Towards a Live Social Science on the Web

A Narrative of the Software Project, IssueAtlas.net

by Richard Rogers

Software development is excruciatingly detailed, and tricky. It's a bit like turbulence theory, if that allusion is not too overused. Changesto one small spec reverberate across the entire software project. For the purposes of a narrative, it is perhaps best for me here to write about the changes and reverberations, and refer the reader to project documentation for the more official view.[1]

The project is really two projects in one: a co-link machine, and an atlas of issue network maps generated by the machine. The server-side software, the co-link machine, crawls a set of specified sites, brings back the sites' outgoing links, looks for common outgoing links (the co-links), and delivers the co-links by name (e.g., greenpeace.org) and by category (.gov, .com, .org, .edu, and their country-specific equivalents) to an XML file. Writing the software that generates the XML file is basically the job of Oneworld International, London, whom I hired with a grant from the Soros Internet Program, New York. What goes into the software, and what is done with XML output, is basically the job of mine and the other researchers and web geographers involved. The output is to be rendered into maps that make up the atlas. This is the second part of the project: actually having a set of maps that not only make sense to the viewers, but also capture the state of an issue network on the Web at any given time. That state may be counter-intuitive, i.e., it may exclude actors you'd expect to be on the map, and include the converse. The method must be robust enough to withstand the incredulous eye, expecting x and getting y and z (and only a bit of x). Moreover, the software has to work, and has to be workable for issue cartographers and also for less sophisticated users, who nevertheless should be trained to operate the software. (One of our constant reminders to ourselves is that we're not rebuilding Google. On this project, there's hardly a large database in sight.) And, finally, the maps have to be decent on screen, and printable, not only for the map readers but also for the purposes of later making a printed Issue Network Atlas.

I first hired two designers, graduates of the Design Academy in Eindhoven, to do not only the look, feel and object design of the piece of software as well as the entire issueatlas.net site, but to deal with the myriad problems of navigation and use sense. I also secured 'proper users' at this early stage, former students of mine from the University of Vienna, who'd already suffered through 3 of my classes and who understand the theory and method of network location and issue mapping. These folks would be the co-cartographers and the user-testers, and attend a series of four mapping workshops where we would push the theory and standardise the practice. Early on I also made sure that I secured the services of my right-hand theorist from the University of Amsterdam (Noortje Marres), in order to continually talk through the logics and to make sure the issue selection and the mapping made sense. She has run a series of four mapping workshops in Budapest, entitled 'The Social Life of Issues'. At the workshops we also joined by a colleague from Boston College (Greg Elmer), who often plays the skeptic.

The Narrative Specification we wrote at the start of the project defined a 'narrative algorithm' which crawls sites, and returns co-links. It was specified to bring back not co-sites, but co-pages. This was the first conundrum. If you generate a map of relevant pages related to an issue, and more than one of those pages come from a single organisational site, the map may look strange. What's Greenpeace doing on the map three times? On the other hand, we are looking for the most relevant stuff on the web per issue. If geocities or oneworld is hosting a set of distinct sites, then we don't want the crawler to bring back 3 geocities sites, and 4 oneworld sites, if geocities and oneworld are only hosting others-organizations divided from a mother host by a mere slash. Then you'd have quite the inaccurate picture. So the solution is two-fold. We build a 'switch' that allows the cartographer to 'match pages' or 'match sites', and once the network is returned to you, you may 'edit' it. If you've matched pages, you select the page of a site that appears most frequently, so there's only one site per map, but with the most relevant page. If you match sites, you delete the double sites. For the geocities case cited above, you can check for a network in two ways (pages the first time, sites the second time, say), compare them, and be reasonably assured that eventually you have the right nodes for the map.

Building in switches - allowing one or another method to be employed by virtue of turning on and off particular settings - has been our solution to many of the other conundrums. For example, the Narrative Specification also called for the starting points to be privileged. Starting points are a set of URLs one enters initially into the software, to be crawled to bring back a network of interlinked sites. Starting points are privileged in that you find their external links, and then you crawl the starting points and the external links together to find external links anew (your 'pool' or population in the parlance). The next iteration of the co-link analysis returns your sample, in which you seek a 'network'. This 'biasing' of starting points is one heuristic 'trick' to the algorithm, for it ensures that the network you capture has a semblance of the starting points you entered. It meets some expectations of the issue network seeker, whilst also producing a few new unexpected actors in the outcome. (It also assumes some sophistication in choosing the initial starting points.) The Amsterdam theorist was against this, for she believes in 'brutal co-link' and hard network location analysis (my phrases), while the project scientist (yours truly) believes that without privileged starting points you will get 'issue drift'. Global civil society, we both agree, is not really made up of single-issue actors (as old social movement theory has it), but rather of a more free-floating protest network potential (to paraphrase Heidegger and Dieter Rucht) that moves from issue to issue. If you do brutal co-link analysis, and only match sites, you run the risk of having a similar network for every issue. Of course, I exaggerate. I write that one believes in this, the other in that, but it's really a heated and friendly debate, to be resolved in future on the basis of results, probably in the same manner as debates are resolved around the 'best' search engine - through empirical practice. In the event, we've built in a switch to allow both methods.

Another example of the switch solution is the number of iterations one requires in order to find a network. By iterations, I mean the number of times a set of sites are crawled and common external links returned. It's the network location heuristic. The minimum requirement is really 3 iterations (with or without starting points privileged), so we have made this the default setting, but the program allows as many iterations as one wishes. Also the depth of the crawls of the sites was an issue solved by a setting. So on the crawler interface, you'll have a number of settings (privilege starting points [default=off], sampling iterations [default=3], crawl depth [default=2]). There's also a setting called 'use blacklist', with the default on. This blacklist is a site/page exclusion list that excludes software download pages and the like. Some cartographers protested that you actually may wish to map this sort of 'issue', so we allow you to turn off the stop list. A debate continues currently on whether you should be able to view that list (yes), but also edit it and save it anew (probably not). There you get into another kind of privileges debate, i.e., whether any user or just the administrator or some kind of user in between can save a new blacklist, and then this blacklist becomes the default list, etc. As I said at the outset, the moment a spec is changed or added for political or other 'vibe' reasons, it reverberates across the many other pieces of the puzzle. Another design change because of 'vibe' reasons is that the issueatlas.net homepage does not have a cartographer's login page link (or shall I say 'object') on it, for such a feature is off-putting for the underprivileged. Cartographers enter through a separate door.

The original project name is Live Issue Atlas. In discussions between the Soros Internet Program and myself before the grant was allocated, we debated whether the project was primarily about making a piece of software or making an issue atlas. My response was that we'd make the software and the atlas to budget and schedule, but the 'live' part would have to wait another day. In my view, an atlas, or a set of maps, becomes 'live' when they know when to refresh themselves. They'd know to refresh themselves, I believe, if the network they're based on is hot, i.e., is increasing the frequency of its page modification behaviour, perhaps increasing the link density of its network. In order to have a network (and a map) learn about itself well enough to refresh itself, it needs to first schedule a series of refreshing crawls, and note the differences in heat over time. The hotter it is in comparison to a set of previous crawls, the more frequently it should refresh itself. If it learns, the atlas is not only live, but it's also webby and self-reliant. It's webby in the sense that it is responding to Web dynamics, and is responsive to web users, who would be sensing for any number of (online and offline media) reasons that an issue is heating up. If the live atlas meets that expectation for those web users, then it's timely. (Perhaps it could be said to be performing live social science.) Also, it could alert folks to particular issues heating up. Finally and perhaps most importantly, it's self-reliant - in the sense that it maintains itself, sort of like artificial life.

Some issues emerge if you try to design this, one of the larger of which is the effects of dynamic html. My solution was to exclude those pages from the refresh analysis whose datestamp is about the same time as the crawl was performed. We shall try to make the maps learn in future, but for the time being we have built in a scheduler for regularly scheduled refreshes of the network. For operators sensing a heated issue, that refresh schedule could be made shorter, what have you. Thus the atlas will not be 'live' in the sense above, but the sets of maps will still be able to show 'evolution' of an issue over time through the scheduler feature.

But refresh what? One could plug the starting points back in, and determine quite wholesale changes, potentially, or, as our solution has it, especially after a long discussion with the Oneworld programmers (Cambridge University math graduates - this came in handy), we can note the smaller changes in the network (who's now in, who's now out), by taking the results of the last iteration as the new starting points. So, we are refreshing the 'network' on its own terms. The starting points become a little less relevant, and thus partially address some cartographers' concerns of bias in starting point selection, i.e., whether the 'network' is ultimately more a product of the starting points then web issue network dynamics.

Above, I mentioned the default number of iterations as well as the default crawl level. This brings me to the most frustrating aspect of the software, and that's the speed at which it returns a query (and the planning, and administrator crawl cancellation moments, that will have to go into making the atlas proper). I'll preface this by reiterating that we're not Google, and we're not operating a database farm. We're operating a lonely Oneworld server, with some distance from a backbone. The maximum number of starting points that can be crawled - or is 'spec'd to be crawled; it could be more - is 300. Say you start with 300 URLs, you set the crawler depth to 3, and iterations to 3, the crawler could be working perhaps for hours. This was first brought to my specific attention when we discussed 'email notification'. The cartographer is notified by email when the network location crawling and co-link analysis are completed. (The cartographer also may move to his/her member's page [renamed cartographer's page], and note a completed crawl.) Really the only way to speed up this process is to pull a SETI, and do some distributed computing. I have asked our PERL programmer to write a spec on devising a piece of downloadable software - like the SETI screensaver - that has a machine, once slumbering or perhaps not slumbering at all, contact the server-side software, telling it that more bandwidth and machine power are available for crawling operations, and thereby extending a helping hand for the cause. (This presupposes the emergence of a 'community'.)

Here we should segue from the most frustrating to perhaps the most exciting aspect of the project. For approximately three months, few people in the project - maybe I should just say I - really had much of an idea about how we would render the XML file (the crawl results) into graphics on the fly. We talked a little about scalable vector graphics, but this requires some code to be written that parses the XML - something the Eindhoven designers aren't equipped to do. The arrival of a Viennese PERL programmer has given us means in executing this crucial part of the project. Subsequent discussions amongst the designers and programmers revealed an intriguing project option, and perhaps the final large-ish change to original 'how to build the engine and the atlas' narrative specification, written in February 2001.

A few weeks ago I assisted a colleague at the weekend in mapping the Russian HIV-AIDS network on the Web. Up until that point most of the issue network mapping work had been done from a 'govcom.org' perspective. That is to say, we have been looking for the composition of issue networks (and the extent of the debate on issues within them) amongst three to four leading actor types per issue - governments, companies, ngo's and scientific institutes. Noting that an issue is occupied only by com's and org's indicates one thing, whilst one occupied by gov's and com's another, and gov's and org's another still, and by gov's, com's and org's yet another. (I'll spare the analytical framework at this point.) Knowing its composition (who's who) and its composition type (e.g., what we call an 'unholy alliance' by .gov and .com, as above) was enough to build theory, talk practice, make claims. The colleague, however, was interested in the interplay between national and international groups, and whether the nationals defined the problem (and their audiences) in one way, and the internationals in another. She also was looking for the best-positioned international actors in the Russian network. In fact, she had many new questions because she came to the problem with a new pre-classification of node types. The breakthrough came when we actually mapped the two groupings, in a two-node-type scheme. Before coming to the breakthrough, I should mention that the visualisation of the Russian HIV-AIDS map was inspired by the conversations with Oneworld programmers, and the idea that a network is refreshed by using the last iteration outcomes as starting points. We can visualise not only the final network but also those parties from the last iteration that did not make the final network. Thus we used a kind of Turkish eye visualisation, a circle within a circle with the bottoms of both circles meeting. This shows who's in, and who's *just* out - and perhaps obviously (only to the cartographers) - from whom those actors just outside the circle would have to receive a link in order to make it into the inner circle, and count as relevant in the issue network on the Web.

The colleague was interested in a different node type naming convention (international and Russian), and thus only a two-colour as opposed to the 4/5 colour gov.com.org.edu.country scheme. Recall that once the crawler returns your network (you've been notified by email to go to your cartographer's page), you may edit your network. This page is called the 'network tuner', where upon tuning and saving, you render your network into an actual map. At the tuner, you may edit the URLs; you may also edit the node names. (You also can raise the authority of your network from co-link to tri-link and upwards, and back downwards again before rendering the map.) Now, at the network tuner, we decided that you may also edit your node types, as well as perhaps the node colours. In those fields, the gov.com.org.edu.country scheme will automatically appear as suggestions (or defaults), but you may edit them. Once edited, you save your network which could just as well be with your own node naming, node type naming, and node colouring assignations. Thus, all of a sudden, we have a general mapping tool, with the gov.com.org.edu.country as suggested frame only.

There are more details to be perused in the project documentation on issueatlas.net, but allow me to conclude with map viewing, the piece de resistance of the project. The maps come as html tables for the low-end computing crowd. They also come in SVG, which requires a plug-in. They also may come as .png files - with the usability of .gifs or .jpegs but without the proprietary stipulations. Now, the SVGs will come in layers, which you may turn on and off. If you have a gov.com.org.edu.country scheme (because you've kept the defaults, saved and rendered that kind of network), you may turn links on and off, and node types and links on and off, from a straightforward gov, com, org, edu-type legend. (It wouldn't make sense to turn off only node types and be left with just links, unless you're an artist perhaps. Though we may wish to view collective link shapes at some point.) Intriguingly, if you have tuned and saved your network with a non-gov.com.org.edu.country scheme, your map legend will be dynamically generated, so you can turn on and off mylinks, and mynode types and mylinks.

(I gest with the MyNegroponte allusion.) Here we still have to sort out the colour paletting for new and different maps so that they do not correspond to the gov.com.org.edu.country colourisation. Nevertheless, there it is - a server-side generic network location and mapping tool, based on basic scientometric analysis, with settings to convince even rather hardened webometric methodologists.

[1] See http://www.issueatlas.net.