# ConcensUS – a site to bring us together

by discussing important issues in a democratic environment

# Problem

As advertisers drive not only web content but also discussion topics, the information that is most prevalent on the Internet is not being driven by social need. Other important issues and perspectives can be drowned out or not even addressed because they lack a revenue stream. While traditional media bundles different viewpoints and provide an overall perspective, the unbundling phenomenon of the Web 2.0 leads to an unbalanced perspective. We are in danger of succumbing to sensationalism and one-sided reporting.

# Solution

An aggregator app consolidates information available from different perspectives, and crowd sourcing determines the real issues that are important to the people.

The citizen can view topics that are relevant and important to them, and gain an insight into both sides of the story. They can express their opinion and conviction, and view how others feel about the issue. If a topic has been voted on, they can see how their representatives have voted.

Looking at the crowd-sourced data the other way, the representative can see how people within their district feel about a topic, compared with the regional average. This data should be shown graphically.

# Scope

To fit within the confines of the Hackathon, the team identified a general framework but then scoped it down so that a prototype could be demonstrated by closing. Since the one team member knew Drupal Content Management System (CMS), and since many governments and other aggregators are using Drupal for CMS, we chose to fit our design into a easily constructed Drupal design.

There are many ways we thought about grouping or filtering topics: by geographic region and by personal relevance. Geographic Regions include:

* Local / County / School District
* State
* National /Federal
* International / Global (\*version 2)

Limiting or presenting articles “by personal preference or relevance” is a both a subjective and an exposure problem. If you only say you are interested in topics 1 and 2, and are never exposed to a particular topic, you will never develop an interest in topic 3. It also means that someone has to subjectively categorize a topic. SPLOST, for example – is it a taxation issue, or transportation issue, or state budget issue? If I am interested in one or none of those, but yet I am a commuter that would benefit from the law and be willing to pay for it, I might never see it. Therefore, we did not try to filter the list of ‘Hot Topics’ and instead present it to the user in a list based on their zip code (group) based on groups they want to belong to. Since groups are geographically organized, the editor need only assign an issue to the geography that it affects.

The user can vote a topic ‘hot or not’ to them; the list of topics are presented to them in three sections: un-voted, voted hot, and voted not (in case they change their mind). (version 2?)

# Team

The June Atlatna RHoK team was comprised of:

Monique Boea, Data scraper, CFM

Jason Crim, Web design and API

John Kirkley, lead Drupal CMS guru

Maggie Lee, Journalist

Dave Savage, Video

Roger Wickes, team lead

# Requirements

The Agile project management method was used to define scope, refine scope, and iterate to ensure we had a demonstrable prototype by the deadline 24 hours away.

# Actors

Citizen – a person who is interested in topics

Editor – identifies topics to be discussed, cites credible sources and if it advocates for or against the topic.

Representative – Citizen with additional functionality to ensure they can see how their constituents feel about a topic.

# Use Cases (User Stories)

Citizen register to use the site, indicates their preferences, and subscribes to Groups. Groups control what topics are relevant to them. If they live in Atlanta GA and their Mom is in Rome Italy, then they may want to subscribe to see issues both in Atlanta and in Rome.

Citizen views the list of Current Topics for a particular zip code

Citizen views a Topic Summary: title, synopsis, pro, con.

Citizen researches a topic by following the provided links.

Citizen votes a topic hot or not to them. Citizen provides a range between -5 (strongly against) though 0 (neutral) to +5 (strongly for) and indicates their conviction (-5 = should not be an issue, +5 = feels very strongly about the issue)

Citizen views details on how fellow citizens feel about a topic. This shows them a heat map (scatter chart) that plots all responses or just responses from citizens in their district.

Citizen changes their mind on whether a topic is hot or not

Editor approves new account login.

Editor becomes aware of a topic and adds a hot topic

Editor cites credible sources

Editor curates user submissions (links or documents)

Editor moderates crowd-sourced comments based on policies

Representative views how citizens in their district voted on a particular issue, in context with how many voters are registered or elligible

# Policies

User content and comments will be moderated and removed if they do not add substance to the conversation.

# Objects

In the future, a robust logical object model would be nice, but for now, here is a list:

* Article: a title, url, and text that takes a position (for (pro), against (con), or neutral) on a particular topic.
* Citizen
* Representative
* Legislative body
* State
* Country
* Topic
* Topic Vote
* Links

Architecture

We used Drupal, and the free Drupal site hosting at

Views/Forms

Current Scope

Known Issues

# Issues

Sock Puppets: people that are paid to vote on issues or blog about certain issues, even though they personally do not feel that way. While it is illegal for a registered politician to accept bribes for votes, there is no law or a current way to police such behavior from the general public. Our feeling is that it will be a self-correcting behavior or will be a minority behavior.

Squeaky Wheel: Any time you crowd-source an opinion poll, you get people that will try to vote on a topic many times, or consistently vote a certain way, while everyone else is too busy to use the site, or even does not know the site exists. That leads to a skewing of the data and potential misrepresentation. To mitigate this, we have the Editor approve logins; that process and data could be tightened up or even connected to the voter registration system in the future. We want to show not only how many of the users voted one way or another, but also wanted to show the total population that could vote. However, we could not figure out the APIs in the time allotted.

Voter Uniqueness: Drupal restricts voting on a topic to one vote per user. However, there is nothing stopping someone from registering multiple times.

Legal Voters: There is nothing stopping under-age, unregistered, or even out of state/out of country people from registering and ‘voting’ on a topic

List of Hot Topics: It is the editor’s job to be aware of topics and bills that come up, which is a manual process and is error-prone.

Editor Research and Bias: Selecting relevant articles from credible sources (biased one way or the other) is a manual process. The editor has to be professional and ensure that both sides of the story are presented and represented through articles that provide balanced reporting.

District Maps Change: we initially wanted to make the site automagically find a citizen’s representatives at the local, state and federal level. However, mapping a person’s zip code to their voting districts is not possible, since a zip code may span voting districts. What GA voters need to do is visit http://mvp.sos.state.ga.us to find the US Congress, GA Senate, and GA House representatives.

Data Sources: Getting the names of the GA State House of Representatives and how they voted was tough, since the site had a malformed XML

Voting: initially we planned for citizen to indicate a numeric vote for/against as on a scale, as well as how strongly the felt about the issue. This data could be plotted as a heat map. In the interests of KISS, we reduced the voting to simple Yes/No/Undecided and a pie chart to show results.

# Future Direction

A voter can go to the Secretary of State site and find out their state reps, and then to the Federal House of Representatives and find out who their Senator Who are my reps?

User uniqueness (prevent multiple logins)

Scraping: Automate getting the list of districts and the current rep

Automate getting the voting record of a rep on a piece of legislature

Expand the database to link user <- district - representative

Scrape the link between a district and the census blocks that it encompasses, so that we can then

Automate the ESRI pull to count voters (eligible and registered) per district – surely this has been done before

# Lessons Learned

Have an idea

Drawing content from scrapers is problematic because the sites are changed structurally so often. The Editor must have the technical skills to repair a scraper. Don’t assume the site produces a valid XML document or has no broken links

Rate amount of spin in texts by doing a word match most often used by public relations ppl.

Data sources

<http://panda.readthedocs.org/en/latest/> **PANDA: A Newsroom Data Appliance**

<http://sampleserver1.arcgisonline.com/ArcGIS/rest/services> - Population info

localecrocy.org – see matild’s email

Redistricting in GA: <http://www.legis.ga.gov/Joint/reapportionment/en-US/default.aspx>

We launched a new comments template this week our SCOTUS coverage, framing conversation with a specific question and asking people to vote one way or another before commenting. We then removed threading and encouraged folks instead to recommend the best arguments. Pretty pleased with the result (and confess being inspired to do this by PopVox). Curious what you all think, how you'd improve: [http://wapo.st/GVRA5p](http://www.facebook.com/l/XAQGFEQDyAQEaA1tC7jsagmpj_xbgXaOL0KY5GCG37-ESow/wapo.st/GVRA5p)

|  |  |
| --- | --- |
|  | **[The health-care law and severability: Make your case - The Washington Post](http://www.facebook.com/l/PAQEalW39AQFgBSRl8xQI2lrNataoNVnbti0ttkHZ4BBgPw/wapo.st/GVRA5p)**  **[www.washingtonpost.com](http://www.facebook.com/l/PAQEalW39AQFgBSRl8xQI2lrNataoNVnbti0ttkHZ4BBgPw/wapo.st/GVRA5p)** |
|  |

<https://github.com/sunlightlabs> - Government site scrapers in ruby, php, python. Quite a few repositories to browse.