

11 Jul Call

Gretchen, Bill, Chris

Gretchen intro

Prepping for July 27

1. GA Design task: What about copy paste?
2. GA Production task: Move scenario to Illustrator
3. GA Production task: Break out scenario into smooth pass
4. CN Production task: Exception cases
5. CN Production task: Produce deck
6. CN Research task: Structure research questions to get the right feedback

Sprint 1 Week 5

13–17 Jun

The wireframe depicts a document review application. At the top, a navigation bar includes 'My Madison.io', 'ABOUT', 'MY DOCUMENTS', 'DOCUMENT TYPES', and a user status 'Logged in as: Gayle: Fairview'. Below this, a header section displays 'Author: Paul G.B.', 'State: Private to Fairview', and a 'SAVED' status. A central message states, 'You've been asked to Review & Comment by 23 JUN 2017 (12 days left)'. The main content area features a rich text editor with a toolbar containing bold, italic, underline, bulleted list, numbered list, indent, outdent, undo, and redo icons. The text within the editor reads: 'RESOLUTION', 'WHEREAS, um "umum" umum umum', and 'BE IT RESOLVED, um um um um'. To the left of the editor is a sidebar with a 'Resolution' dropdown menu and a 'tracking sheet' section containing 'title', 'Whereas 1', 'Resolution 1', 'Resolution 2', and 'Signatures', each with an 'add' button. At the bottom, a footer bar shows 'Contact Paul G.B.' and a 'SUBMIT AS REVIEWED' button. Handwritten orange annotations include: 'type/slector' pointing to the sidebar; 'Login/out Name Group if any' pointing to the top navigation; 'Metadata' and 'Formatting' pointing to the header and toolbar respectively; 'Commenting (placeholder)' pointing to a comment icon; and 'ACTION BAR' pointing to the bottom bar.

Thinking though some of our chat on Friday, I've drawn some changes to the layout.

- I added the **document type** as a selector above the structures pane. I'm not 100% certain how this works in practice, so consider it placeholder. I just wanted to account for it from the current Madison header.
(It seems really tricky to change doc type mid-stream: What happens if she changes a resolution to an ordinance? All the structures will suddenly become problematic. It might be better to handle that from My Documents and walk her through a wizard to help her convert it, but that should become clear in a later scenario.)
- I drew the Mymadison nav pane rather than handwave it. There are some changes from the current version.
 - I suggesting moving the login to the far right, since it's more complicated and different than the general nav.
 - I adjusted the login to show her name. I presume she has a profile page somewhere where she or an admin can provide it.
 - Additionally, you'll see how I'm suggesting to handle it if she's in multiple groups, as a selector to the right of her name. It shouldn't be a form-type dropdown, since that will draw too much attention to itself. But a triangle to indicate that she can change it if she needs to. If Gayle is only in one group, we can hide this.
- There's a **metadata panel** to house information around her workflow/task. In this case, she's being asked to comment on a document by another author.
 - Who that author is
 - The state of the document (from the existing header, helps her understand who can see her comments)
 - Her task
 - The deadline for her task, both as a calendar date and a time-remaining display.
- The **formatting bar** is drawn rather than handwaved
- There's a new **action panel** at the bottom
 - It's placed at the bottom as the "last place" on the screen for when she finishes her task.
 - It gives her ways to contact the author to discuss it if she needs to
 - Any action buttons (like you see) here related to the task. Another example of an action might be "send for review" (if she's the author).
-

-

Sprint 1 Week 4

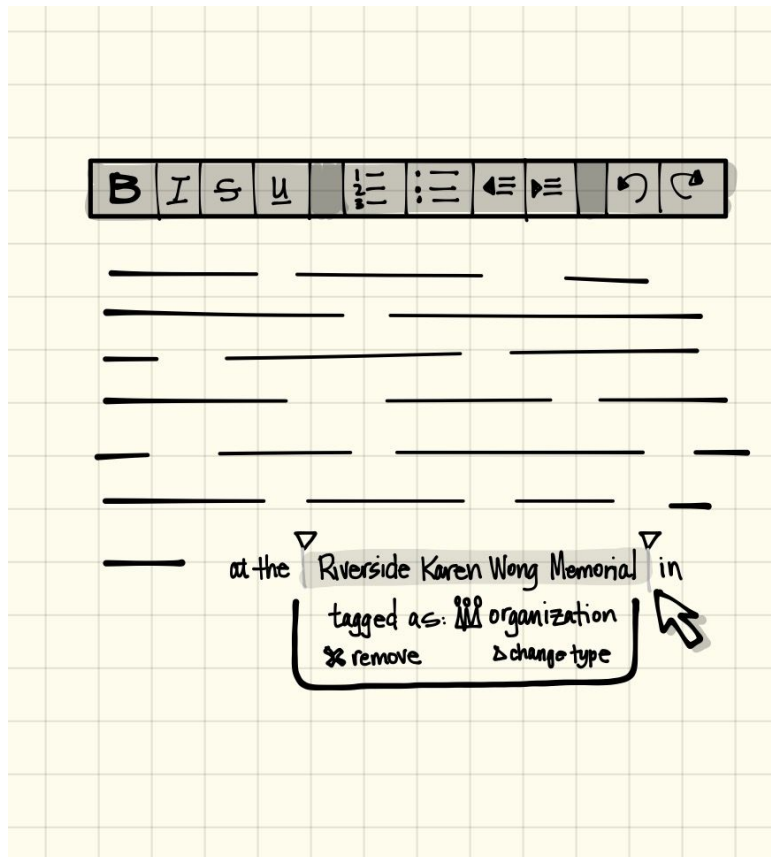
06–10 Jun

Feedback from Week 2 of this sprint indicated that custom form creation or a form builder system was way beyond the scope of the current engagement. So attention is heading back to details in the free-form legislation tool.

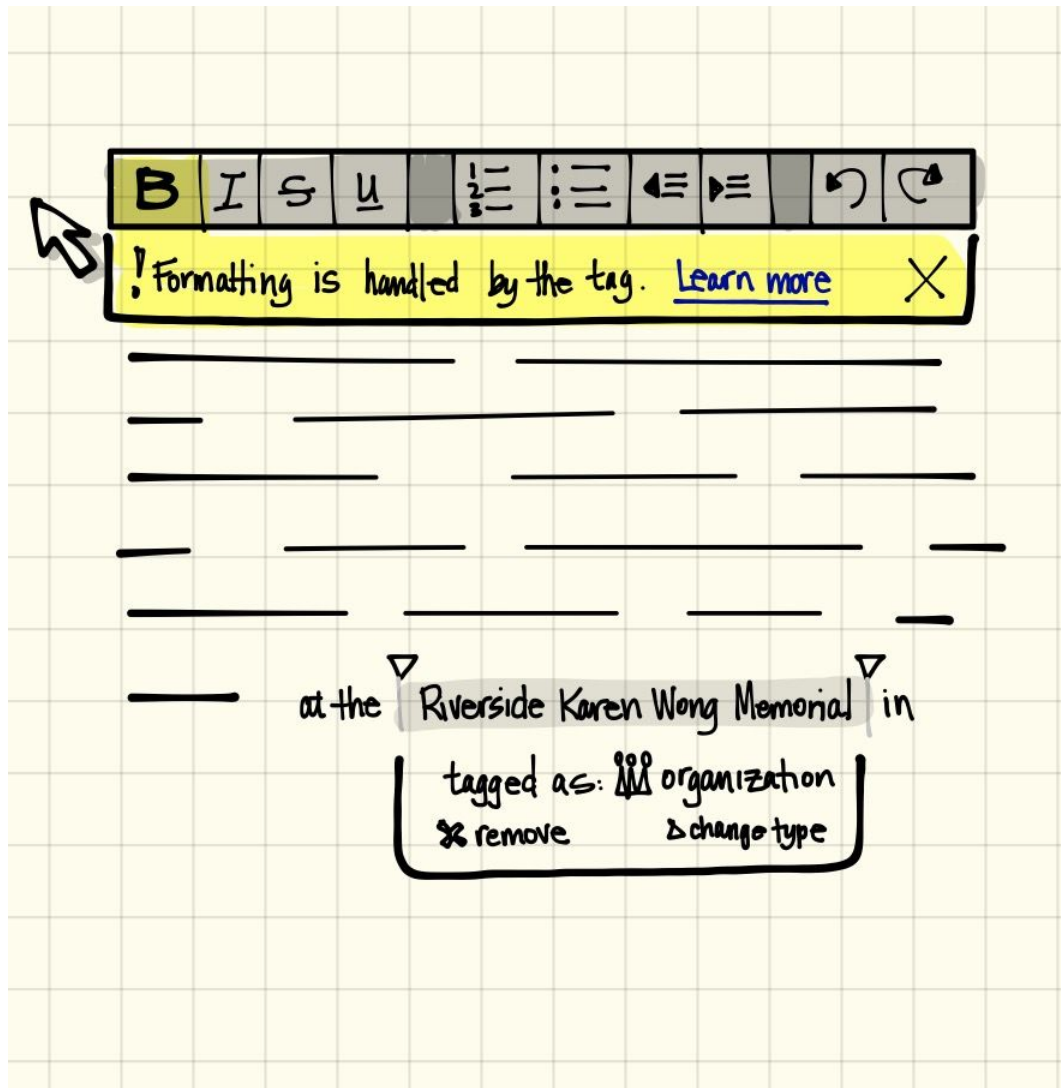
Formatting vs. Semantic Markup

One of the questions I had was the conflict between formatting markup (e.g. Bold, italic) vs. semantic markup (e.g. Location, organization name). It is very much in everyone's interest to have things marked up semantically (and have it automatically formatted for that type of entity) rather than formatted. Semantic markup gives more confident search and analysis capabilities. So the system should help Gayle err on the side of semantic markup when it can.

The first way Madison helps her is by disabling/dimming formatting tools when semantically tagged text is selected. See the image below. This gives a visual cue that semantic tags have priority.

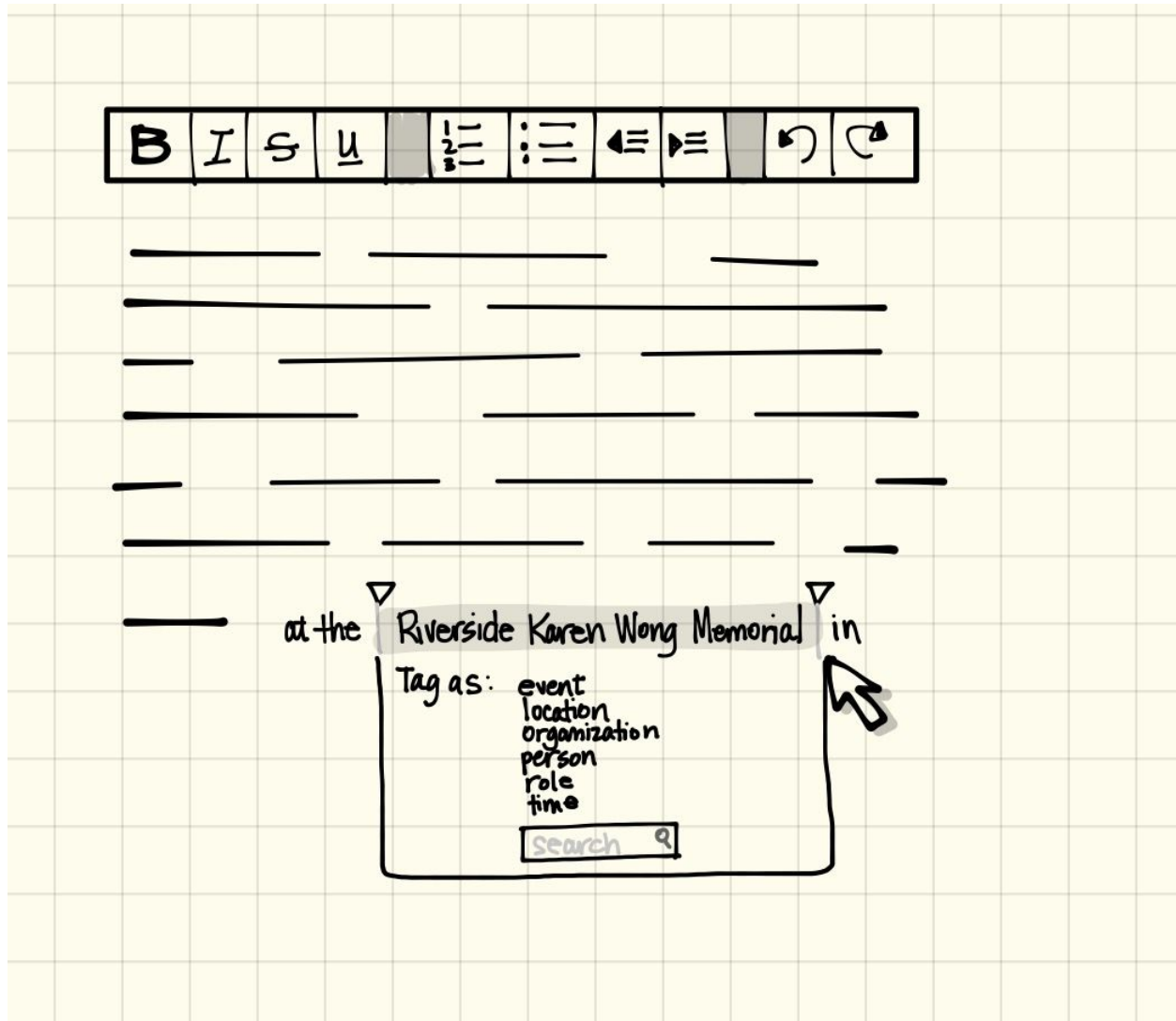


If Gayle has a tag selected and clicks a formatting tool or presses the keyboard command anyway, e.g. CMND+B for bold, the interface provides a small audible buzz and a visual explanation. This notification stays visible for 10 seconds before disappearing on its own, but she can dismiss it at any time.



Another way that we encourage semantic markup first is by pre-tagging entities Madison identifies. This should take care of 90% of the taggable content. It would require her to dismiss the semantic tag to add the formatting tag, and that extra work should dissuade the habit.

The last way we help her is with the placement of the semantic markup tools. Being very near her cursor, it will get her visual attention more than the formatting bar when she makes a selection. Compare that to the formatting tools, which are in a bar high up on the page.



Structure tags

We have not yet discussed a mechanism for selecting, removing, or changing structure tags for a given text. To some extent, the logic of structures will help constrain Gayle. She cannot add a title to the middle of a document, and must include a title at the top. But I can see where some middle, optional structures are hard to confidently recognize and might be one of several types. (Is this correct?) I suspect this this needs to be an in place menu structure similar to the element tags, but let's punt this to next week to enable the test to progress.

Keyboard control for semantic tags

GOMS studies tell us that switching input modes (mouse-to-keyboard or keyboard-to-mouse) has the biggest impact to microinteractions efficiency. To that end, the software should allow Gayle to navigate the semantic tagging menu with keys when it is visible. Access through the

arrow keys for selection and RETURN key for activation should be sufficient. No chorded commands are necessary.

Remaining topics for the document page

In a call with Bill, we discussed a next issue may be groups/permissions and workflows

- People can belong to groups, e.g. A legislative committee, but it is not required
- Groups are primarily useful for permissioning (admin / invite / approve members. edit / comment / view Documents)
- If users belong to a group, 99.9% of the time users only belong to one
- OpenGov admins will belong to multiple

We discussed the question of whether Gayle would want to flag some legislation as important, and since the majority of users are only working on one document at a time, this seemed less necessary for near term development.

Other topics to pursue

- What does the system do with suggested but unconfirmed semantic tags?
- Workflows of selecting a template (which entails an existing-documents menu and controls)
- Workflows of passing a document to other individuals or committees for review. These always have a deadline, indicating a window when responses will be accepted.
- Search, finding, and ordering documents by ownership and (very important) deadline
- Workflows of applying a template and/or applying tags after beginning a free-form document
- Workflows of changing a template

These will be marching orders for the later part of today and next week.

Sprint 1 Week 3

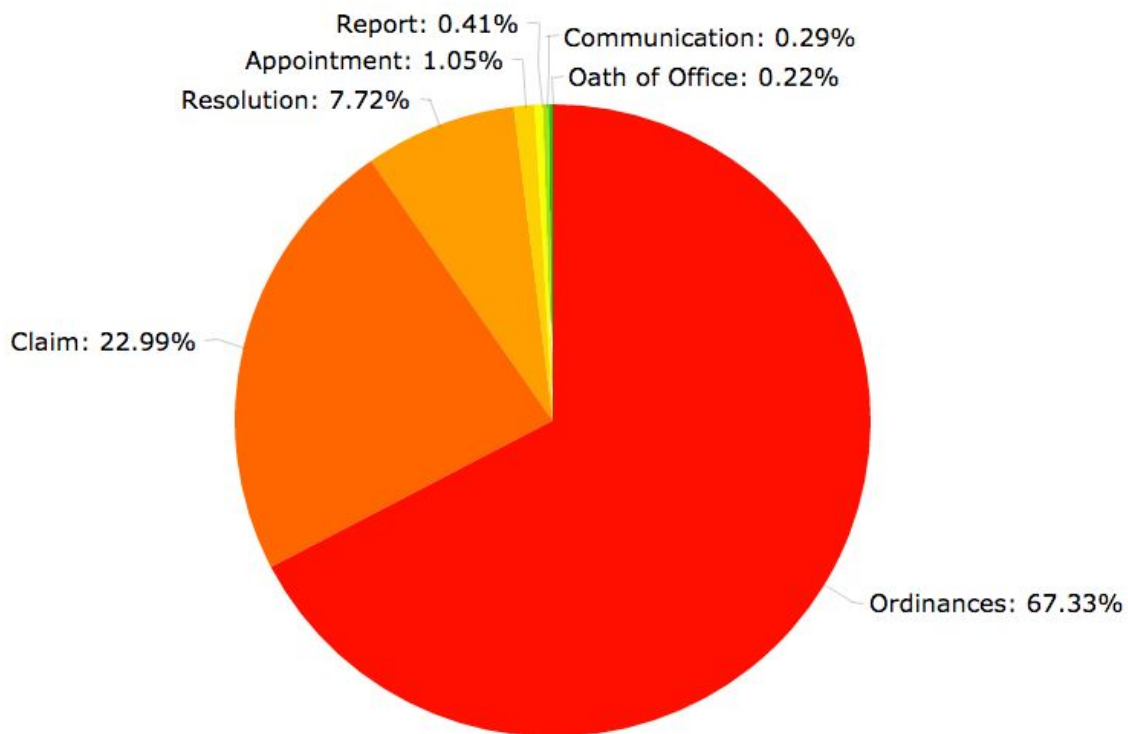
(30th Memorial Day) 31 May–03 Jun

Since the prior weeks we looked at two different methods of interaction: highly constrained and free-form, this week we're going to be looking at

- Workflows of selecting a template
- Workflows of applying a template and/or applying tags after beginning a free-form document
- Workflows of breaking from a highly-constrained form to free-form
- The breakdown of Gayle's time: How many highly-constrained and how many free-form
- Completing some of the outstanding design items from the prior sprints, namely adding in smart elements and tagging items as elements after entry.
- Shifting views between current state of the law and a timeline of events.

We recognize the need for a provisional citizen persona, and are using "Barry" as a name placeholder for now. E.g. "Barry can roll over the defined word "temporary" and see "45–90 days within..."

When we look at the legislation types on the councilmatic website, the overwhelming majority of legislation is ordinances. While most of these can be form-based (and driven in form templates similar to the New Parking Ordinance wireframe below) many still need **free-form descriptions** as described in the Resolution wireframe below. And it's not clear that Gayle will know whether exactly what template she might need at the beginning will be, and she might guess incorrectly and need to switch templates mid-task, which tells us that we need to complete Gayle's initial scenario.

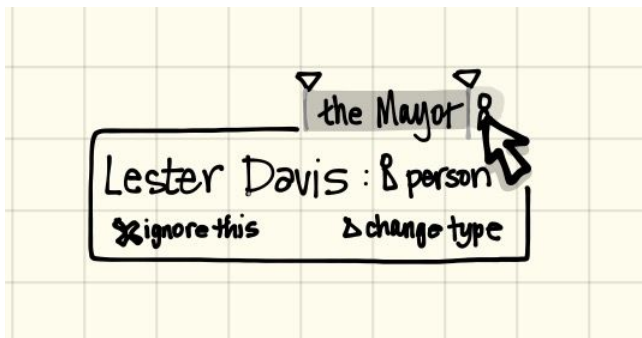


Below is Gayle's original scenario. It is copypasted here so we can iterate it.

Gayle has received a new resolution. It is a copy of an older, similar resolution with few handwritten changes. She opens Madison to transcribe the resolution and ___(whatever is next in her workflow, we don't know)___.

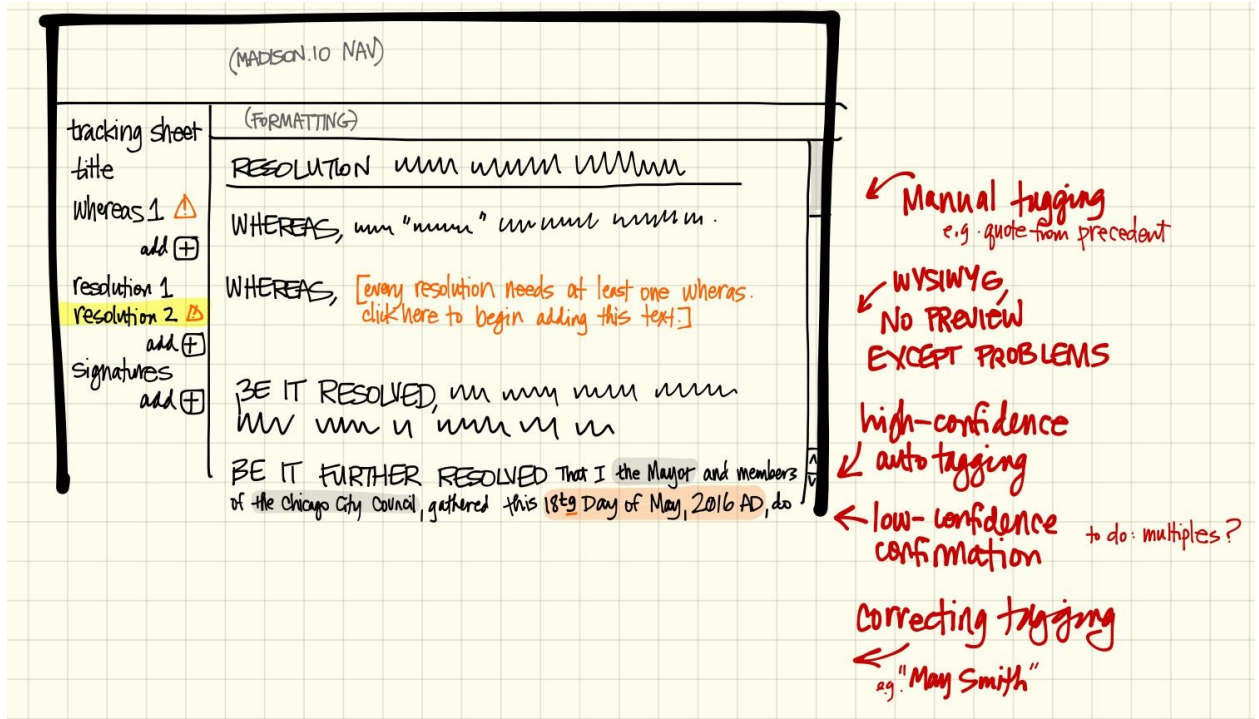
- Gayle selects "new resolution" within Madison.
- She sees a blank resolution, with formatting tools at the top and the outline of the document on the left. [The "structures" and "tools," in Madison lingo.]
- She sees "title" highlighted on the left and the word "RESOLUTION" prepopulated with the correct formatting for a title. Her cursor sits to the right of "RESOLUTION", prompting her to type the title. She types the name of the resolution and hits return.
- She sees "WHEREAS" appear in the line below with her cursor after it, and she transcribes the first WHEREAS per the handwritten changes indicated on the paper. [Note: This WHEREAS will have a quote from precedent law and she will need to manually tag it, but we haven't completed this yet.]
- She hits RETURN and sees that nothing is prepopulated on the next line (It could be another WHEREAS or the first resolution.) She types "W" and the word WHEREAS auto-completes for her. She sees a second WHEREAS appear in the outline to the left. [Note: If her attention was on the paper rather than the screen and continues to type "...HEREAS", Madison is smart enough to disregard it and not produce "WHEREAS, HEREAS"]

- She knows there will be a second WHEREAS clause but doesn't have the text for it yet, so hits RETURN again. She sees a soft prompt reading, "every WHEREAS clause requires some text. Be sure to complete this before submitting." She also sees a small alert in the outline to the left next to this WHEREAS in the outline. She disregards these prompts, knowing she'll have to come back to this clause later in the week.
- She types "B" and the "BE IT RESOLVED" auto-completes for her. She types the first resolution text and hits RETURN to begin the second resolution.
- She types "B" for the auto-completion and transcribes the entire second resolution text from the paper. This second resolution includes references to "the Mayor" and the "Fairview City Council." She sees that these terms have been highlighted as elements with small icons that indicate their type, and that "the Mayor" highlight includes confirmation text of "Lester Davis," so she knows the autotagging did its thing correctly.
[This implies a lot of smarts around offices and time, let us know if that's too much work.]



- **CORRECTING TYPOS**

She also sees that she mistyped "18tg Day of May 2016 AD," as it has a different color highlight, a question mark icon, and a prompt for action. "Resolution 2" in the outline also indicates that an error exists in this structure. (see drawing)



She clicks the misspelled date and sees a suggestion for the correctly spelled date. She confirms the change with a click, and sees its highlight change to match the others. [It would be great if we could avoid her mode-switching to the mouse and back, but this would require some thinking of the mode-keys (Word uses all of them now) and some training, so will leave for later.]



● CHANGING TAGS

The third resolution is similar to one from another document. She opens it, selects the resolution text, copies it, and returns to this document. She inserts her cursor at the end and pastes the text. Madison recognizes that the paragraph begins with "BE IT FURTHER RESOLVED," and automatically tags it in the semantic pane as "resolution 3." Madison also parses the text.

- Gayle sees that "May" is highlighted as a date with low confidence, and "Smith" is highlighted as a person. She knows the algorithm has misunderstood the text. She clicks the calendar icon next to "May." She sees a familiar popup asking her to confirm that this is the

upcoming May in 2017. She also sees controls to *remove* or *change* the tag to another type.

- She selects *change* and sees a short list of common types as well as a search box for uncommon types. She sees “Person” in the list and selects it. Gayle sees a prompt reading “Tag ‘May Smith’ as a person?” She accepts, and sees that May Smith now has a highlight and icon indicating it has been tagged correctly.

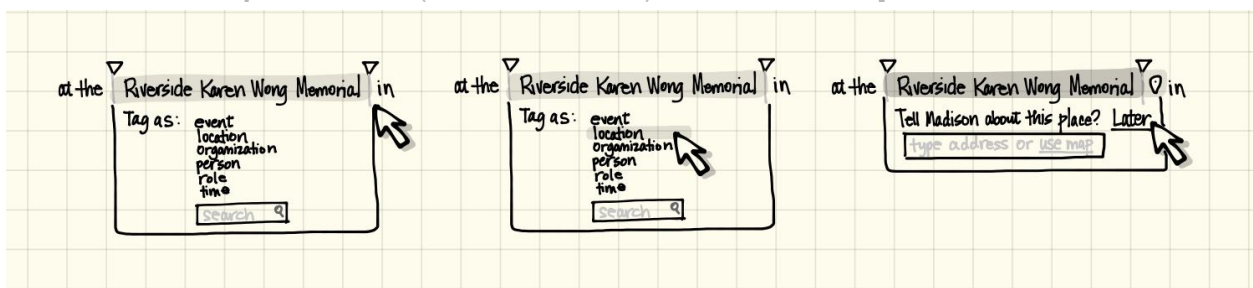
[This presumes around a dozen common elements and up to hundreds of less common element types]



- **MANUALLY ADDING A TAG**

She looks over the text and sees that “Riverside Karen Wong Memorial” does not have a highlight, and it should. She uses her mouse to select the text (it smartly snaps to the ends of words). She sees a control to add a tag to the selection. She selects “place” from the list, and sees a highlight and place icon confirming the correct tag. She also reads that she can “Tell Madison about this place,” but she’s got to get to lunch, so she keeps typing and sees the prompt disappear.

[Note that we have presumed that a simple “place” tag is too vague to be of use. We could be wrong in this. Also, Madison could require her to pause her work and tell it about the place, e.g. physical boundary, other names, etc. But we don’t want to slow Gayle down. So we give her the option and if she declines, Madison records it as an item for “**cleanup**” which she (or someone else) can handle later.]



- On the new line she types “S” and sees the signature element appear, correctly naming Davis as the signatory.

- **AUTOSAVED**

She looks over the text and feels confident that she’s done everything she can until she

gets the new WHEREAS clause. She sees at the top of the screen that the document has been auto-saved as of 1 second ago, so closes the document for the day and heads to her email to remind her supervisor that she needs the WHEREAS clause by the end of the week. [The Madison document list will also know whether documents have errors and can show that in file lists.]

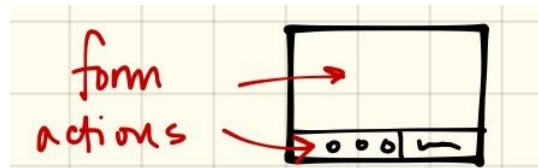
Sprint 1 Week 2

23–27 May 2016

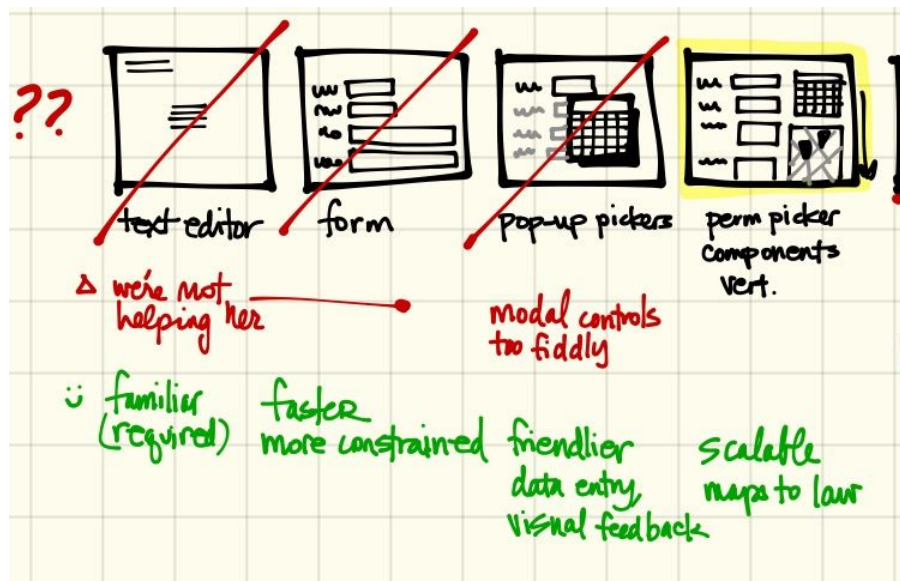
Focusing on the differences of **ordinances**

- Ordinances are much more **data-driven** than **language-driven**, e.g. parking ordinance can be type, date start, date end, start address, end address
- When open-ended language is *not* needed, working in language is error prone
- Semantics can be embodied in form structures
- Opportunity to provide visual input and verifications (fewer errors, don't know if that's ever a problem). I'll call these "pickers" though that sounds dopey.

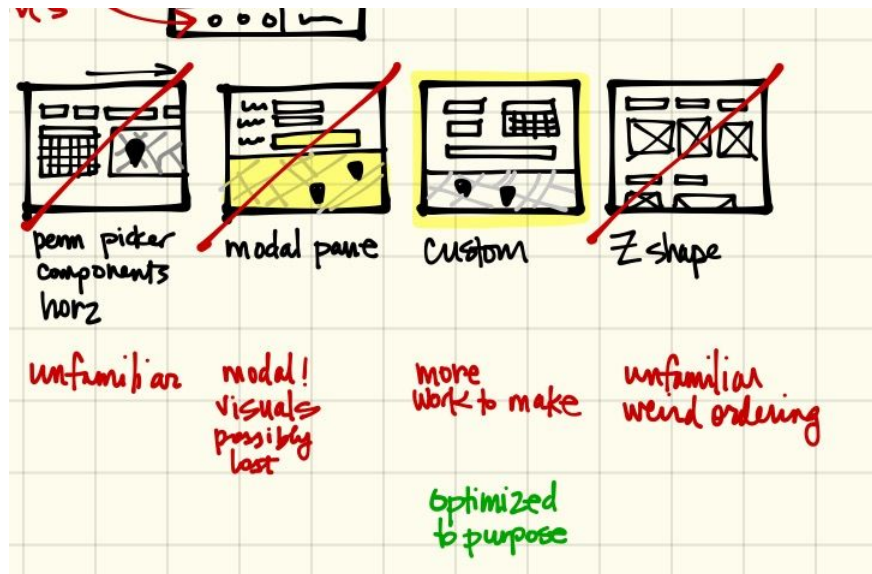
Basics are a form followed by an action bar.



Looking at broad patterns for what the **form** could be...

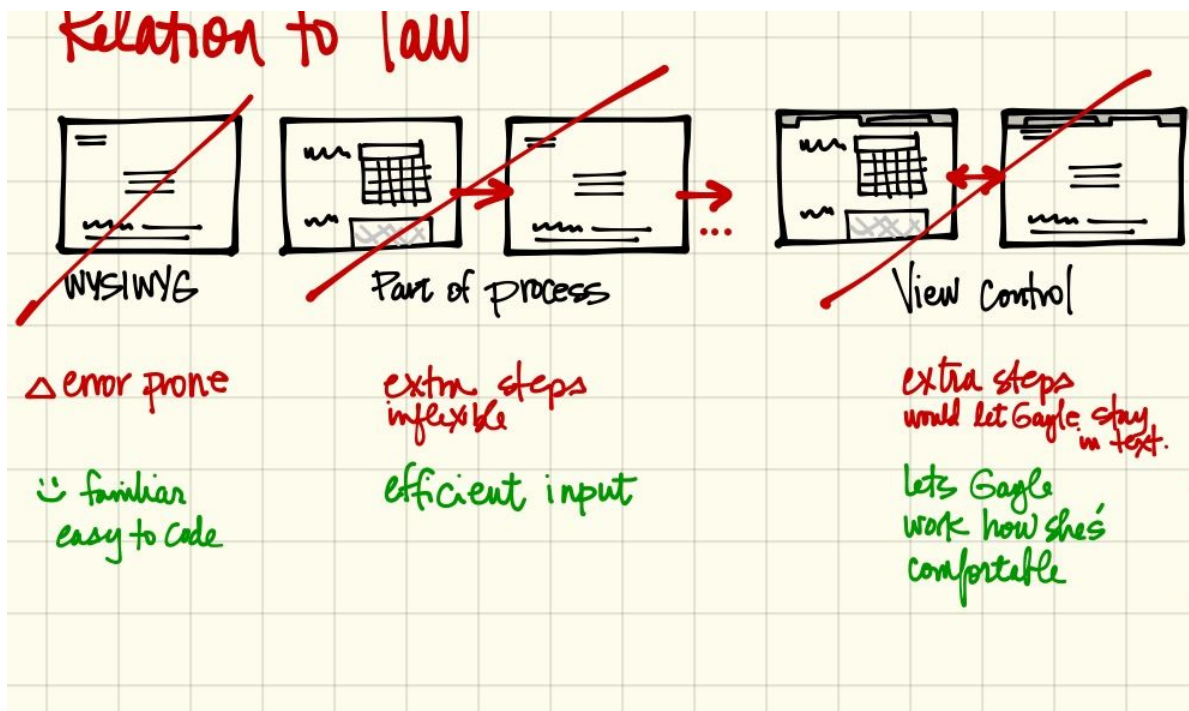


- Having modal pickers (like pop-ups) feels too fiddly, too much management of the UI, plus doesn't serve the purpose of a visual reinforcement of the data. (This nixes a modal window, too.)
- If we keep the pickers persistent, then they need to line up with the form inputs, vertically or horizontally. Horizontally might match a tabular layout of similar ordinances, but it scales in unfamiliar ways, either requiring horizontal scrolling (no one likes) or Z-shape layouts, which are awkward in their own right. So vertical makes the most sense.



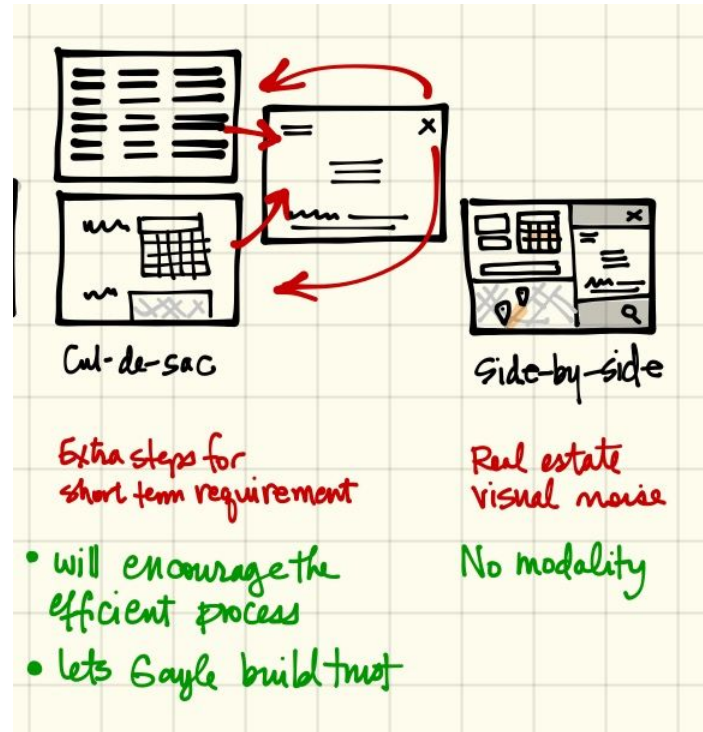
- You could simply **vertically stack** the form elements and pickers on top of each other. This is easy to build and scale, but isn't optimized to the task.
- **Custom layout per ordinance type** is more work to build and manage, but makes for a better experience, so I'm lobbying for that. (I can be talked down from this.)

We can't lose the legal formatted document, so the next pass is at how the input/edit form and that legally formatted text relate to each other. First, what doesn't seem to work...



- Again, not considering WYSIWYG text editing since that's error prone for ordinances.

- We could add a step after filling out the data, where she reviews auto-formatted text and either modifies it or, more likely, just submits it. This is troublesome because if she modifies the text directly, how does that affect the form if she opens it up later? It's also an extra step in a process that may happen many times a day.
- We could provide a view control, allowing her to work in either text mode or form mode, letting her work in either. But, my guess is for familiarity sake she would stick with text, and so we've ultimately not helped her.



- A **"cul-de-sac"** mode provides a control from the list of ordinances or form to open an auto-formatted preview of the legal language. It's not editable, but does show Gayle what it will look like when she submits. This extra step encourages her to work in the data and let Madison handle the formatting.
- A **side-by-side** mode lets Gayle open up a formatted page on the same screen and watch it update as she modifies the form. You *could* have a back-and-forth editor where she could just go over and change the text directly and have those updates propagate to the form, but that requires a lot of natural language processing and is pretty fraught for the benefit. Better is to have it static, I think, and closeable. That way she can lose the real estate (the form would need to be responsive) as long as she wants to work side-by-side, and swap to a pure form mode when she's built enough trust that the formatting is being handled correctly.

I like that last one, so drew it up as a more detailed image below so we can discuss.

(MADISON.IO NAV)

NEW PARKING ZONE ORDINANCE

DATA ENTRY

with Visual pickers/confirmation

actions

Preview

auto formatted preview

Cancel Submit & Close SUBMIT FOR SIG & START NEW ORDINANCE

I have no idea what the real actions might be at the end of the task, so take the lower bar to be for placement only.

Note that the rich visualizations of the map locations are derived from the input forms, and the input forms must allow free-form text. The form does its best to resolve it to an actual address, perhaps even offering suggestions, but if Gayle needs to enter, “The large pine tree at the northwest corner of John Smith’s farm,” she should not be prevented. In that case the visualization grays out, explaining that it can’t resolve the address to a map. We might give her a tool to select a spot on a map, but that will be extra work that doesn’t help her do her job, which is the text on the right.

Sprint 1: Week 1 Semantic Markup

16–27 May 2016

Task

Produce designs that allow users to tag a given text semantically, for **structure** and **elements**, as described [here](#), without interfering with basic markup tools.

Use cases

- **Keying in a text and tagging it for structure & elements**
 - Common, used more by clerks while migrating laws to Madison
 - Rarer later and used more by legislative Directors, counsel, or lobbyists once Madison is in use
- New law based on old law: Modifying a duplicated text that already has structure & element tags (common once Madison is in use)
- Reviewing someone else's text, structure, & elements for accuracy
- [Note sure: Suggesting an amendment to someone else's bill?]

Persona: Gayle

We are using Gayle as a provisional persona. We know very little about Gayle at this point, other than she is a legal clerk in a town called Fairview who works in the tool a lot and has a high throughput of legal text to process each day.

We only have the vaguest confidence in who Gayle is, what her day is like, and what her needs and frustrations are. We still hope to have some in person conversations to ground this in some sort of reality.

Scenarios

We presume the overwhelming bulk of work to be versioning to comport with new policy: collecting comments/ modifying small bits of language, rather than, say, writing a new law from scratch. Both must be accommodated, but it should be optimized for the small tweaks.

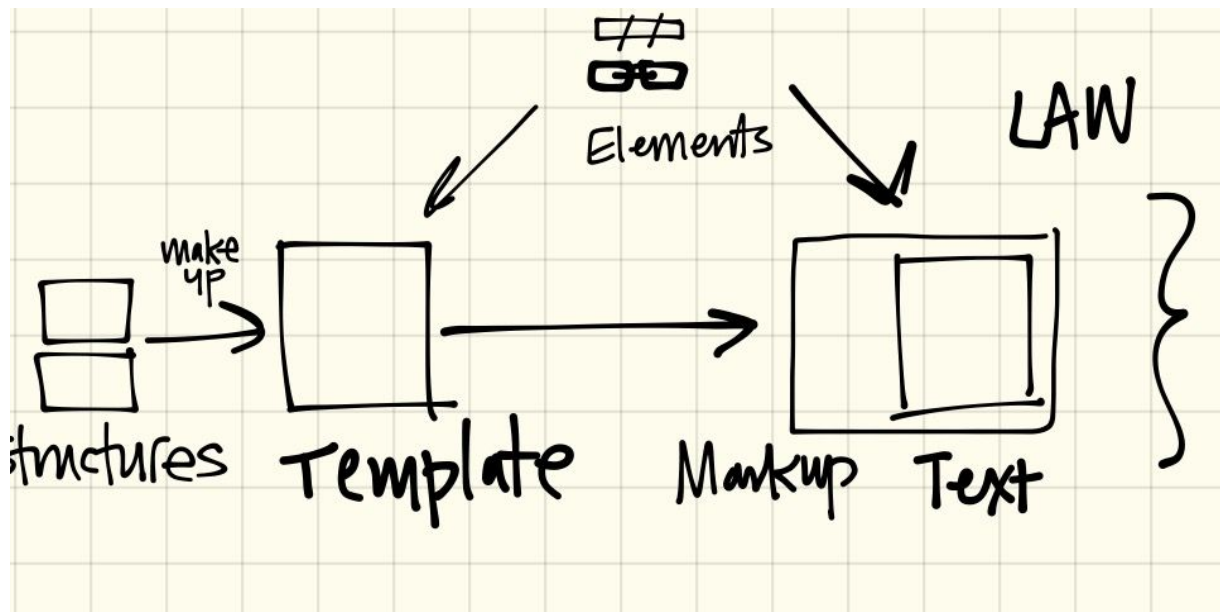
More research would help us understand the scenarios and workflows to consider and test against.

Outstanding questions

- Q: Is the primary scenario keying in text from scratch, duplicating and modifying, or tweaking in process?
- A: <https://www.govtrack.us/blog/2010/03/24/who-writes-our-law/>
- A: https://www.reddit.com/r/answers/comments/2l6wr4/where_are_the_laws_written/
- Q: Will a document ever have its template switched (e.g. from ordinance to statutory)?
- Q: How frequently are templates (and structures) changed (e.g. iterated or improved)?

The Objects

The [Document Properties](#) contains lists of elements. We are thinking of them in the following way.



The user works in text with semantic markup. The semantics are governed by the document type, its status, and the template's rules, which define the structures that must be present in the text, and their order, and what they must and can contain. Certain smart objects like dates or authors may be included in the text and may be required by the template.

Highlights indicate our thoughts rather than something we've read so far.

Template

- Document type e.g. Resolution, ordinance, claim, etc.
- Made up of structures e.g. statement, preamble, recital, etc.

- Rules (there can be one or more whereas clauses, a date is required, etc.) (Chris: Is this like a regular expression?)

Metadata

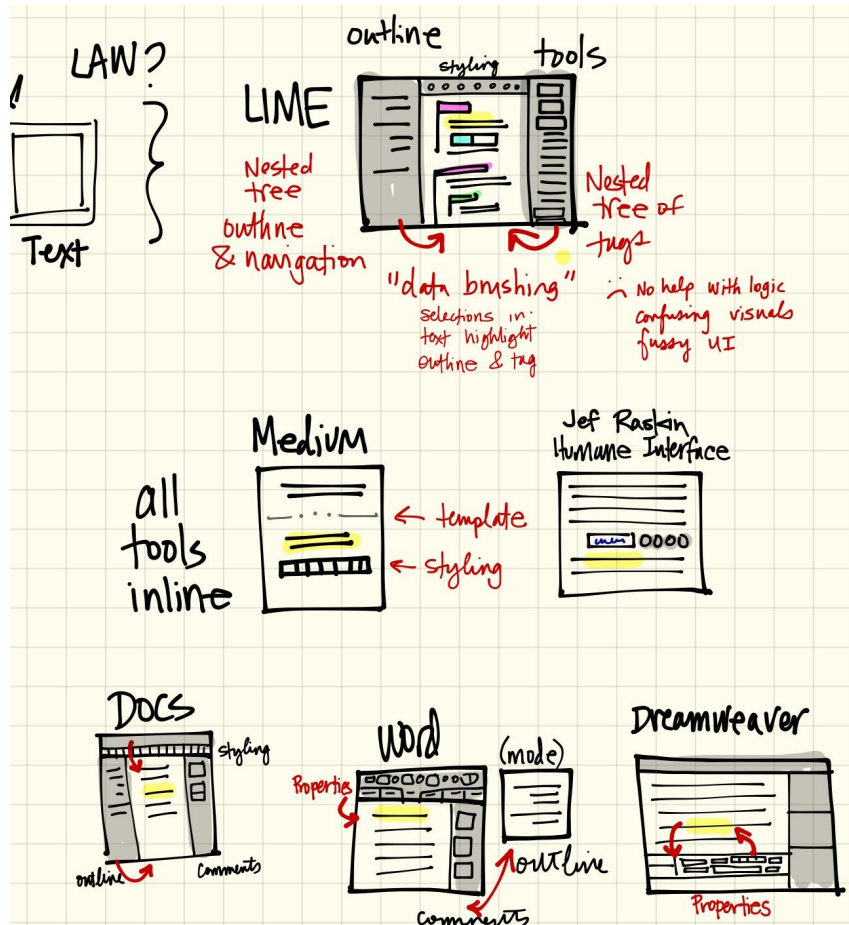
- Status (proposal, in discussion, law, superseded, etc.)
- Package (group of bills)
- Related to... law
- Sponsors
- Legislator comments
- Citizen comments
- Related research/resources
- Previous versions
- Authors
- Copied from...law
- Precedent
- Judicial rulings / opinions

Derived Data

- Relevant news
- Similar legislation
- Cases
- Citizen and legislator followers

Some existing models

We took a look at the frameworks for a few apps to compare them.



- LIME: Very busy presentation, with overwhelming visual complexity, number of tools, etc.
- Google Docs: Clean but constrained for having just styling controls.
- Microsoft Word: Heavy and modal ribbon for all tools. Separate, modal outline view.
- WordPerfect (was not yet reviewed for lack of access or familiarity)

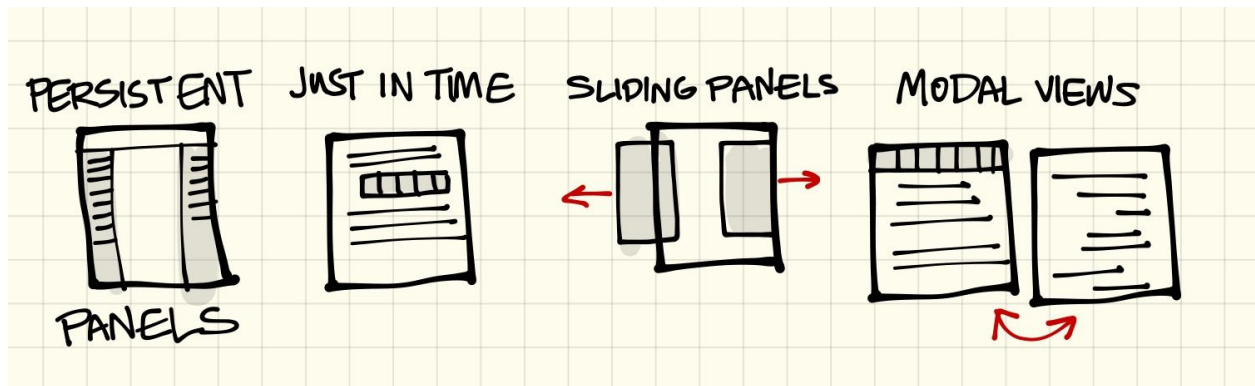
We also looked at some tools that are popular and involve markup but that are less general-purpose. More could be considered with more time.

- [Dreamweaver](#) (We know, but we had it on hand) Very technical, exposing a lot of technical detail
- [Medium](#): Very Spartan, focused on elegance
- Jef Raskin's [Humane Interface](#), focused on efficiency.

Generally, the key task embodied in the frameworks is to input text, select bits of that text, and apply attributes to them (either styling or semantics). Each of these systems have adhered to one of four broad interaction models for presenting the key components.

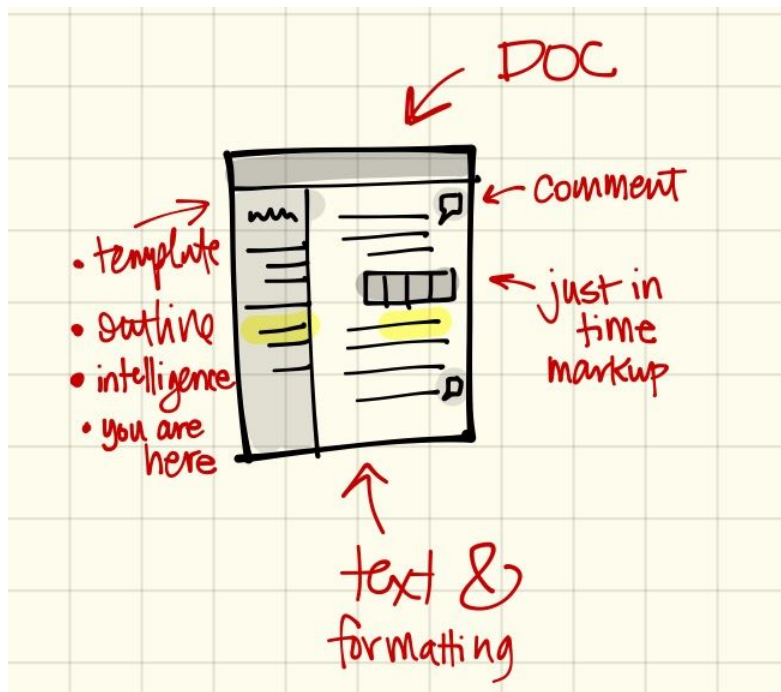
- Outline (and, implied, the structure)
- Text,

- Markup tools.



1. **Persistent panels**, in which the panels take up real estate and are persistently available. User selects an outline to jump there. User selects text, navigates to tag and clicks to apply.
2. **Just in time tools**, in which tools are invisible until text is selected, and context-appropriate tools are floated near the selection. Select-navigate-click model.
3. **Sliding panels**, in which the panels act as drawers that can be summoned or dismissed as needed. Select-navigate-click model.
4. **Modal views**, in which the outline is a modal view to which the user switches. Modes reduce the number of tools needed at any one time, as the tools only apply to the current mode.

Suggested direction



We recommend going with a hybrid framework for Gayle.

- A **document** panel at the top that contains the metadata (like status) and action buttons (like “submit to _____ for review”)
- A persistent **outline** on the left that tells her
 - What sort of **template & structures** she’s working with
 - Any **system intelligence**, e.g. this structure is not yet complete
 - **Wayfinding** features that let her know where in the document she is
- A body of **text** that is a styled display of the law
 - **Just-in-time tools** near her locus of attention
- A text-formatting toolbar along the top, similar to Madison now.
- (Though it’s not key to this particular sprint) Comments in a standard position to the right.

Benefits

- Gayle doesn’t have to fiddle with the UI too much. She wants to stay focused on content.
- The text looks like what she’s used to seeing *until she interacts with it*, so the UI complexity is revealed progressively. That should be more comfortable.
- We should be able to spare her from switching modes too frequently (more below)

Risks

- This puts a lot of **functional weight on the just-in-time tools**. What if there are hundreds of things that need to be accessed there? Can it fit in that spot? We note that

the same problem will exist even if these tools are in a panel along the side or top, but will be more constrained for space. Between search tools and most-likely-guesses to present a small set of controls with access to the entire set, we *think* we can deal with this, but that will be played out in the next step.

- We also want to avoid having Gayle **switch input modes frequently** (e.g. keyboard to mouse) as this is the most annoying and costly thing to do in a frequently used interface, so even as we solve the just-in-time tools, we'll want to keep her hands in the same place.

Questions

- Can we have a WYSIWYG model rather than markup and preview?
- The existing demo has only weak template-level logic, e.g. errors like “no Context” and whereas. Can it have higher-level logic, e.g. “Whereases must be at the beginning, Resolveds must follow after. There can be only one title”?

More detail as a scenario: **Gayle transcribes a new resolution honoring May Smith.**

Gayle has received a new resolution. It is a copy of an older, similar resolution with few handwritten changes. She opens Madison to transcribe the resolution and _____ (whatever is next in her workflow we don't know) _____.

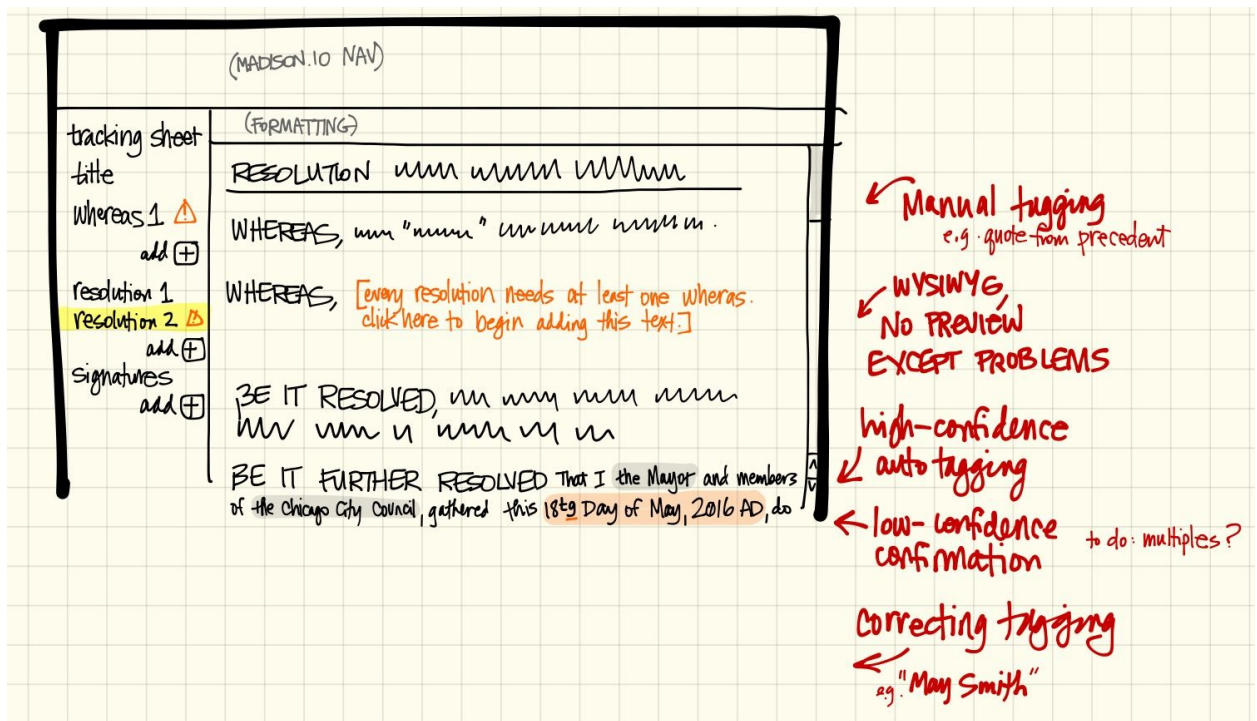
- Gayle selects “new resolution” within Madison.
- She sees a blank resolution, with formatting tools at the top and the outline of the document on the left. [The “structures” and “tools,” in Madison lingo.]
- She sees “title” highlighted on the left and the word “RESOLUTION” prepopulated with the correct formatting for a title. Her cursor sits to the right of “RESOLUTION”, prompting her to type the title. She types the name of the resolution and hits return.
- She sees “WHEREAS” appear in the line below with her cursor after it, and she transcribes the first WHEREAS per the handwritten changes indicated on the paper. [Note: This WHEREAS will have a quote from precedent law and she will need to manually tag it, but we haven't completed this yet.]
- She hits RETURN and sees that nothing is prepopulated on the next line (It could be another WHEREAS or the first resolution.) She types “W” and the word WHEREAS auto-completes for her. She sees a second WHEREAS appear in the outline to the left. [Note: If her attention was on the paper rather than the screen and continues to type “...HEREAS”, Madison is smart enough to disregard it and not produce “WHEREAS, HEREAS”]
- She knows there will be a second WHEREAS clause but doesn't have the text for it yet, so hits RETURN again. She sees a soft prompt reading, “every WHEREAS clause requires some text. Be sure to complete this before submitting.” She also sees a small

alert in the outline to the left next to this WHEREAS in the outline. She disregards these prompts, knowing she'll have to come back to this clause later in the week.

- She types "B" and the "BE IT RESOLVED" auto-completes for her. She types the first resolution text and hits RETURN to begin the second resolution.
- She types "B" for the auto-completion and transcribes the entire second resolution text from the paper. This second resolution includes references to "the Mayor" and the "Fairview City Council." She sees that these terms have been highlighted as elements, and that "the Mayor" highlight includes confirmation text of "Lester Davis," so she knows the autotagging did its thing correctly.

[This implies a lot of smarts around offices and time, let us know if that's too much work.]

- She also sees that she mistyped "18tg Day of May 2016 AD," as it has a different color highlight and a prompt for action. "Resolution 2" in the outline also indicates that an error exists in this structure. (see drawing)



She clicks the misspelled date and sees a suggestion for the correctly spelled date. She confirms the change with a click, and sees its highlight change to match the others.

[It would be great if we could avoid her mode-switching to the mouse and back, but this would require some thinking of the mode-keys (Word uses all of them now) and some training, so will leave for later.]

- [To do: Describe the third resolution as a paste of a block of text to illustrate dealing with multiple tags, as well as the name "May Smith" which she'll need to correct.] She hits return.

- On the new line she types “S” and sees the signature element appear, correctly naming Davis as the signatory. [This and subsequent steps will be numbered after we complete the third resolution, which will likely be many steps.]
- She looks over the text and feels confident that she’s done everything she can until she gets the new WHEREAS clause. She sees at the top of the screen that the document has been auto-saved as of 1 second ago, so closes the document for the day and heads to her email to remind her supervisor that she needs the WHEREAS clause by the end of the week. [The Madison document list will also know whether documents have errors and can show that in file lists.]

What did we just see and why is this awesome for Gayle

- She doesn’t have to think about “tags” “semantics” or any of that technical stuff. For the most part, her focus stays on the content like she always has. Having selected “resolution” we have enough to help her with everything else.
- Madison helps Gayle through some common complications.
- Gayle only has to take her hand from the keyboard to the mouse very occasionally, keeping her fast and efficient.
- The WYSIWYGs lets her work in a familiar format and not go through a “preview” mode to see what the “real thing” would be.
- High-confidence autotagging just happens.
- Low-confidence prompts draws her attention and helps her resolve errors.
- Just-in-time tools help her manually tag, keeping the options constrained by the template and structure, so she only has to deal with relevant tags at any one time.

Next tasks

To write:

- Designing the third resolution, with its block of pasted text from another document and resolving multiple low-confidence tags.
- Adding a tag with the just-in-time markup tool.
- Correcting a tag when she types in “May Smith” and Madison thinks “May” is a date.
- Drafting a next scenario where Gayle returns to the document, to add the second WHEREAS clause, and forward the doc to Lester for his signature.
- To answer: Can we help Gayle if she uses *style* formatting when she should be marking something up as an *element*? E.g. She **bolds** a book title instead of tagging it as a `<book>book</book>`.
- Write up a best-case scenario without the problems Gayle has to illustrate how it might work in most cases.

To draw:

- Adding more state-by-state illustrations to the text for communication purposes (Sketch? Would like OpenGov approval before committing this time)

To consider:

- A more fill-in-the blank scenario, ORDINANCE, e.g. A parking permit

