

## Perspectives on the Central Role of Open Source Code for Participatory Governance: Human Laws / Coded Conditionals

The social, legal, political and economic ramifications of using Open Source Code as the basis for the Participatory Governance Environment are profound. At core – “self governance” is about a community taking responsibility for ownership and control of that community. In the foundational book *City of Bits*, William Mitchell, Dean of the MIT School of Architecture and Planning (and Patron Saint of the MIT eCommerce Architecture Program) was the first scholar to reveal and describe the interplay between computer code and the laws by which we govern ourselves. In the aptly named section “[Human Laws / Coded Conditionals](#)” Dean Mitchell sets out the challenge in four seminal paragraphs:

Out there on the electronic frontier, code is the law. The rules governing any computer-constructed microworld-of a video game, your personal computer desktop, a word processor window, an automated teller machine, or a chat room on the network-are precisely and rigorously defined in the text of the program that constructs it on your screen. Just as [Aristotle](#), in *Politics*, contemplated alternative constitutions for city-states (those proposed by the theorists [Plato](#), Phaleas, and Hippodamos, and the actual Lacedaemonian, Cretan, and Carthaginian ones), so denizens of the digital world should pay the closest of critical attention to programmed polity. Is it just and humane? Does it protect our privacy, our property, and our freedoms? Does it constrain us unnecessarily or does it allow us to act as we may wish?

At a technical level, it's all a matter of the software's conditionals -those coded rules that specify *if* some condition holds, *then* some action follows. Consider, for example, the familiar ritual of withdrawing some cash from an ATM. The software running the machine has some gatekeeper conditionals; *if* you have an account and *if* you enter the correct PIN number (the one that matches up, in a database somewhere, with the information magnetically encoded on your ATM card), *then* you can enter the virtual bank. (Otherwise you are stopped at the door. You may have your card confiscated as well.) Next the program presents you with a menu of possible actions -just as a more traditional bank building might present you with an array of appropriately labeled teller windows or (on a larger scale) a directory pointing you to different rooms: *if* you indicate that you want to make a withdrawal, *then* it asks you to specify the amount; *if* you want to check your balance, *then* it prints out a slip with the amount; *if* you want to make a deposit, *then* yet another sequence of actions is initiated. Finally, the program applies a banker's rule; *if* the balance of your account is sufficient (determined by checking a database), *then* it physically dispenses the cash and appropriately debits the account.

To enter the space constructed by the ATM system's software you have to submit to a potentially humiliating public examination-worse than being given the once-over by some snotty and immovable receptionist. You are either embraced by the system (if you have the right credentials) or excluded and marginalized by it right there in the street. You cannot argue with it. You cannot ask it to exercise discretion. You cannot plead with it, cajole it, or bribe it. The field of possible interactions is totally delimited by the formally stated rules.

So control of code is power. For citizens of cyberspace, computer code-arcane text in highly formalized language, typically accessible to only a few privileged high-priests-is the medium in which intentions are enacted and designs are realized, and it is becoming a crucial focus of political contest. Who shall write the software that increasingly structures our daily lives? What shall that software allow and proscribe? Who shall be privileged by it and who

marginalized? How shall the writers of the rules be answerable?

The theory underlying the work of this semester's seminar is that code used for self-governance should – perhaps “must” – be owned by those who use it. Unlike a social chat room or hobby club, the code being designed by through our graduate seminar is intended to be used for serious, formal and legally binding decisions. For example, a municipality might use it to set ordinances and bylaws; a Union might use it to debate and vote upon a multi-year contract; the Share Holders of a corporation could use it to make decisions governing the corporation they own; or members of a political party could use it to set their platform and hold conventions at which their candidates are chosen to hold the party banner in elections. These business contexts – whether in the public or private sectors – involve substantial and solemn transactions. Legal rights and duties of users of the system can be significantly affected by the outcomes. This “problem statement” was deliberately chosen precisely because it is challenging. It is hoped that modeling these difficult and formal processes will result in the development of code that is broadly useful to a variety of self-governing communities.

What the Internet has made possible – tens of thousands of people working on the same topic over a simultaneous and collaborative system, pushing decision making power down to many individuals working over a “many to many” network – was not practical outside of the laboratory until recently. Even a couple of years ago, it would have been unthinkable audacious to seriously model these types of transactions. The technology was not yet mature enough; the models did not exist; institutional openness to embracing these core technologies was not yet ripe; market adoption and facility with networked software – while noteworthy – was not yet sufficient to make such a system practical for most communities. Today, the eCommerce Architecture Program has met with much success in finding ready partners in the public and private sectors for user-testing of our system, requirements gathering and eventually for early deployments.

Clearly, the answer to Dean Mitchell's rhetorical question “Who shall write the software that increasingly structures our daily lives . . . how shall the writers of the rules be answerable”, will differ depending upon the context in which it is applied. In the case of ATM software, the answer would seem to be that the Bank owns and controls the code. Yet, even in this context there are a myriad of oversight and regulatory authorities with the power to compel banks to modify the code to reflect certain public policies. For example, some areas have prohibited excessive charges for the use of ATM machines by non-card issuing banks. Predictably, banks have exercised their political power with executive and legislative branch authorities and their ability to mount legal challenges through the courts to fight against this lessening of fees exacted from users. In this context, as in nearly every other example of private proprietary ownership of code used for serious transactions, the owner largely or exclusively calls the tune and the users must dance accordingly. And under this system, the market is largely working. The system of private property in code, enforced through copyright, trademark, patent and trade secrets, is fulfilling the public policy deal underlying Intellectual Property (IP) laws. The IP owners, secure in their government enforced monopoly on the code; have incentives to innovate, to invest in growing a market for the code, and in making code (slowly) better. The Goods and services are ever more abundant and useful. Theoretically, in the event of a truly unconscionable overreaching by code makers, users may hold the coder owner accountable through some legal remedy – perhaps a tort or through the intervention of a government regulator in the public interest. There is also a theoretical connection between an ability of users to “vote with their wallet” (by buying competitors code that is more suitable to their needs) and thereby to hold the writers of the code “answerable”. However, some have argued that the software monopolies and other entrenched interests which largely control the market for such code are only glacially moved by these types of user-accountability and responsiveness measures. Nonetheless, this private ownership based system of code creation has no apparent superior alternative for most types of software. But is this the best – or even a tolerable - answer for code that is used for self-governance rather than for commercial purposes?

The operating thesis of this semester's graduate seminar curriculum is that Open Source Code – of, by and for the users – is a superior approach for communities to enable their online self-governance. In this way, Dean Mitchell's challenge is answered by putting the users in control of the “law making”. This is the clearest expression of self-governance. The eCommerce Architecture Program will make the final code implementing this semester's design freely available to any person

or community via our Internet web site: <http://ecitizen.mit.edu/> The Open Source code software license will require that users openly and freely share any enhancements or other modifications they make to our code. Our code is expected to include and build upon other Open Source code, potentially including slashcode, mysql or php. The special role of self-governance and participatory democracy require that this code be subject to Open Source principles. Self-governance is a unique and important aspect of any free-society and open economy. The software used by a community to self-govern itself can become inextricably tied to the autonomy of the community itself. The “logical” location (as opposed to the “physical” location) housing the community, along with the processes of interacting and the methods by which members are “virtually present” in the community are all fundamental aspects of the group. Self-governance software is unlike a physical building in which debate and votes occur, where a community can move to a different physical location if a landlord unduly raises rent, makes unreasonable demands or otherwise fails to provide for the needs of the occupants. Self-governance software, by contrast to a physical location, may embody the very way a community defines itself and the processes by which it exists. The software and surrounding technologies are the coded instantiation of the community and its laws, such as the balances between privacy and authentication; the unique algorithms underlying filters and the history of user ratings and filtering results; the honed features and flows through the system resulting from accumulated experience of the users; and even the newly evolving parliamentary processes by which decisions are made, archived, and the knowledge managed.

A central component of self-governance is the right to create, use and change the methods by which that governance occurs. This is the centerpiece of autonomy and sovereignty. As such, allowing the code that supports and reflects a community’s self-governance to be owned externally (whether by a for-profit vendor, or any other form of private ownership based upon the proprietary power to exclude use except upon exaction of the terms of the owner) would permit that external party to in effect govern the community through feature selection, support and enhancement determinations; pricing policy and other methods. Ownership is the ultimate trump. It provides the right to prevent others from using the property and the right to designate how, when and by whom the property may – or may not - be used. While the policy premises favoring private property are well know and certainly can be applied to such intangibles as software, databases and even business methods, it is the contention of this academic research initiative that such property mechanisms are not well suited to the needs of open self-governance. For these reasons, it is posited as a design principle for the work of this seminar, that self-governance software is best designed to be Open Source. This emergent political philosophy enjoys [echoes](#) from debates and documents surrounding the founding of the American republic.