting cannot be discussed here, that online members of a system can own it brings our social logic full circle. Control "by the people" is fair, and "for the people" is socially beneficial, making democracy a legitimate solution to the social problem of who owns the community. The challenge is to translate successful social rights into software code. The rights framework outlined in this paper can help meet that challenge, and provide a basis to teach a rights information analysis in social-technical system design classes.

References

1. Ackerman, M. S. (2000). The Intellectual Challenge of CSCW: the Gap Between Social Requirements and Technical Feasibility. Human-Computer Interaction, 15(2), 179-203.
2. Alter, S. (2001). Which life cycle --- Work system, information system, or software? Communications of the AIS, 7(17), 1-52.
3. Geen, R. G., & Gange, J. J. (1983). Social facilitation: Drive theory and beyond. In H. H. Blumberg, A. P. Hare, V. Kent & M. Davis (Eds.), Small Groups and Social Interaction (Vol. 1, pp. 141-153).
4. Grudin, J. (1990). The Computer Reaches Out: The historical continuity of user interface design. Paper presented at the Proceedings of CHI '90, ACM SIGCHI Conference, Seattle, Wash., USA.
5. Kuutti, K. (1996). Activity Theory as a Potential Framework for Human Computer Interaction Research. In B. A. Nardi (Ed.), Context and Consciousness: Activity Theory and Human-Computer Interaction. Cambridge, Massachusetts: The MIT Press.
6. Lessig, L. (1999). Code and other laws of cyberspace. New York: Basic Books.
7. Locke, J. (1963). An essay concerning the true original extent and end of civil government: Second of 'Two Treatises on Government' (1690). In J. Somerville & R. E. Santoni (Eds.), Social and Political Philosophy (Vol. Chapter 5, section 27, pp. 169-204). New York: Anchor.
8. Rawls, J. (2001). Justice as Fairness. Cambridge, MA: Harvard University Press.
9. Shannon, C. E., & Weaver, W. (1949). The Mathematical Theory of Communication. Urbana: University of Illinois Press.
10. Whitworth, B., & de Moor, A. (2003). Legitimate by design: Towards trusted virtual community environments. Behaviour & Information Technology, 22 (1), 31-51.
11. Whitworth, B., 2006, Social-technical Systems, in Encyclopedia of Human Computer Interaction, Edited Claude Ghaoui, Idea Group Reference, Hershey. p533-541 http://brianwhitworth.com/hci-sts.pdf
12. Whitworth, B. and E. Whitworth, *Reducing spam by closing the social-technical gap.* Computer, 2004(October): p. 38-45.