Social Regulation in an Online Game: Uncovering the Problematics of Code

Mark S. Ackerman
School of Information and Department of Electrical Engineering and Computer Science
University of Michigan, Ann Arbor
ackerm@umich.edu

Jack Muramatsu
Vinyl Pulse
Los Angeles, CA
jmuramatsu@gmail.com

David W. McDonald
The Information School
University of Washington
dwmc@uw.edu

ABSTRACT

More and more interaction is becoming code-based. Indeed, in online worlds, it is all there is. If software is providing a new basis for social interaction, then changing the infrastructure of interaction may necessarily change social interaction in important ways. As such, it is critical to understand the implications of code – we want to know what the use of code means for socio-technical design.

In this paper, based on an ethnographic study of an online game, we examine social regulation in an online game world as a case study of socio-technical design using code. We wanted to know how changing interaction based in code conditioned use in our site. We found that code changed social regulation in three specific ways. First, code made some user actions that were socially unwanted to be immediately visible. Second, code could prevent some actions from occurring or punish users immediately. Finally, software was not able to see all action. Some user actions were too nuanced or subtle for code to catch; others were too ambiguous to place into code. Following Agre, we argue i that a "grammar of action" resulting from the use of code limits the kinds of behaviors that can be seen and dealt with.

These findings suggest that there is more than just a gap between the social world and technical capabilities. There are new possibilities, tradeoffs, and limitations that must be considered in socio-technical design, and all come simultaneously.

Categories and Subject Descriptors

D.2.1 [Software Engineering]: requirements/specifications, H.5.2 [User Interfaces] evaluation/methodology, user-centered design, H.5.3 [Group and Organization Interfaces] collaborative computing, computer-supported cooperative work, evaluation/ methodology, theory and models, K.4.2 [Computers and Society] social issues, K.4.m [Computers and Society] miscellaneous, K.8.0 [Personal Computing] games.

General Terms

Human Factors.

Keywords

Socio-technical design, games, social computing, online communities, virtual communities, social regulation, social control, police, social interaction, social life, software infrastructure, code, user study, ethnography

1. Introduction

Any socio-technical approach to design must consider the how the social and technical are intertwined. As Berg and colleagues wrote:

> ...understanding information systems requires a focus on the *interrelation* between technology and its social environment. 'Sociotechnical approaches' aim to do just this.... With this aim comes a concurrent ambition to improve these systems. ([6], pp. 297-298, emphasis in original)

Socio-technical design, then, implies that one cannot create systems that are fully specifiable and understandable in advance. However, any designer would wish for some ability to conform to the requirements of everyday work practice or life, some understanding of the adoption and adaptation likelihoods and issues, and the likely social and technical effects, before releasing the software and, indeed, before constructing the software. One must understand the implications for design decisions before any final commitments when constructing software, let alone putting it in place. One would like to understand as much as possible about the potential trajectories of a design in advance – that is, to understand code as a design medium.

How possible is this?

This paper is an examination of software as a design medium. Software is a relatively new material for design, and the effects are much less understood than those materials used in physical designs or construction. As Ackerman wrote earlier:

> ...human activity is highly flexible, nuanced, and contextualized and that computational entities such as information transfer, roles, and policies need to be similarly flexible, nuanced, and contextualized. ... [W]e do not know how to build systems that fully support the social world....

I argue here that it is not from lack of trying. Nor is it from lack of understanding by technical people. Numerous attempts have been made, not only within CSCW, but within many other subfields of computer science to bridge what will be called here the social-technical gap, the great divide between what we know we must support socially and what we can support technically. Technical systems are rigid and brittle - not only in any intelligent understanding, but also in their support of the social world. ([1], p. 179)

This necessarily requires considering how to design both technically and socially, in a socio-technical co-design space [3]. More importantly, one also needs to understand the tradeoffs in using code.

Indeed, Lessig [18] argues that software provides an architecture and an infrastructure for social control (in the sociological sense) and social regulation. If this is the case (and it is clear from numerous studies that it is), then by analogy one might expect that interacting through software structures would potentially change the nature of interaction. But Lessig largely argues at the societal level, as do most writers about social control, online privacy, cyber-crime, and the like.

Changes in interaction occur not only at the large, societal level, they also occur at a more micro-social level. Most design occurs at this level. Most products merely change the routines of normal users as they go about making their breakfast toast, paying their bills, driving to work, playing games at home after work, and so on.

Little is understood about how code affects interaction at the micro-scale (although see Latour's seminal study [17] in how to do so). Yet, software *is* providing a new basis for social interaction, especially at a micro-social scale. This parallels earlier examinations of how computer-mediated communication systems changed social interaction. Meyrowitz points out:

...The introduction and widespread use of a new medium of communication may restructure a broad range of situations and require new sets of social performances. ([21], p. 39)

In his book, Meyrowitz argued that new media have an extraordinary potential for creating new types of social spaces. However, Meyrowitz' point can be extended to software with the following conjecture:

Changing the infrastructure of interaction necessarily changes important aspects of social interaction.

Understanding any shifts in the infrastructure of interaction is likely to be the key to understanding the use of code and the design of software systems. If the infrastructure of interaction is now shifting, what might the resulting changes look like? And more importantly, can we say anything about the design implications for these changes in the co-design of a socio-technical space?

We believe that a socio-technical approach to design, and thereby socio-technical co-design, must be examined with field studies in order to fully understand the issues and tradeoffs at a micro-scale in using code. We employ a critical-realism stance [14, 16], examining the phenomena with all of its issues instead of engaging in utopian or dysutopian speculation. To do this, we examined game environments [23, 24], as all interaction is mediated by software. This work examined whether where code was better than humans at social regulation (or social control in the sociological meaning of that term). The goal was to understand the possibilities, limitations, and tradeoffs in using code: when to use code to prevent actions that were not considered appropriate (leaving aside for the moment who decides those acts are appropriate), when to use it assuming it was imperfect, and when to use it to augment a human's ability to police the world. This study offers potential design implications, and we go into those following a description of the study and the study site.

2. STUDY AND SITE

This paper is based on [23], which examined two text-based MUDs (Multi-player Dungeons). While the technology is outdated, the benefit for a study is that the social world could more easily be studied, partially because it is simpler, partially because it is smaller, but mostly because of the text-orientation itself. We discuss here only one of the worlds studied by the second author, a site we call Illusion here. Illusion was previously described in [24], primarily from the player perspective rather than the perspective in this paper.

2.1 Study

This study consisted of an approximately three-and-a-half year ethnographically-based study of Illusion by the second author. This participant-observation study examined the work performed by Illusion's system administrators (or Immortals, to be described more fully below) to regulate behavior within this game world. This data collection was preceded by another year or so of participant observation as a player on Illusion [24]. Additionally, the Illusion data is supplemented by a year long participant observation study of a secondary site, Odyssey.

The observational data was also supplemented with frequent open contextual interviews with both players and Immortals. Approximately 20 formal interviews and innumerable discussions were carried on with the Immortals; over 100 formal and informal interviews were done with players. At any given time, there were 6 to 8 Immortals (although these varied over time). At the time of the study, there were between 60 and 80 players online at any given time.

In addition, the second author closely examined Illusions' "penalty" messages that immortals wrote as a means of reporting each regulatory situation they personally handled. During the study period, there were over 400 separate incidents. The second author had Immortals reflect on a particular incident and comment on the process of handling it.

In addition, the second author had complete access to Illusion's code. A range of secondary material, including help files, web pages, background stories for the MUD and its groups, and bulletin board messages, were also used.

At a basic level, the data analysis was approached from the general approach advocated by Miles and Huberman [22] which relies on a set of initial research questions to frame the subsequent data analysis. The analysis was strongly influenced by symbolic/social interactionism ([8, 29, 31]) as its underlying theoretical, sociological perspective.

2.2 Illusion

Certain elements of Illusion must be described, in order for the discussion of its use of code to make sense, and so we turn next to its description.

Illusion is an elaborate multiplayer game. While there is chat and sociable activity on Illusion, much of the social activity centers on the game.

Players interact with the system and other participants through the use of personal characters they have created. A character, then, is a virtual fictional embodiment of a player. (We will generally ignore the distinction between players and characters, but a player may have many characters. This is important in some social regulation issues, as will be discussed below.) There are two basic types of characters on Illusion, Mortals and Immortals. Each type has a very different role in the game. The overwhelming majority of users have only Mortal characters. Mortals interact with one another as they play the game and explore the world of Illusion. A much smaller group serves as the game's administrators; that is, they own Immortal characters. They are responsible for overseeing all aspects of the MUD. Most, if not all, users who are Immortals also have Mortal characters. While an Immortal's primary responsibility is to oversee Illusion's day-to-day operation, many Immortals still enjoy playing the game with their Mortal characters as well.

2.2.1 Game Play -- "Leveling"

Illusion supports two very different forms of game play. While both forms of game play flow together within the MUD, players view them as distinct activities. The first form of game play is often known as "leveling", and involves improving one's character by killing computer-controlled opponents, mobs (short for "mobiles"), while exploring the virtual world.

Once characters reach a predefined amount of experience points, the system automatically promotes them to the next level. When a character "levels", key abilities and statistics that determine one's survivability increase. In general, higher-level characters are more powerful than lower level characters. Collaboration among players ("grouping") is key as players seek to advance their characters by leveling.

Nearly all players engage in this form of play. Some players may tire of constantly trying to level but at least initially most players strive to level and improve their character. For many players, leveling becomes the central focus of play and is the primary means of measuring their success within the game.

2.2.2 Game Play -- "Player Killing"

In addition, many players engage in a second form of play. Adversarial play includes player killing ("pkilling" or "PK") and stealing from another player's character ("p-stealing"). While Illusion imposes several restrictions on player killing, this activity is accepted and fairly commonplace.

Death on Illusion is not a permanent state of affairs. When characters die, they are reborn in the central town of Praxville. Characters do not lose any of their inherent abilities, but all the character's equipment (swords, armor, etc) remains on the corpse at the location of the demise. In order to retrieve the equipment, the character must retrieve the items from the corpse in a timely fashion, or risk their loss. This often involves a hectic effort to retrieve the equipment before other players or even the MUD's mobs take various items.

Many, if not most, of the players who engage in player killing find it to be a rewarding, pleasurable and fun experience. There are numerous, varied motivations that drive players to engage in adversarial play. Many players begin to engage in adversarial play after the experience of leveling their character(s) over and over again loses its appeal. The desire to experience novel and exciting forms of play motivates many players to explore the complexities of adversarial play. Players who engage in this style of play ("PKers") find competition against human opponents to be more rewarding than simply trying to kill computer-controlled creatures.

2.3 Immortals

Unlike Mortals, Immortals ("Imms") do not play the game per se; Immortals are responsible for the management of the MUD. They create, administer, maintain, and organize Illusion. Immortals have the ability to control almost every aspect of a player's game experience on Illusion.

This control is provided by specific Immortal-only MUD commands. Higher level Imms can modify the scores and attributes that dictate how a character interacts with the game and other players (experience points, hit points, etc.). They also have the ability to exact punitive forms of control such as removing a character's ability to converse over the public chat channels, banning connections from particular computers, and the like. Illusion's MUD code also provides Immortals with powerful control in terms of their awareness of Mortals and vice versa. They can hide their presence from Immortals ("Wizard Invisibility" or "Wizi"), or see everything a player types and sees without the player's knowledge or permission ("snoop").

Mortals are expected to comply with the judgments of the Immortal staff. Mortals have very little recourse in terms of appealing decisions and face punishment if they disobey Imms. This power is explicitly stated and reinforced in the system's formal rule set.

There is a loose Immortal hierarchy with specific job titles. The "Imps" or Implementers are responsible for the overall operation of the MUD. The Imps oversee and manage the activities of the other Immortals on their staff. On most MUDs, the Implementers are also responsible for maintaining the underlying code base. For a while, the second and third authors served as the MUD's two coders. Imps are responsible for maintaining the code, fixing bugs, and adding new features to the game. As well, Illusion has Imms who are responsible for overseeing "clans" (groups of players who have affiliated with one another), running "quests" (special mini-adventures designed to provide players with an occasional novel play experience), and handling player administration. There is also a "law" Immortal charged specifically with the enforcement of rules and to a lesser extent the creation and revision of Illusion's rule set. During the study period, there were eight active Immortals and three "semi-retired" Imms.

In practice, Imms do not confine their activities to their particular specialization. In fact, the Immortals have very versatile and to some extent interchangeable roles. For instance, it is not unusual for a player administrator to run a quest. In addition, all Imms are expected to enforce Illusion's rules. There is of course some limit to this interchangeability. Due largely to the lack of coding expertise among Illusion's other Immortals, only the two coder Imms work directly with Illusion's source code. In addition, decisions about the propriety of an Imm's actions are the responsibility of the Imps or head Immortals.

It must be noted that Illusion's Imms are all volunteers. This tends to limit the amount of time and dedication that Imms expect from one another. Coordination among Immortals is fairly limited. Efforts to improve coordination on Illusion such as the use of mailing lists and regular meetings have not been very successful. There are no pre-scheduled shifts for Immortals to be online. This can result in very sporadic Immortal presence online. At times there are many Imms online while at other times there may not be any Imm online.

3. Social Regulation and Police Work

As mention, of particular concern in this study are the social regulation actions of Imms. Social regulation (or social control in the micro-sociological sense of that term) concerns itself with getting people to behave in ways conformant with group or social norms and rules.[1] Deviant activity (again in the micro-sociological sense as activity that has been labeled problematic by parts of society) is to be limited, so that the actions of many people can proceed without undue disruption. Social regulation is of particular interest because it mimics certain "real life" work for police, watch people, system administrators, and even camp counselors, this allows the comparison between activities that are not code-based and activities that are.

We chose to compare Imms to police in order to compare code-based worlds to "real" worlds. We could have also examined barkeepers, club owners, and the like, as these people also engage in social regulation of their places. However, Imms have many of the formal and informal activities as police. As well, unlike occupations that engage in informal social regulation, there is also a large literature on police activities (e.g.,[7, 27, 28, 33, 34]).

In general, police work is characterized by very situated activity. Rubinstein, in his ethnography of police work, wrote about how police see a car, going on at some length about the nuanced observation and decision making they do:

> Many of the signs he is looking for are on the rear of a car, and this is where every policeman looks most closely. The first thing he checks is the license plate. Any unusual aspect immediately arouses his interest. A car with a paper tag or a temporary license is more likely to be stopped than one with a regular plate, because a person driving a stolen car frequently disposes of the original plate and replaces it with a paper tag.... A tag that is not properly attached, one fastened by wires or hanging on one bolt, is suspected because these are signs indicating that the plate may have been recently changed. Similarly, a dirty tag on a clean car or a clean one on a dirty car implies to the office that the two have only lately been mated and their marriage not celebrated legally. [27]

The stated purpose of the police is to enforce laws and regulations. Our emphasis here, however, is not in the specifics of code enforcement per se, but in the routine, everyday activities that constitute police work. Much of what police do is not, in fact, code enforcement. Instead, as Spradley [28] and Bittner [7] both point out, a considerable amount of police work is engaging in peace keeping and maintaining order, rather than in enforcing laws strictly or "by the book."

In both code enforcement and keeping order, a critical, but often overlooked, activity is that police must engage in quickly comprehending unknown situations. For example, a potentially dangerous traffic stop must be quickly assessed by a police officer. Stopping lower-class youths in a new luxury car may turn problematic, and quickly understanding a situation is critical to police work. Bittner [7] terms this routine, everyday activity "normalization," changing an unknown, chaotic situation into a known situation We will see below that these two types of work, maintaining order and normalization, become difficult in a virtual world like Illusion.

Much of what Imms do is similar to the police work. Imms have to determine what a situation is and whether their information is complete and correct. In the case of inter-player conflict, they may have to determine who is correct in their story, based on the players' stories, their own observations, and log data. One player or groups of players may feel that they have been the unwelcome victims of a rule violation committed by one or more other players. Not surprisingly, these players often lodge complaints with the immortals seeking some form of authoritative response to the alleged rule violation. While some complaints are handled almost immediately, most receive immortal attention after one or both of the parties involved has logged off. It is not unusual for several hours to pass before an immortal becomes aware of a complaint, typically through an asynchronous message post. Since direct observation is not possible, immortals must often take the written complaint and then proceed to try to make sense of the claim and attempt to verify whether events did indeed unfold as claimed. They must also decide whether a particular sequence of events warrants a punishment.

An example of one such situation follows. Nurb and Karn are Immortals, and they believe that Brosen may have broken a rule. In the following exchange, Karn decides to "snoop" (or invisibly watch) Brosen until he sees something suspicious.

```
Nurb: brosen doing something suspicious?
Karn: Yah.
Karn: He's trying to weazle out of an
indirect PK [player killing] he did this
morning.
Nurb: sigh
Karn: And without the victims here, I'm
having a hard time pining him
down on it.
Karn: Possibly.
Nurb: he's back
Karn: I'm going to snoop [watch] him
until he slips something.
```

Unlike "real life" police, Immortals do not have to make these decisions in real time generally – no one has a gun in cyberspace.

[1] Occasionally there have been objections to this work that 'social control' is not desired in online worlds. This is a confusion of prescriptive and descriptive stances. Social control verging on 1984 cuts against the standard libertarian stance of computer scientists and decency in general. However, some level of social regulation (or social control in the sociological sense of the term) is required for any collectivity engaging in social activity. We all accept the necessity of handling, in some way, people who are attempting to harm others or who keep others from engaging in their own acceptable behavior. This is what we mean by social regulation and social control, and nothing more.

In addition, the Immortals have additional information at their disposal. While they lack the face-to-face cues in conversation, they have log files of people's activities.

4. Code-based social regulation

4.1 Code and social regulation

In our study site, three broad categories of social regulation based on code exist. Some of the immortals' actions are similar to face-to-face interaction, such as community outreach, motivation, and so on, but our interest here is on those aspects of social regulation where code makes a significant difference.

The three broad categories were:

1. *Code that allows Imms to observe player actions.* This allows Imms to investigate player complaints or other problematic interactions on an as-needed basis. Imms may also be able to replay play logs to examine player behavior after some period of time.

   An example of Imms' needing to observe players include situations where Imms try to detect and determine multiplay. Multiplay is when one player uses two characters simultaneously or in combination to gain an unfair advantage. As mentioned, players are allowed to create, own, and use as many characters as they would like. This flexibility in terms of character ownership policies allows a player to explore different aspects of the game by playing the game with various character classes and races. Illusion places a broad restriction on the use of multiple characters. A player must use his characters separately and as independent rather than cooperative entities. The term "multiplay" refers to any and all violations of this restriction.

   One form of multiplay, difficult to automatically detect, is transferring equipment (swords, armor, etc) from one of a player's characters to another of the player's characters. This is sometimes done by dropping equipment in a specific location, logging off, logging back on under a different character, and picking up the equipment in the same location. The most egregious place is the donation box in Praxville, but other more subtle forms of this multiplay are harder to automatically detect. Imms must also watch for similar transfers by using an intermediary character owned by a different player.

   Another example is Imms' detecting players who are exploiting a "crash bug" – a bug that is known to crash the system. It is against the rules to knowingly exploit a bug in the MUD system software that defines and manages Illusion. The idea behind this rule is to prevent someone from leveraging an unintended loophole in the game code so that they can give their character(s) an unfair advantage. One specific example of this is that Illusion has a bank where players can deposit and withdraw gold and silver they have accumulated during their adventures. At one point, the system used to crash whenever a Mortal ordered a charmed mob ("pet") to carry out a bank transaction. Crash bugs are not only an annoyance to other players but can also be exploited to duplicate ("dupe") valuable equipment. A player can dupe a piece of equipment by saving his character, then giving the equipment to another player, having the second player save (including the borrowed equipment), and then crashing Illusion. When the MUD reboots, both players will each have a copy of the piece of equipment. Players are expected to report any bugs they discover instead of exploiting them for their advantage.

   Immortals on Illusion can employ a real-time configurable event notification system known as WizNet. For example, WizNet can provide a notification whenever a player logs a character on or off the system. In addition, an Immortal can receive notifications whenever a player's character dies. There are numerous other such events that WizNet can monitor which would otherwise not be easily detectable. WizNet is a standard feature of the ROM code base upon which Illusion is based. In addition, two network commands provide information about which players are currently logged on to the system, providing more information useful for detecting multiplay and determining alternative characters.

   The "snoop" command, a directed form of surveillance, can also be used by Imms. Not only does snoop show the Immortal where the player's character is and what they are doing, but the Immortal can also see any conversation in which the player is involved. It is as if the Immortal is standing behind the player looking over his shoulder. Snoop does not provide any notification to the player that she is being scrutinized in this manner. The use of snoop usually follows some prior indication that a character may be up to no good or may be triggered by an Immortal's concern about a particular character. While it is technically possible to snoop multiple players simultaneously, the pace of the information may make it hard to comprehend and follow.

   We will return below to the issues in observing player actions to detect and manage problematic behavior.

2. *Code that automatically prevents certain types of interaction.* Preventative social regulation keeps players from taking specific actions detailed, in this case, in the software code [19]. The advantage is that problematic activity is automatically eliminated. This requires detection of people trying to engage in this behavior, and this detection is not a trivial accomplishment, as will be discussed below.

   How is this different from code that allows Immortals to observe players? Basically, WizNet and other sources of activity data require detection of problematic behavior, interpretation of the log files and the inferred behavior, potentially the gathering of additional information, and a response by the Imm. Software, in the form of coded regulation, continuously monitors players' activities and responds automatically. Code that prevents interaction or, as we will discuss below, code that takes punitive actions against players who engaged in specified interactions, *automatically* engages in social regulation. The types of interactions that can be automatically detected are more limited than those that can be inferred from observation, and this will also be discussed below.

   A simple example of preventative code is that characters under level 15 are prevented from posting public messages to all players. This prevents people from creating a throwaway character to post a note expressing a controversial or insulting view and then deleting oneself to avoid consequences.

Another clear-cut, although more complex, example of preventive code on Illusion is the code that controls “grouping”. As mentioned, grouping is when Mortals combine their efforts in a “group” in order to improve their chances of leveling. The code prevents characters with strikingly different attributes, for example the alignments “angelic” and “satanic” from grouping with one another in order to force role-playing on Illusion. Players attempting to form groups with characters of diametrically opposed alignments will receive a message indicating that the alignments of the characters are incompatible. Accordingly, the system will refuse to form the requested group.

Finally, code was put in place to facilitate the separation of players who want to engage in player-killing (“PKers”) and those who do not. Preventative code was created, for example, to change magic spells; otherwise, PKers when casting a deadly spell could adversely and unwittingly affect non-PK players. This prevents accidental but problematic behavior, and thereby reduces Imms’ workload.

3. *Code that allows specific types of interaction but automatically assigns a penalty.* This is considered better for gameplay, and signals the general libertarian assumptions in the code-base. (See [15] and [2] for other examples of this ideological bias enscribed in systems.) Players can take the action they desire, but they will be punished for doing so. Thus, it is up to the player to decide whether to take a penalty, sometimes severe, or to forego the action.

   One of the simplest examples is how the game code deals with players who kill another player in the central town of Praxville. If a Mortal kills another character in town or steals from another player, the Praxville guard mobs will attempt to jail that character on sight. The player can do the action, but then is punished.

   Another example is the code that involves “out-of-level player-killing,” or OOL PK. Higher level characters are generally considered to be more powerful and formidable in combat than lower level ones. Therefore, it is relatively unfair for a character to attack and kill a character that is of considerably lower level. Such an act is considered OOL PK. (Note this is slightly different than pkilling newbies. Both characters in this situation may be reasonably high level.) The MUD currently attempts to detect such instances and punishes the offense by deducting a certain amount of experience points from the offending character. The penalty is proportional to the level difference between the two characters. It is possible and quite common for player characters to lose a sufficiently large amount of experience points that the MUD automatically lowers their level; that is, the players are demoted to a lower level. It is also not uncommon to see Mortals take this penalty in order to PK someone who has offended them.

As well, there was the potential for a fourth category, with code that allows interactions but flags them for later inspection. This is a mixture of the first and third category. We will not deal with this separately, as it was not used extensively in Illusion.

5. Changes from using code

Code as the basis for interaction changes everything. One can observe every person in this world simultaneously, all of the time. Surveillance is ubiquitous - through walls, in places we would consider private in "real life."

5.1 First Change: Visibility and Coverage

There is no doubt by the police about what is in the trunk of the car once they open it: it can be directly observed. However, the police cannot see into all car trunks. In Illusion, there would be no way to steal without being noticed; it can be directly (and post hoc) observed. This is the first change that providing interaction through a software infrastructure plays in design: General awareness and subsequent detection of potentially problematic activities within a setting are constrained by their visibility, the extent to which information about their occurrence and other related contextual details is available for observation. One of the key aspects of visibility is coverage, that is how much of the observable world can be viewed or monitored at the same time.

In many physical settings, an individual can only observe nearby. Thus police officers in the physical world rely on a moving patrol within small well-defined areas. The coverage generated by this approach is limited primarily by the number of available officers. Because placing an officer at every conceivable location is both impractical and untenable, coverage is necessarily limited. As a consequence, it is quite likely that a portion of activities is not directly observed by the police.

In contrast, immortals make use of both real-time event notification systems and targeted persistent logging systems to maintain awareness of specific predefined events across the entire mud simultaneously. These systems provide pervasive coverage for supported activities; regardless of where an activity takes place in the world, a notification will be generated. Furthermore, once the events are defined, the information is made available without any additional manual effort by the immortals. Because this form of monitoring is not based on physical observation, there is no need to move around the world or engage in a patrol in order to acquire the information provided through the notifications. These systems facilitate a level of coverage with minimal effort that is not generally achievable in physical settings.

In a sense, Imms can engage in surveillance through walls and inside structures. The logging has no bounds, although there are some practical constraints (e.g., disk space).

5.2 Second Change: Prevention of Actions

The second change is some actions and social interactions can be prevented in advance through the software. The desired activity cannot be carried through – there is no way to do it through the system or the system actively disallows it. An example mentioned above is that Illusion prevents certain types of players from grouping with others. As well, certain kinds of theft are just not possible. In "real life", it would be as though it were possible to arbitrarily create new physics laws that changed physical interactions. Still other rules automatically punish players when they act problematically. As mentioned above, it is possible, for example, to a player to kill other players "out of level" (OOL).

Again the level of automaticity is not present in “real life.” Immortals were observed to consider delegating rules with a high frequency of violations to coded rules. The basic rationale for this is that coded rules would be better equipped to handle a large set of frequent violations as they provide continual rather than spotty detection and enforcement. Additionally, immortals are concerned about the amount of effort and time spent on regulation as it limited the amount of time they could spend on potentially more interesting and rewarding duties such as area building and storyline writing. The subsequent desire to reduce the effort in regulating behavior serves as a motivation to delegate rules to code.

5.3 Third Change: Routiniziation of Action

The final change that code as interaction infrastructure introduces is more subtle. Code is, in its essence, bureaucratic, in the Weberian sense of the term [11] – predictable, standardized, rationalized decisions made according to published regulation. Code is always fair and impartial, returning the same results for all players, given specific circumstances. Context is largely ignored, as was part of the original intension of bureaucracy. Any exceptions or any changes in circumstances (that is, bringing criteria in from the situational context to part of the decision) must be separately coded. In a game, interestingly, this often appreciated and indeed expected by players (perhaps because it is part of their social conception of fairness), but this is clearly not the case with police or other law enforcement.

We will return below to a further examination of this third issue and why it is critical. It deeply influences the first two, and it is the key to understanding the tradeoffs when using code in a socio-technical design.

6. Code-based regulation and grammars of action

As one might expect, all is not straightforward. All of these new or changed possibilities for social regulation based on code suffer from the common problem of brittleness – or the lack of contextualization and nuance – so familiar to socio-technical researchers.

Monitoring and detection can be done only through computational mechanisms, and these mechanisms lack the nuance and flexibility of "real life" social interaction. Social inquiry has established that the details of interaction matter to people in their interactions [10, 31], and that people use this detail with considerable agility to frame their interactions and actions in social settings [10, 13, 32]. Code, of course, cannot support this completely, creating a tension between what needs to be done socially and what can be accomplished technically [1].

Below we examine where this tension becomes a problem in Illusion, illustrated by specific examples of this tension. Following, we will also examine the generalizability of these same problems to socio-technical design overall.

In this discussion, Agre's "grammar of action" provides a useful tool for examining these problems [4, 5]. As part of what Agre calls the "capture model", an analyst decomposes a subset of human activities into "a repertoire of atomic elements. ([5], p. 187)" as one is trained to do in systems analysis. The analyst then goes on to "devise a grammar that can generate, and thus represent, all of the institutionally permitted sequences of action. (p. 187)" Thus, a grammar of action is the representation enscribed in the system that describes all of the activities allowed. As with any grammar, the expressions and statements that can be formed must cover the universe one wants to design and control. Although any language is inherently limited in practice (as Garfinkel so aptly points out), a grammar of action, as a formal representation of human action, is necessarily limiting, because it is designed to be so: It creates a reduced universe of activities that are allowed by an organization, institution, or other social world.

In Illusion, the grammar of action is limited in important ways with several critical implications. First, the objects of inquiry in the grammar can include actors, entities being acted upon, entities in the situation (i.e., the important context), and some events. These must be created by known computational mechanisms –by simple variables, patterns of variables, or classification mechanisms (such as algorithms or heuristics). This creates a simplified and discretized design space.

Sometimes this is appropriate. For example, in the earlier case of players under level 15 not being allowed to post messages, the rule easily translates into the actors and objects of Illusion's grammar of action. (One may note, even in this case, Agre's acute observation that the decomposition by a systems analyst into a grammar of action then becomes the only acceptable manner of describing activity and thereby controls and defines the activity itself: We do not normally describe people by specified, discrete levels.)

Other times, the limitation becomes problematic. Sometimes this is the result of not coding particular patterns. Surveillance at a distance (as opposed to direct observation) is possible only if previously coded:

> However, the in-game notification systems provide selective visibility in that they rely on the creation and existence of predefined events – such as logins and logoffs. Activities that are not captured by an existing event are not reported by the system and are not made available to immortals. [23]

Other times, the problem results from an imprecise matching between the representation in Illusion's grammar of action (or any grammar of action) and the activity itself. In Illusion, the Imms decided to forbid posts that contained profanity. One could imagine that a set of terms, with constant updating, could serve to determine profanity, but some terms are ambiguous and the matching would be imprecise. In this case, there is not a good match between the actual activity and the objects in the grammar of action, and the resulting discrepancy requires human monitoring.

More importantly, any grammar requires composition of elements into expressions and statements. In this case, the grammar of action allows the composition of only some activities in Illusion. That is, only some sequences of actions can be represented in Illusion's code. Those that cannot be composed properly – because there is no effective match with the objects of inquiry, because some objects are left out, because some sequences are not specifiable in advance, or because the specification does not match human activity well – cannot be monitored or controlled.

As an example, the Imms forbade communications that were harassing to others. Players can post any text. They can also message one another through Illusion's "say" command. The representations in Illusion's grammar of action, however, cannot distinguish harassing from non-harassing text; they are all strings of text. Indeed, the same string of text can be harassing or not, depending on the game character involved, the assumed player, the history of the conversation, and the history of previous interactions. If this is the case, however, utterances, which are an enormous part of this social world, are not handled well by software control.

In addition to compositional problems, the complexity of the design space is also a problem. Indirect PK is player-killing through indirect means. Normally, player killing is direct – one player attacks another in some form of combat. Indirect PK is generally defined as what is not direct PK. That is, players may place spells in this game world on others, forcing another player to starve to death. Or, players may tele-transport others into the middle of an ocean, effectively drowning them. It turns out there are many, many ways of doing indirect PK, so many that the Immortals have not seen fit to enumerate them. Instead, the Imms weigh whether the results of a PK episode were indirect and unfair.

This is not an esoteric problem. Earlier, we suggested that out-of-level (OOL) PK was relatively straightforward. There is an algorithm that determines the difference in levels between players, and above some threshold, it penalizes the higher-level player if he kills the lower-level. However, indirect PK complicates this enormously. OOL PK was abused by clever players so as to essentially force another player into this penalty. For example, a third player might charm a lower level character into attacking the higher-level player, or conversely, a third player might charm the higher-level player into tele-transporting a low-level player into an ocean. In sum, the addition of two complex problems creates an interaction among the software features that becomes difficult, if not impossible, to adequately encode.

Indirect PK highlights another problem in Illusion. While social activities are not always well represented, neither is punishment or social regulation. Indirect PK is too complex and nuanced a problem; attempts to regulate it require human intervention. Intent of the player becomes critical, and intent is often determined by understanding the past history of that player. For Illusion's Imms there are players who are "problem players," who cause trouble, just as there are for police [9]. There are also players who are not; those play the game without trouble and help other players. Problem players, who may, for example, have a history of indirect PK, are punished more severely; the Imms believe that the trajectory of a player and his trustworthiness are critical to assessing the need and type of social regulation invoked. This cannot be done adequately with code, as it requires an understanding and categorization of a player's likely social trajectory.

What are the practical consequences of these limitations to the use of code? First, while it would appear that code makes social regulation easy, in many respects, the Imms can be unusually blind as well. In effect, these systems provide almost limitless visibility for a subset of all potentially observable activities occurring within the world and no direct visibility over the remainder [23]. Furthermore, since only some things can be effectively coded, this problem is not one that will go away.

Second, automaticity is strongly desired for efficiency: it should make the Immortals life easier and fit the idealized views of "fairness" by both Imms and players. However, automaticity is possible only if the conditions can be made explicit (e.g., in player killing) or the antecedents can be made explicit (e.g., kind of player). Both depend on what can be detailed in the representation, and we have seen the grammar of action is inadequate for important social interactions and activities. Again, this problem is inherent in the brittleness of software (rather than software's brittleness being the result of carelessness or even the result of inadequate grammars of action). We believe it is an inherent limitation based in the current nature of software mechanisms [4], [1].

Finally, the side effects and alternative ways of doing any given problematic behavior are often not specifiable or well understood in advance. We have already seen this is a result of the poor match between social activity the grammar of action's representational capabilities and of the inherent complexity of the socio-technical design space. Problem players will continue finding ways to be problems – whether it is indirect PK, harassing other players, or new forms of cheating. This complexity combined with the continued existence of socially problematic people leads directly to inconsistent software, side effects in the software, and bugs.

What are the implications for socio-technically oriented design? As with any design material, it is important to consider the limitations of that material as well as its capabilities. We have argued here, based on our field site's data, that the grammar of action possible in a code base *fundamentally* is limited. This is true for both the grammar of action's composition and in its requirement for explicitedness.

7. Discussion

There are a number of counter-arguments that could be made to our analysis. First, one could argue the Illusion site is an amateur world, and professionally run worlds would have better code infrastructures or a better understanding of the social requirements. While it is true that Illusion was an amateur world, the underlying code base was shared among many MUDs. Regardless, if this argument were true, it would be true only at the margin. The problems, we believe, result from the fundamental nature of code, as it currently exists. (While one might imagine a type of code or computational mechanism that had greater fidelity to social experience, our argument is based on what currently exists.) It is difficult to imagine a professional world with a substantially greater fidelity to what we know we have to support socially [1]. Based on what exists now, only machine learning or data mining might produce better results and only if the equivalent of Imms looked for the appropriate cues. Again, the issue becomes one of whether the right behaviors are visible and searchable, how any equivalent of Imms can observe players' actions, and whether the grammar of action is a suitable, full representation. We find the a solution to the last issue to be unlikely. Whether the difference at the margin between amateur and professional worlds is important, however, remains to be investigated.

Second, there is another argument that social computing will get better at this sort of thing over time, and the problem will go away. For example, one might expect rules that better mimic what players want to do. One form of this argument is that better mechanisms or a greater number of them will be possible in time, and this reduces to the previous counter-argument. An alternative form, however, is more interesting: We will gain better and better approximations to "natural" behavior over time. It would well be (and probably will be) an interesting blend of co-evolutionary social construction where users come to expect specific types of standardized behavior, the system comes to support (and regulate) those standardized behaviors better and better, and so on in a continuous turn. This is a question that must be examined empirically over time.

In any case, one might expect there to always be a gap between better notification systems that better detect problematic actions and users' capabilities and actions. Representations are inherently limited, and users do not follow prescriptive rules blindly (i.e., they are not "cultural dopes," as argued in [10, 13, 32]).

There are two clear limitations to this study that must be acknowledged. We cannot evaluate, using ethnographic data, what might have happened without the intervention of the software infrastructure. That is, we do not know what Illusion's users would have done in the absence of software. As well, as with any ethnographic study, the findings could be considered inherently ungeneralizable. However, we believe that we have theoretically abstracted suitably from this specific online community to indicate an entire range of issues with software as an infrastructure for interaction. We believe, as in other interpretivist work, we have achieved an adequate theoretical generalizability [30]. As such, the findings are generalizable, although they must be tested in other settings. Indeed, we believe, but do not know, that they generalize to many common artifacts, such as the possibilities for social regulation of cars in traffic.

8. Conclusions

We began with a claim that software is providing a new basis for social interaction, and drew a parallel to Meyrowitz's earlier observation:

> ...The introduction and widespread use of a new medium of communication may restructure a broad range of situations and require new sets of social performances. ([21], p. 39)

We generalized this observation to be that *changing the infrastructure of interaction necessarily changes the social interaction in important ways.*

More and more interaction is becoming code-based. Indeed, in online worlds, it is all there is. As such, it is critical to understand the use implications of that infrastructure – we want to know what it means for socio-technical design. As Postman argued:

> Technology is not an unmixed blessing.…we also need for them to talk about what new technologies can undo. … I just don't think we can go into these things anymore with our eyes closed. (as cited in [20], p. 84)

In this paper, based on an ethnographic study of an online game world, we have shown that changing the infrastructure of interaction to be based in code conditioned the interaction in our site. We examined social regulation in his online game, and found:

- The use of code changed social regulation. We are necessarily limited to an indirect comparison of social regulation in Illusion with police work in real life. However, we believe that both the ways of regulating action *and* user actions in the game were formed by and contingent on the software infrastructure. This study showed that software as a medium has specific properties, at least for social regulation, and we showed many cases where those properties played out.

We also found that code changed social regulation in three specific ways:

- Code made some user actions that were socially unwanted to be immediately visible and punishable. This is essentially seeing through walls by those controlling the online community.
- Code could prevent actions from occurring or punish the appropriate parties automatically.
- Software, however, was not able to see all actions. Some were too nuanced or subtle for code to catch; others were too ambiguous to place into code. Indeed, we argued that the "grammar of action" in an online world must limit the kinds of actions that can be seen and dealt with.

These three findings are actually the flipside of the same coin – they are caused by the visibility of actions in a software-based world and the nature of grammars of action.

What we have seen in Illusion is that code – even professionally-written, well-maintained code – is likely to have severe limitations when considering micro-scale social interaction, and the trade-offs must be deeply considered in design. We have also seen the utility of field-based research and a stance based in critical realism [14, 16] in examining the problems of code. The popular or trade press' utopian or dysutopian visions of the use of code are limited. We believe studies that are empirically-grounded and that examine people actually using, constructing, and designing code will continue to show the profound trade-offs in using code.

These findings suggest that there is more than just a gap between the social world and technical capabilities [1]. There are new possibilities, tradeoffs, and limitations that must be considered in socio-technical design. This argues for important future work that directly follows from this study to consider the implications of using code as an interactional infrastructure over time. Do these tradeoffs and limitations become routine? How does that routinization affect interaction in a continuous cycle of structuralization and enactment [12, 25, 26]? Do rules become more normative over time, requiring users to be self-regulated and further oriented towards even more rules? Or will users find ways to make these mechanisms of social regulation and even rules themselves into resources that they can appropriate to their own purposes (as was suggested in this study). What are the specifics of enscription and encoding whereby social regulation or preferred social behavior gets placed into software? As interest in socio-technical research and design grows, we believe that these questions and other similar ones would be profitable as the next line of investigation.

9. Acknowledgements

This paper benefited from discussions with Paul Dourish, Gloria Mark, Wayne Lutters, Madhu Reddy, Kjeld Schmidt, Phil Agre, Jonathan Grudin, Howard Rosenblum, Steve Sawyer, Chris Schmandt, Christine Halverson, Steve Jackson, and Ayse Buyuktur. None are responsible for any of the misunderstandings here. The work was also deeply influenced by the work of Rob Kling, John King, Wanda Orlikowski, and Leigh Star.

10. References

[1] Ackerman, M. S. The Intellectual Challenge of CSCW: The Gap Between Social Requirements and Technical Feasibility. *Human-Computer Interaction*, 15, 2-3 2000), 179-204.

[2] Ackerman, M. S. *The Politics of Design: Next Generation Computational Environments*. Information Today, City, 2008.

[3] Ackerman, M. S., Darrell, T. and Weitzner, D. J. Privacy in Context. *Human-Computer Interaction*, 16, 2-4 2002), 167-176.

[4] Agre, P. *Computation and Human Experience*. Cambridge University, New York, 1997.

[5] Agre, P. Changing Places: Contexts of Awareness in Computing. *Human-Computer Interaction (HCI) Journal*, 162001), 177-192.

[6] Berg, M., Aarts, J. and van der Lei, J. ICT in Health Care: Sociotechnical Approaches. *Methods of Information in Medicine*, 422003), 297-301.

[7] Bittner, E. *Aspects of Police Work*. Northeastern University Press, Boston, 1990.

[8] Blumer, H. *Symbolic Interactionism: Perspective and Method*. University of California Press, Berkeley, 1986.

[9] Cicourel, A. *The Social Organization of Juvenile Justice*. Transaction Publishers, New York, 1995.

[10] Garfinkel, H. *Studies in Ethnomethodology*. Prentice-Hall, Englewood Cliffs, NJ, 1967.

[11] Gerth, H. H. and Mills, C. W. *From Max Weber: Essays in Sociology*. Oxford University Press, City, 1968.

[12] Giddens, A. *The Constitution of Society: Outline of the Theory of Structuration*. University of California Press, Berkeley, CA, 1986.

[13] Heritage, J. *Garfinkel and Ethnomethodology*. Polity, Cambridge, 1984.

[14] Kling, R. Computerization and Social Transformations. *Science, Technology and Human Values*, 16, 3 1991), 342-367.

[15] Kling, R. *Value Conflicts in the Design and Organization of EFT Systems*. Academic Press, City, 1991.

[16] Kling, R. *Computerization and Controversy*. Academic Press, City, 1996.

[17] Latour, B. Mixing humans and nonhumans together: The sociology of a door-closer. *Social Problems*, 35, 3 1988), 298-310.

[18] Lessig, L. *Code: And Other Laws of Cyberspace, Version 2.0*. Basic, New York, 2006.

[19] Marx, G. T. *The Engineering of Social Control: The Search for the Silver Bullet*. Stanford University Press, City, 1995.

[20] McCreary, L. Postman's Progress. *CIO*, 7, 3 1993), 74-84.

[21] Meyrowitz, J. *No Sense of Place: The Impact of Electronic Media on Social Behavior*. Oxford University Press, New York, 1985.

[22] Miles, M. B. and Huberman, A. M. *Qualitative Data Analysis*. Sage, Thousand Oaks, CA, 1994.

[23] Muramatsu, J. *Social Regulation of Online Multiplayer Games*. PhD thesis, University of California - Irvine, 2004.

[24] Muramatsu, J. and Ackerman, M. S. Computing, Social Activity, and Entertainment: A Field Study of a Game MUD. *Computer Supported Cooperative Work: The Journal of Collaborative Computing*, 7, 1 1998), 87-122.

[25] Orlikowski, W. Using Technology and Constituting Structures: A Practice Lens for Studying Technology in Organizations. *Organization Science*, 11, 4 2000), 404-428.

[26] Orlikowski, W. J. The Duality of Technology: Rethinking the Concept of Technology in Organizations. *Organization Science*, 3, 3 1992), 398-427.

[27] Rubinstein, J. *City Police*. Farrar, Strauss and Giroux, New York, 1973.

[28] Spradley, J. P. *You Owe Yourself a Drunk: An Ethnography of Urban Nomads*. Little, Brown, and Company, Boston, 1970.

[29] Strauss, A. *Creating Sociological Awareness: Collective Images and Symbolic Representations*. Transaction, New Brunswick, 1991.

[30] Strauss, A. L. *Qualitative Analysis for Social Scientists*. Cambridge University Press, New York, 1987.

[31] Strauss, A. L. *Continual Permutations of Action*. Aldine de Gruyter, New York, 1993.

[32] Suchman, L. A. *Plans and Situated Actions: The Problem of Human-Computer Communication*. Cambridge University Press, New York, 1987.

[33] van Maanen, J. *The Asshole*. Goodyear Publishing Company, City, 1978.

[34] van Maanen, J. *Tales of the Field: On Writing Ethnography*. University of Chicago Press, Chicago, 1988.