



Choosing a platform for the telecentre.org network

Telecentre.org has outlined plans for a distributed network of telecentre network sites around the world, which will include a site for telecentre.org itself. Mark Surman engaged me to help specify the platform requirements for creating a distributed network of sites that will share and exchange content.

This document outlines the process we followed for choosing a telecentre.org web platform; the platform we ultimately selected was Drupal. For the specific requirements and overall web strategy that guided this process please see the separate telecentre.org online strategy document. The overall vision is of a network of sites that will likely run on a variety of platforms, including a growing number of sites that run on our Drupal-based "support net in a box".

RSS offers the obvious means of sharing and aggregating information across the sites in this network. Because telecentre end users are often accessing the Internet with low-bandwidth or intermittent connectivity, and RSS-based strategy also offers options for getting content easily and storing it locally for later reading. By tagging (assigning keywords to) as much of the sites' content as possible, it will be easy to organize content across sites and even across multiple languages. The implementation of RSS and tagging was thus central to our review of software options.

Our search for the optimal web platform for the telecentre.org support-network-in-a-box (SNIAB) focused on options that might combine content management features (like the ability to easily create and edit web pages, register users, and automatically create navigation structures) with community and collaboration features (like RSS, blogging and wikis). We initially expected to choose a separate (though compatible) platform for the event-in-a-box (EIAB), with some overlapping functionality. As our vision for both SNIAB and EIAB grew more nuanced and concrete, we concluded that there would be significant advantages to selecting the same platform for both purposes. The primary advantage would be the ease of incorporating event-driven content into the flow of content among telecentre network sites; a secondary consideration was that it would be easier for telecentre partners to become familiar with a single tool, rather than two.

We gathered preliminary candidates by reviewing the tools used to create sites with community features, reading online discussions of CMS tools for

the non-profit sector, and soliciting suggestions from colleagues. We wanted to find a tool that wasn't just a CMS with a bunch of blog-like features attached to it; we wanted a tool that was fundamentally suited to a distributed network in which content would be continuously exchanged among sites. We also wanted to ensure that sites that were not built on the core platform would still be able to access content from the sites that were; we recognized that RSS was the only standard that could serve this purpose.

The goal of establishing a distributed network that circulated content via RSS led us to three core questions that served as a filter on our potential platform:

1. Has this platform been used to rapidly deploy a large number of interlinked sites from a basic template?
2. How widely does the platform integrate RSS into its structure – both in terms of creating outbound RSS feeds and in enabling easy, flexible aggregation of inbound RSS feeds?
3. Does this platform support blogs or wikis, which are the most common tools for online community and collaboration?

These three questions provided a litmus test for whether a platform had "community DNA", and allowed us to quickly narrow a long list of possibilities. Some of the initial options that were set aside due to lack of community functionality included:

Ekton: no blogging
Kintera: minimal RSS orientation
MCMS: no blogging
Red Dot: blogging not core
Sharepoint: no blogging
SiteRefresh: RSS feeds are not core to software

We took a closer look at seven options:

APC ActionApps: The distributed network model is central to the design of ActionApps, which lets users share "slices" of content with other ActionApps sites. The limitation of the slice model is that it is a content distribution scheme that is specific to ActionApps, so any telecentre network sites running on other systems would not be able to participate in the content-sharing scheme. While there is some RSS support it is not the primary mechanism for content-sharing, and both blogging and wiki functionality are still under development. We were also concerned that the small developer community for ActionApps did not promise significant growth on the platform.

Mambo: The user and administrative interface for Mambo was one of the best-designed options on our shortlist. We were also interested to note the emergence of Soapbox, a Mambo toolset geared towards the nonprofit community. The Soapbox toolset is however limited (at this point) to integration with the Democracy in Action CRM, and to single sign-on among sites. The information architecture of Mambo itself proved incompatible with the telecentre.org project, since it is structured around categorization (rather than tagging), which in practice imposes such limitations as precluding multiple categorization of inbound RSS feeds. Since much of the telecentre.org network's content will need to be distributed to multiple categories (e.g. a story on a Bolivian wifi project needs to be tagged "wifi" and "Latin America"), the categorization structure was a deal-breaker.

Plone: As a CMS based on the Zope platform, Plone offers much greater programming extensibility than other CMS options we considered. The flip side of this virtue is that Plone's relative advantages are much less compelling for a project that (like telecentre.org) specifically wants to limit its custom programming commitments. Ultimately our biggest concern was that Plone's RSS aggregation capacity was not part of its standard install; while adding an aggregation module is a trivial technical challenge, the lack of native aggregation support spoke to the platform's orientation towards single-site content management rather than distributed community.

SocialText: We used SocialText as a platform for circulating our requirements, which gave us an opportunity to try out the software in action. SocialText was appealing primarily as an EIAB option, and once we decided to focus on choosing a single platform for both EIAB and SNIAB, SocialText made little sense. In particular we found that its pricing model – based on per-user or per-day pricing – made little sense for our purposes, and reflected the fact that it is really a collaboration platform rather than a content platform.

Userland Manila: As a CMS that began as a blogging platform, Manila seemed like a promising option for a community-oriented web platform. However it lacked many key features we required, including wiki support and flexible, native RSS support. While it would be possible to extend Userland with scripts that allowed (for example) aggregation of different RSS feeds to different areas of a site, the extent of custom-scripting needed to effect the kinds of RSS control we'd like indicated that Userland was oriented to single-site CSS rather than distributed network community.

TIG (Taking it Global): Taking It Global is an NGO that runs both its own TIG community on a homegrown platform, and offers that platform to other users (such as the Digital Divide Network). The platform is specifically oriented toward supporting multiple sub-sites, but its model for supporting multiple sites is based on storing all sites on a single server (and in a single database) rather than as a truly distributed network; this would make it difficult to integrate non-TIG sites into a TIG-based telecentre.org network. While we

were confident in TIG's ability to quickly and efficiently deliver additional programming as needed, the relatively long list of features that would require custom programming for our purposes (site-wide tagging support, separate aggregation pages, wikis) again suggested that the TIG platform was not fundamentally oriented towards an RSS-based distributed network.

Drupal: With a fast-growing user base in the non-profit sector, Drupal's strong online community focus made it an appealing prospect. Nonetheless we were concerned about documentation and interface issues and its wiki support. Our wiki concerns were resolved by a demonstration of a Drupal site with installed wiki-like features that fully met our needs. Documentation remains a concern but appears to be a priority for both Bryght and CivicSpace, two major players in the Drupal community. Likewise an improved administrative interface is high on the development agenda at CivicSpace, and may be partly manageable in the shorter run by implementing a custom theme for administrators.

Most importantly, Drupal was alone among all CMS options in its compatibility with a distributed network approach. The platform is essentially built for exactly this kind of approach: it supports ubiquitous outbound RSS feeds, complex aggregation of inbound feeds, per-feed or per-item non-exclusive tagging, and native support for blogging. Compared to the other options, which are virtually all CMS platforms that have developed distributed community features, Drupal is innately oriented towards community networking and distributed content creation. The following outlines how we anticipate using particular features of the Drupal platform to support core elements of the telecentre.org web strategy.

Questions? Please contact:

Alexandra Samuel
alex@alexandrasamuel.com
August 2, 2005



This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike License.