

Software (,) Politics and Indymedia

Benjamin Mako Hill

mako@debian.org

Table of Contents

1. Introduction and History	1
2. Learning to Look at IMC Software.....	3
2.1. Active's Template and Points of Convergence.....	4
2.2. Active's Spin-Offs and Points of Divergence	5
3. Case Studies.....	6
3.1. SF-Active	6
3.2. Mir.....	8
3.3. FreeForm.....	9
3.4. Other Indymedia Software.....	10
4. Conclusions.....	11

1. Introduction and History

In addition to many other milestones in the activist and anti-globalization communities, the 1999 meeting of the WTO in Seattle marked the birth of the Independent Media Center, also known as Indymedia. Within a year, Indymedia had exploded in size. With slogans like “Be your own media” and a grass-roots publishing structure to back them up, Indymedia’s attempt to provide a

non-corporate and more democratic alternative to mainstream media struck a chord that resonated with activist communities across the globe.

For the first year, Indymedia's face on the web was *Active*, a web application written by a group of Australian hackers for the purpose of facilitating independent media. However, *Active* was unable to keep up with the IMC's tremendous growth in size and political diversity. There were more people interested in reading Indymedia and interacting on the IMC websites than there was bandwidth and computer power to support them. Users and media activists demanded performance, internationalization, flexibility, and features; *Active* and its developers were unable to cater to all of these needs.

The more technically inclined in Indymedia banded together under the auspices of the Indymedia Tech Collective (*IMC-Tech*) and tried to relieve the pressure on *Active's* developers. A year after Seattle, IMC-Tech had installed *Active* so many times that they had automated the process so it was as simple as: "'I'd like an Indymedia site.' *Click Click*. 'Here's your password.'" As people put *Active* in a growing number of places and used it for a growing number of purposes, its shortcomings became more difficult to ignore. Technical "under-the-hood" complaints were paired with calls for new features, increased flexibility and maintainability by less technical volunteers.

Within six months of *Active's* christening in Seattle, IMC-Tech was excitedly discussing specifications for *Active's* replacement, dubbed simply *Active 2*. In these discussions, the political, social and technological collided on the IMC mailing lists. Suggestions for user-moderation, in the manner made famous by websites like *Slashdot* or *Kuro5hin* were viewed by some to be analogous to the advocacy of a minor form of fascism--moderation makes some voices more visible than others and, according to some, institutes an indefensible hierarchy. It was clear that *Active* needed to be replaced but that was more difficult said than done.

The geeks of IMC-Tech were keenly aware that each technological design or set of features creates a particular publishing structure, and, as a result, empowers users to "be their own media" in an equally particular way. In an organization constituted by extremely political individuals who accurately examined the political implications of every technical decision, *Active's* minimalist feature set acted as form of common ground--a least common denominator. *Active 2* was never

written.

There's a story, perhaps apocryphal, that describes a point in Indymedia history when there simultaneously existed three different and disconnected attempts to rewrite *Active* within the IMC Tech community--each attempt kept hidden for fear of political clashes over functionality or design decisions. True or not, it seems possible.

It seems possible that the politicization of each technical decision and long conversations with what appeared to be impossible resolutions made writing a new piece of software as a group seem impossible. It seems possible that Indymedia simply encompasses multifarious political and social ideologies that can only be represented in multiple pieces of software.

Fast Forward a year and a half. While *Active 2* is still little more than a discussion topic on mailing lists, there are at least eight pieces of software in use by Independent Media Centers across the globe and countless slightly modified derivatives. At least five have been written from scratch for use by IMC tech-activists. Others have been adapted from existing pieces of software (like the weblog/content management systems Slash or Drupal) to fit an IMC's needs. Indymedia's political and social differences and their ideas of what the most fair publishing structure does or does not contain have spawned technical divisions in the software it uses.

2. Learning to Look at IMC Software

Other authors have established the existence and importance of the connection between the technological (Lessig calls it *code*) and the social structures that are created by and reflected in technological choices. In a case like Indymedia, this connection is explicit and central.

However, every effective developer knows that if every line of code and every technical detail must first stand up to political debate, every piece of software will have a history like *Active 2's*--no history beyond the conceptual. Every effective developer knows that bugs should be fixed, while features or design decisions should be discussed or debated. This division is a muddled one.

Indymedia provides the perfect venue for an analysis of this muddy distinction. Because most

IMC-Techs see their work on Indymedia software as a part of their political and social activism, they make the political or social motivations behind technical decisions unusually explicit.

But IMC software is also a useful example in that it lets an analyst easily isolate the technical choices. Every piece of IMC software serves the same fundamental function--empower Internet users to be their own media--and does so by following extremely similar models. It is difficult for even an educated visitor to determine which software an IMC is running at first glance.

But the software is *not* all the same. Differences can seem subtle but they are intentional, considered, and extremely important. For some, these “subtle” differences represent the difference between media that is democratic and media that is tyrannical or fascist.

For these reasons, an analysis of the points of convergence and differentiation between different Indymedia software can give us invaluable insight into the nature of the fuzzy area at the intersection of the political, societal and technical.

2.1. Active’s Template and Points of Convergence

As the first piece of Indymedia software and the application behind a majority of Media Center websites, *Active* provides the template on which all other Indymedia software has been based. *Active* development has included little more than minor and necessary bug fixes since it was rolled out in 1999. As it stands, it provides the basic functionality common to all of the major pieces of Indymedia software.

Active’s interface design and layout may be its longest lasting legacy. Its basic design is similar to the design used effectively by many mainstream media and information outlets.

The Indymedia front-page consists of a tool bar on the left with links to documentation, information about Indymedia and the software, other Indymedia sites, a simple search, and links to other IMCs. It usually includes a box through which viewers can subscribe to an email newsletter.

The middle column on the front-page is the most visible and prominent piece on the website. Consequently, the space is reserved for “features.” For IMCs, features usually include headlines,

images, thematic text, and links to a selection of representative or exceptional articles of the theme.

In *Active*, features are produced as HTML fragments. As a result, the ability to create and manipulate features requires technical sophistication, familiarity with HTML, and access to the web server on which *Active* is running. It is almost always necessary to restrict the ability to manipulate features to a small trusted group--often an editorial collective. As a result of this structure, features tend to be thematic and include an aggregation of other content submitted through the more open parts of the publishing structure.

The left column contains an overview of the *newswire*, the feature for which Indymedia is famous. Visitors to the site are allowed to follow a form-based submission process that allows them to upload articles, images, audio, or multimedia, into the newswire. On most Indymedia sites, articles are published in the newswire automatically and immediately and are displayed in reverse chronological order. The newswire lists the article subject (headline) and the date and time on which it was first posted.

Active's users have the option of filtering the newswire by media-type (i.e. only images or only text articles). When they click on an item in the newswire, the article is presented along with comments posted by readers. At the bottom of each article page is a form where readers can join reply to the article or comments themselves. This form, like the media submission form, allows users to specify a name or nickname but provides no system of authentication or name registration.

2.2. Active's Spin-Offs and Points of Divergence

Even the brief description of *Active* above alludes to several of the major points of contention and areas of divergence among those who have orchestrated *Active's* spin-offs.

The way featured articles are implemented acts as an example that exemplifies a larger debate over issues of selection and information hierarchy. On *Active* and most of its derivatives, features are managed by a small group of authenticated users. Issues of control and management are compounded by the fact that the creation of features is often prohibitively complicated technically.

Some feel that by limiting the ability to write features to a particular group, IMCs are privileging one set of viewpoints over others and are creating hierarchies, a form of censorship, and power structures that are no better than those in the corporate media.

Authentication is another contested topic. Several of *Active's* re-writers have seen the lack of user authentication for comments and article publishing as a serious barrier to the development of trust within Indymedia. Others see the anonymity provided by this system as essential to Indymedia's goals.

Other major issues include internationalization and localization--features that *Active* left largely neglected in its first incarnation. As alluded to in the discussion of features, the manner and degree to which developers should simplify *Active's* interface for less technically inclined people has provided yet another nexus for diverging opinions as well. Still more differences are evident around attitudes toward the importance of maintenance and updatability as fixing bugs and tracking changes in deployed copies of *Active* proved extremely problematic.

Each *Active* rewrite has evaluated and approach each of these problems differently.

3. Case Studies

While eight pieces of Indymedia software may seem unnecessary, each piece of software exists because it is slightly different, and in the minds of its author, at least slightly *better* than the available alternatives. Each piece of software reflects the technical, political, and social attitudes and opinions of its authors.

An analysis of these applications as a group is interesting and useful in the context of a larger project to chart the intersection of software and its social and political implications. An in-depth discussion of the individual applications sheds light on the specific political and social points of contention, and the ways in which they have been handled.

3.1. SF-Active

SF-Active began out of a technical need to migrate *Active* from a dependence on one piece of database software (PostgreSQL) to another (MySQL) in the summer of 2000. By changing *Active* to support MySQL, a team of San Francisco hackers forked ¹ *Active* development and *SF-Active* was born. The SF hackers have taken *Active* development in new directions by setting new goals and rewriting almost every piece of the code.

The *SF-Active* hackers want to turn active from a web application used for IMC websites into a set of classes (one can think of classes as little semi-isolated bundles of features or functionality) designed to be useful for a more flexible and dynamic type of Indymedia. Their software handles issues of updatability by sharing programming code among a number of sites running on one machine.

One of *SF-Active*'s goals was to balance the need for moderated news queues without prior restraint censorship. Toward this end, *SF-Active* sites each run multiple news wires. All uploaded news is put directly into an "Other/Breaking News" wire and then is "promoted" to "Local" and "Global" news wires by the sites editors.

Control of features is handled by the strong administration system which makes administration accessible to less-technical inclined users--but only those that have access to the administrative section. As a result of this restriction, *SF-Active* attempted to approach these articles, like *Active*, as *thematic features* that are meant to summarize and reflect on a number of the articles in the newswire.

The *SF-Active* team has not chosen to implement functionality similar to user authentication. Entering a name is free form and unrestricted as in *Active*, but they are considering password authentication in a scheme that they conceive of as a form of "nick registration." Rather than a form of authentication or trust building, nick registration's goal is simply to avoid confusion and allow people to develop reputations. If someone registers "Joe", only they can post as "Joe."

1. The Jargon File describes a fork as "what occurs when two (or more) versions of a software package's source code are being developed in parallel which once shared a common code base, and these multiple versions of the source code have irreconcilable differences between them."

Gekked, a long time *SF-Active* coder, tried to sum up the *SF-Active* philosophy saying that, “*SF-Active* coders do not have any psychotic notions about what IMC is and isn’t. Our experience working with collectives from Chile to Palestine to Iowa tells us that any attempt by engineers to prescribe process will be counterproductive and mostly just annoying.”

SF-Active has documentation in English, Spanish and Italian and “almost-finished” translations from English in Arabic, Turkish, Dutch, and French. It is being used in a quickly growing number of Indymedia sites including <http://sf.indymedia.org> .

3.2. Mir

Mir is a java-based system based on a content management system written by German hackers for the blog-like nadir.org and then adapted to the German Indymedia site. Mir hacker Zapata admits that originally, it was “a system fit for the German IMC way of doing things.” The “German IMC way” reflects a legal environment which prohibits racist, hateful, and revisionist speech in way that necessitates prior restraint story moderation in a way that many IMCs are uncomfortable with.

Over time, *Mir* has grown and changed. Zapata describes his own programming philosophy in stating that, “basically I do not, as a developer, want to dictate how a group should run their site.” While he realizes that this is ultimately unachievable, this attitude has directed *Mir*’s development toward this type of flexibility by emphasizing internationalization, static content, and a dynamic system of customizable categorization.

Mir supports readers/posters and authenticated administrators who can write features, and hide, edit and reclassify postings. While there is currently no method for rating or user moderation, there are plans to allow for different levels of administrators (i.e.. one might writes features, another might edit postings) but this feature remains unimplemented.

Mir’s support for internationalization is good and getting better while projects to add support for easy translations of articles in the newswire are advancing quickly. At the moment, this is

handled by a categorization system that is elegant and flexible in its simplicity. Every article posted to an IMC running *Mir* “belongs” to zero or more categories. Users can sort and group by these categories and *Mir* administrators can set up alternative start pages or news wires for each thematic categories. Since *Mir* also categorizes articles by language and “type” (a type or administrative category that might include “newswire” “feature” or “trash”) *Mir* users can easily separate all content in by language, issue, or type.

For example, featured articles in *Mir* are standard newswire articles with a “feature” type. While from a programmer perspective they are identical to newswire articles, they can only be promoted or classed as features by administrators. In practice, features tend to include a mix of articles promoted from the newswire and the sort of editorial “thematic features” used in *Active* and it’s derivatives.

Mir has been translated from German into English, Spanish, Portuguese, Dutch, Euskera, French, Swedish, Turkish, Chinese, and there is work on Arabic. *Mir* is in action at <http://mir.indymedia.de> and a growing number of Indymedia sites.

3.3. FreeForm

FreeForm is a project started and largely maintained by an Ithaca, New York hacker named Arc. Now just over a year old, the software is nearing a point of widespread usability. It is written in Python and will soon be released as a GNU project.² Its developers are more explicitly interested in issues of software freedom and politics than the other Indymedia projects.

FreeForm is particularly different in that featured articles *all* begin on the newswire and are promoted to feature status. This can be done by editors, the public through a system of rating and moderation, or a combination. *FreeForm*’s authors feel that the more traditional editorial method is unfair and hierarchical, and see an open system of moderation and rating as a way to resolve this problem. Arc feels that “politically, a hierarchal system is never good” and makes this a fun-

2. GNU is a recursive acronym standing for (G)NU’s (N)ot (U)nix. It is the effort led by Richard Stallman and the Free Software Foundation to create a totally free operating system.

damental axiom upon which his technical decisions are made.

Outside of DadaIMC, a non-free application from Baltimore, *FreeForm* is the only piece of software to incorporate user authentication. As Arc puts it, “its all great being all informal and such, but hard to build up a real system of trust when everyone is anonymous.” Arc, whose other goals include the creation of a global Indymedia cryptographic “web of trust,” likes the idea of users being identified as *specific* individuals without this information necessarily being connected to a real names, addresses, IPs, and other traceable information. *FreeForm*'s goal is to facilitate a greater degree of trust, accountability, and reputation building within an Indymedia community.

Finally, *FreeForm* is different in that it is the only IMC application that processes all media content--others simply serve the content as is. *FreeForm* will take an uploaded photo, open it it, look at the resolution, and let the user crop it before saving it again at several different sizes. For sound, *FreeForm* integrates the free streaming program Icecast2 to process media that can be immediately entered into a 24/7 IMC Internet radio stream. However, because *FreeForm* refuses to touch media and multimedia formats that are controlled by patents, the software does not support GIF, MP3, AVI, MPEG, QuickTime or Real Media of any type. Because it promotes freedom from patented and proprietary formats, Arc views this as a feature, not a bug. In terms of multimedia, *FreeForm* supports Ogg Theora, a free multimedia compression standard that will hopefully be fully released by this summer. For media activists dependent and accustomed to other proprietary standards, this can be small consolation.

FreeForm has been translated from English to Spanish and there is ongoing work on a Farsi translation. Example sites can be seen in IMCs in Rochester and Ithaca. Like *Mir*, several non-media and non-political organizations have expressed interest in some of *FreeForm*'s functionality.

3.4. Other Indymedia Software

By no means are the four pieces of Indymedia software described in this article the only pieces of software in use by IMCs. Most notably neglected is a version of the software written by tech activists in Philadelphia in 2000 that is built on top of Slash, the software made famous by Slashdot

and includes a good deal of advanced user-moderation and administration features not found in other Indymedia software. While certainly interesting, *IMC-Slash* never caught on and recent talk suggests that Philadelphia will be moving away from their own software to one of the options mentioned above.

Also important to note is *DadaIMC*. Dada is famous for being easy to install and configure. It is infamous for being the only major non-free IMC software. It was written in Baltimore and along with *FreeForm* is the only piece of software that supports user authentication or nick registration.

The Indymedia in Quebec, which has a reputation for doing things their own way, has written two versions of Indymedia software. Their current web application is based off the web-log style content management system Drupal. Their site has a look, feel, and set of features that are massively different than other Indymedia solutions. A fair analysis of the political and social implications of their software in the context of these other pieces could easily take place in its own article.

4. Conclusions

With a critical eye toward the technology, we can analyze the publishing structures created by these differing applications and their political implications. When we look at these systems together, we begin to get an idea of the difficult balancing acts that the programmer-activists in Indymedia struggle with.

Emphasizing a strong and accessible administration structure creates what some view as an indefensible hierarchy. Deemphasizing the role of editors eliminates thematic features which readers find useful. Cutting users' connection to non-free media formats is at the expense of convenience and access by the majority of Indymedia's current video and audio producers.

These decisions are rarely a matter of right and wrong. They are technical decisions that create a particular publishing environment and reflect a particular political ideology. They each aim to create the "best" possible system for the production and distribution of grass-roots media. There

is a diversity of political and social ideologies within Indymedia and there must be a diversity of software to realize them. There will be no single answer.

And diversity, in terms of ideology and in terms of software, is a good thing. The ability to fork software and modify it to fit your differing needs is one reason that Indymedia is so closely tied to free software--it is essential for software with participatory aspirations. Zapata has worked heavily on *Mir* but is interested in getting involved with *SF-Active* development. He says, "I think it's good to have multiple code bases. It helps decentralize the software development part of Indymedia and having competition stimulates me to improve *Mir*. Also, our users have real choice, and can compare advantages between the code bases and choose the one that fits their needs best."

During the early stages of every new IMC, media activists must wrestle with the question, "Independent from *whom*?" There is clearly no correct answer. Through the creation of multiple pieces of Indymedia software with different and explicitly stated political motivations, the Indymedia movement grants us a meaningful form of freedom--the independence to choose the socio-technical terms on which we communicate.