

Exploring the Potential for Open-Source Self-Governance

Michael Mussman

Masters student in Communication, State University of New York at Albany

The open-source programming movement has demonstrated that complex projects consisting of intricate language statements can be successfully created by the collaborative efforts of people with varying levels of expertise, from programming gurus to hobbyist coders. This discovery has applications for groups wishing to collectively self-coordinate. Where once it was thought that complicated governing texts could only be created by small groups of experts, a group that endows its entire membership with the power to edit its governing documents (constitution, by-laws, operating policies and processes, legislation, etc.) may find benefit from collecting its members' diverse scrutinies, considerations, and contributions, as is often the case with open source software projects. It is now time for the democratic culture of open software, specifically the welcoming of anyone's contribution, to be carried into the governmental domain. This paper proposes a way of decentralizing a group's self-coordination by open-sourcing their statements of governance. In this way, regular nonelite citizens can have a hand in building their group's policies and rules, just as novice coders can help build open-source digital code.

A company's operating policies and a government's code of laws can be thought of as a set of rules defining how the group is intended to function. Typically, these rules are set and altered exclusively by business owners and lawmakers, leaving most members of an organization with little influence over its rule set. This need not be the case. The nonexclusive altering of rules is a central aspect to a game called "Nomic," which was created by the philosophy and law scholar Peter Suber (1). Nomic begins with a set of initial rules, one of which says, "Changing the rules is a move." It is a game of self-amendment, where any player can alter the rules governing the play. This paper proposes both the recasting of a group's governance as a game of nomic, as well as the open-sourcing of the group's governing language so that it is directly available for collaborative editing by those being governed. As such, the method of self-coordinate described below may be changed by the group members, since every rule is subject to modification in a nomic.

The specific scheme of self-amendment described in this paper has been given the name Efficasync, which is a combination of the words "efficacy" and "synchronous." The former word is defined as, "the capacity to produce a desired result," and the latter word as, "operating in unison." Hence, the conjunction of these words seemed an apt title for a technique offering coherence to a group's efforts at collective self-regulation and goal fulfillment.

Efficasync describes a content management system (CMS) a group can use to structure and regulate the textual contributions offered by its members towards the group's governance. A CMS is a mechanism that facilitates the mutual building, modifying, storing, and sharing of information (2). A group's members use the Efficasync CMS to form and manage an electronic document, called a *metagraph* (fig. 1), which visualizes the presence of and relationships between different governing statements as a network of linked nodes. The metagraph is composed of five types of nodes: *principles, arguments, laws, techniques, and instances*. The metagraph is arranged so that principle nodes occupy the innermost area, with subsequent node types forming concentric areas until reaching the outermost instance nodes.

The five node categories of the metagraph each hold a different type of governing statement. The two most-proximal node categories hold more abstract notions of the group's governance. Principle nodes contain the ideals and ethics by which a group professes to exist. These nodes are the ultimate justifications for all other governing statements. As the second most-proximal type, argument nodes

develop the implications and consequences borne by a set of principles. Arguments rationally and philosophically extend a group's principles, and must be written so that the logic they contain is open for inspection and critique.

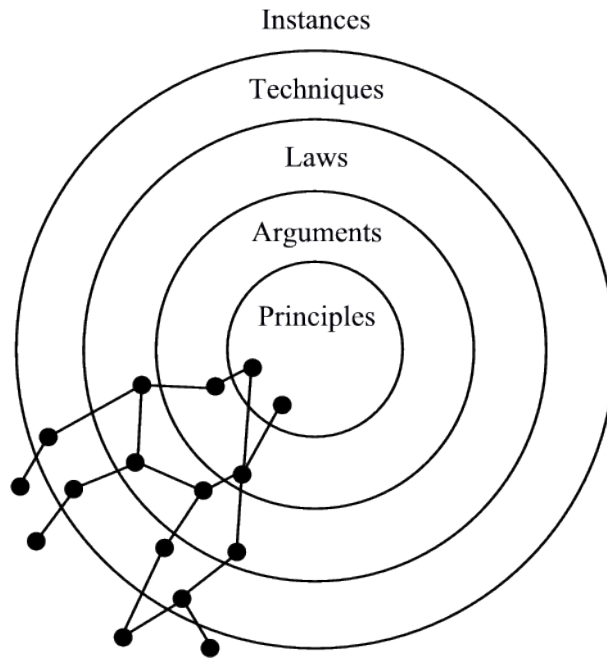


Figure 3: The Metagraph

The three outer node types enclose more practical governing language. Law nodes delimit permitted behavior from unacceptable action, and can describe rights and protections afforded to the group's membership. Next, technique nodes carry recipes explaining how something may or must be done. For instance, the electrical code's method for properly grounding a circuit could be expressed in a technique node. Alternatively, a restaurant could store its recipe for spaghetti sauce in a technique node. A group's techniques are the processes it uses to achieve desired outcomes. Lastly, instance nodes represent specific individuals, groups, institutions, tools and resources which are organized to carry out the techniques upon which they are justified. Instance nodes can be used to record, constrain, mandate, and prescribe who or what will carry out a technique. For instance, a city's metagraph could represent the police department as an instance justified by several law enforcement techniques. In a restaurant's metagraph, a refrigerator instance could be justified by a preservation technique.

Taken as a whole, the metagraph's nodes represent a continuum that bridges a group's abstract foundations (principles) to specific ways of realizing these foundations (techniques and instances). Except for the most central principle nodes, each node must be "justified" by being linked to at least one more-central node that offers some explanation for its existence. If a group's members wish to create a new node, but cannot find a pre-existing justification in their existing metagraph, then either the metagraph must be altered to accommodate the new node or the new node does not belong in the metagraph. This justification requirement is meant to limit contradiction within a group's governance.

A group's metagraph is not static. Rather, in response to the group's changing conditions, members can recommend additions, alterations, and deletions to both the metagraph's nodes as well as the nodes' contents and justifications. Recommendations are carried out if the group ratifies them through their

votes, which occur in voting cycles whose frequency is defined by the group. Efficasync implements the common CMS feature of version control by letting a node concurrently hold several recommendations of its content, each of which is a potential version of the node. Each voting cycle, the group's membership designates one version of each node as active through the maintaining or reallocating of votes among the versions. As long as it earns some votes, inactive versions remain in a node to possibly become active in a later voting cycle. Nodes and justifications may come and go, and their active version may shift over time, as the group evolves to cope with changing circumstances. The techniques listed in this paragraph (and any other of this paper) may be represented as nodes in a group's metagraph, and can be changed by the group's membership.

A group can use Efficasync's CMS to define its own governance operations. For instance, a group may write a law node defining how frequently members may recommend and vote upon changes to nodes and their contents (this has been referred to above as the voting cycle). A group could also create a longer "binding cycle" at whose end all node changes become binding (since it might be impractical for group members to constantly keep up-to-date with several nodes' current versions if the voting cycle is short). Borrowing from Suber's Nomic, a law could be created that requires each node to be designated as either *mutable* or *nonmutable*, and requires that nonmutable nodes first be changed into mutable nodes before their contents can be altered. This law could slow modifications to certain core nodes, allowing more time for group members to consider the proposed changes. A technique node may be written describing how members may recommend changes to the metagraph. Groups could craft a law requiring the complication and publication of lists of new nodes, or nodes with frequently-changing versions, so that members have a quick way of seeing how their metagraph is evolving. These examples demonstrate how Efficasync can be used as a group's internal operating system.

The metagraph is also meant to be a space where a group's differing nodes and versions may compete for prominence. This competition occurs through the group membership's rational/critical debate of the merits and flaws carried by possible nodes and versions. How such debate is implemented can be uniquely decided by each group. Efficasync can be used to periodically record group members' debate in the form of nodes, versions, and justifications. A group can use this never-ending process to rhythmically work out the language they intend to live by. Thus, there is great pressure on members to craft language capable of attaining the approval of a membership ready and able to alter or discard words which fall out of their favor. An Efficasync group with an astute membership may frequently debate and vote among the different nodes and versions which define each element of their governance. Hopefully, such constant vigilance will preserve those nodes and versions that remain beneficial over changing circumstances, alter nodes to the version best suited to the present circumstances, and replace antiquated nodes with ones better suited to upcoming circumstances.

Developing a governance with Efficasync is meant to be somewhat analogous to the way open-source programmers develop a piece of software. A programmer's environment provides the tools to write, update, and debug pieces of digital logic. Likewise, Efficasync's environment describes tools used by a more-general type of programmer called a *citizen*, to write, update, and debug the democracies they live within. Efficasync, like a version control system for software, or like a wiki for text, offers a way to coordinate changes made on a shared set of information (the metagraph) by many different contributors. This coordination may assist a group's efforts at self-management, which will inevitably contain conflicting perspectives, by making available to each member the ability to define aspects of the mechanism used to decide upon the group's constituting language. Hopefully the democratic decentralization of control that may result from open-sourcing governance will prove beneficial to a group's perpetual effort to avoid succumbing to its own complexities, complacencies and contradictions, and create a space in which the group can constructively advance itself.

A summary and a more-detailed explanation of Efficasync is available at <http://www.geocities.com/mikemussman/> and <http://efficasync.blogspot.com/>

References:

(1) Suber, P. (1990). *The paradox of self-amendment: A study of the law, logic, omnipotence, and change*. New York: Peter Lang Publishing. Retrieved June 26th, 2007 from <http://www.earlham.edu/~peters/writing/nomic.htm>.

(2) Author unknown, *What is a content management system, or CMS?* Retrieved April 16th, 2008, from <http://www.contentmanager.eu.com/history.htm>