Quantifying and Reducing the Cost of Web Edits

Edward Benson, MIT CSAIL
eob@csail.mit.edu

Outgoing Links

treesheets.org contact info

1. Problem: HTML complexity is out of control!

Web authoring is an exercise in copy-paste-and tweak

- Find bits of design you like.
- Copy and tweak them.
- Edit to customize content.
- Later, update content.

But HTML conflates design scaffolding with content.

```
<div class="alert alert-error">
  <button type=button"
          class="close"
          data-dismiss="alert">x</button>
  <strong>Error</strong>
  <span>Something went wrong.</span>
</div>
```

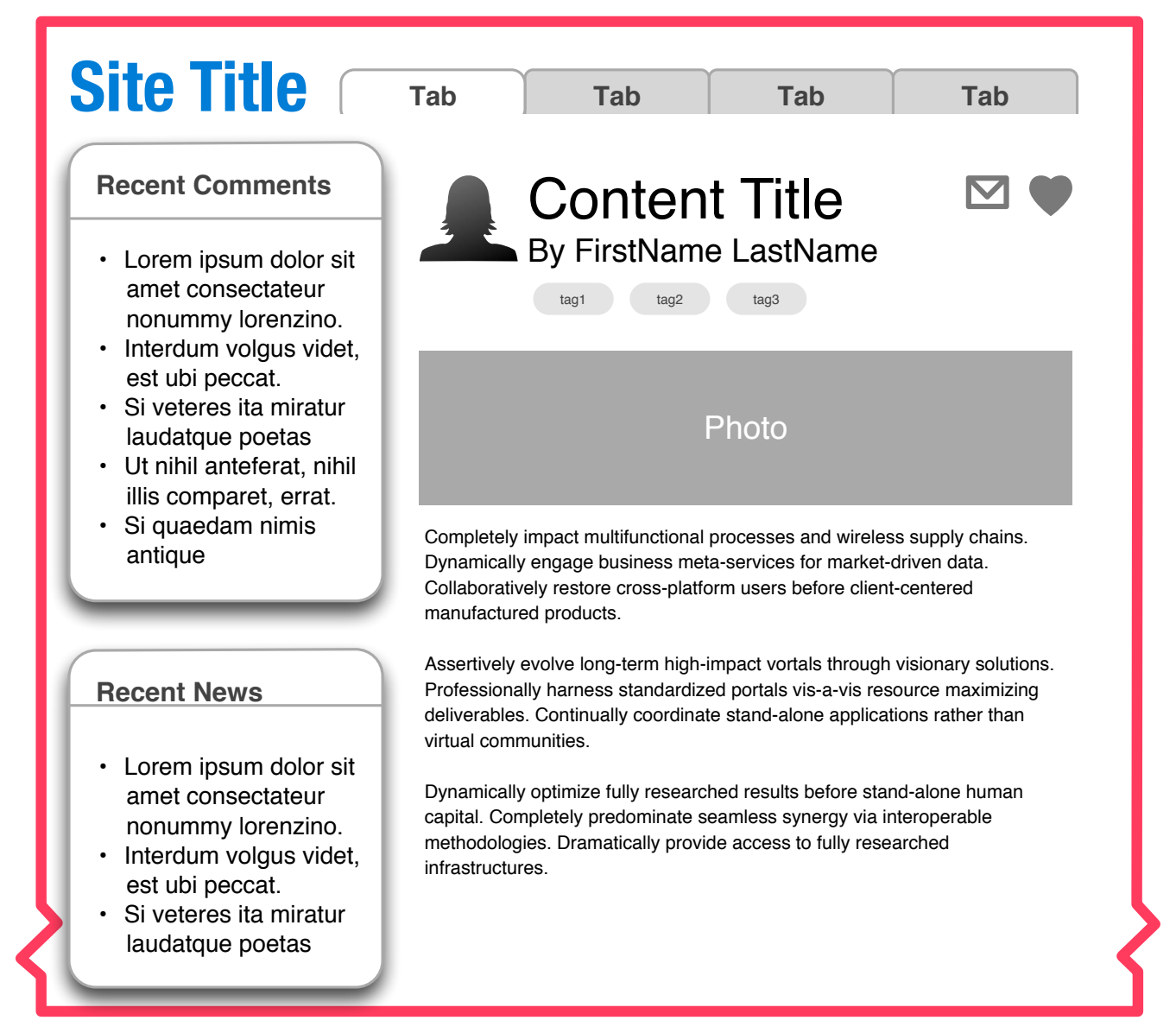
An error button using Twitter Bootstrap. One piece of user content ("Something went wrong.") is wrapped in 4 elements and 4 attributes.

Modern design complexity makes artifacts hard to reuse

- Copy-paste-tweak is increasingly difficult.
- WYSIWYG and CMSs are great, but they mask the deeper problem.
- HTML maintainability impacts the web, apps, eBooks, etc.

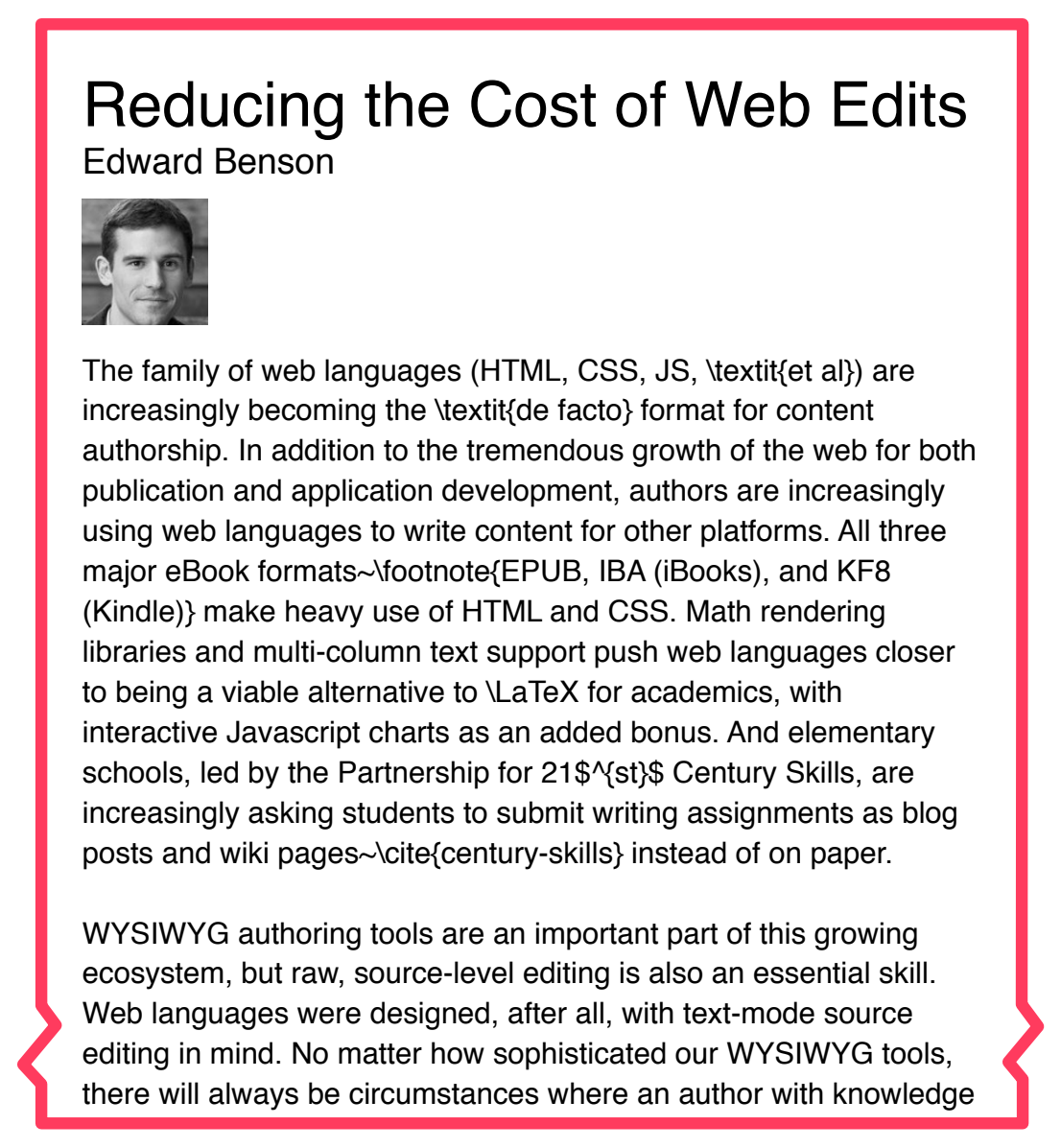
2. Solution: Separate mockup + content documents

Design Mockup



- Ordinary HTML Page (not a template).
- Can be hosted and linked to by third others.
- Every piece of overridable content is placed in a distinct DOM element, addressable by a CSS selector or classname
- Blocks of structure, such as a sidebar widget, also addressable by CSS classname.

Content Document



Reducing the Cost of Web Edits
Edward Benson

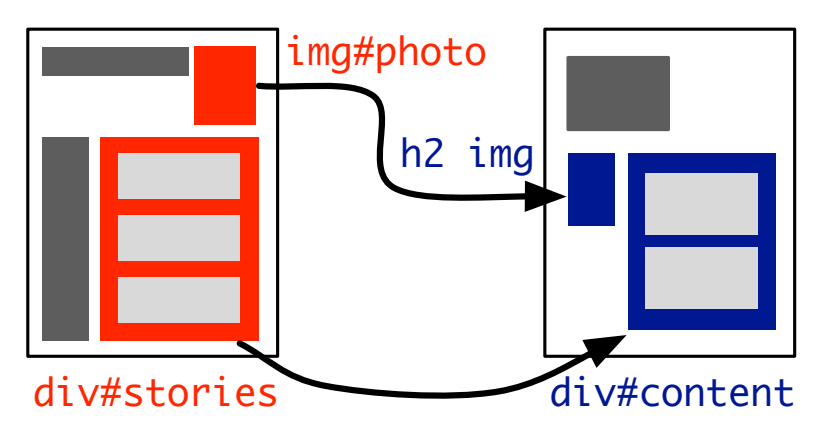
The family of web languages (HTML, CSS, JS, \textit{et al}) are increasingly becoming the \textit{de facto} format for content authorship. In addition to the tremendous growth of the web for both publication and application development, authors are increasingly using web languages to write content for other platforms. All three major eBook formats~\footnote{EPUB, IBA (iBooks), and KF8 (Kindle)} make heavy use of HTML and CSS. Math rendering libraries and multi-column text support push web languages closer to being a viable alternative to \LaTeX for academics, with interactive Javascript charts as an added bonus. And elementary schools, led by the Partnership for 21st Century Skills, are increasingly asking students to submit writing assignments as blog posts and wiki pages~\cite{century-skills} instead of on paper.

WYSIWYG authoring tools are an important part of this growing ecosystem, but raw, source-level editing is also an essential skill. Web languages were designed, after all, with text-mode source editing in mind. No matter how sophisticated our WYSIWYG tools, there will always be circumstances where an author with knowledge

- Simple, 993-style HTML.
- Can be hosted and linked to by third others.
- Every piece of content is placed in a distinct DOM element, addressable by a CSS selector or classname
- Blocks of structure, such as a sidebar widget, also addressable by CSS classname.

3. Technical Approach

Cascading Tree Sheets



CTS is a declarative language that relates structure in one document to structure in another, **grafting the two trees together**. It is compatible with any tree or graph data model: HTML, JSON, XML, RDF, etc.

A tree sheet can be used for
- *templating* (e.g., JSON -> HTML)
- *scraping* (e.g., HTML -> JSON)
- *retargeting* (e.g., HTML -> HTML)

and supports the following relations:

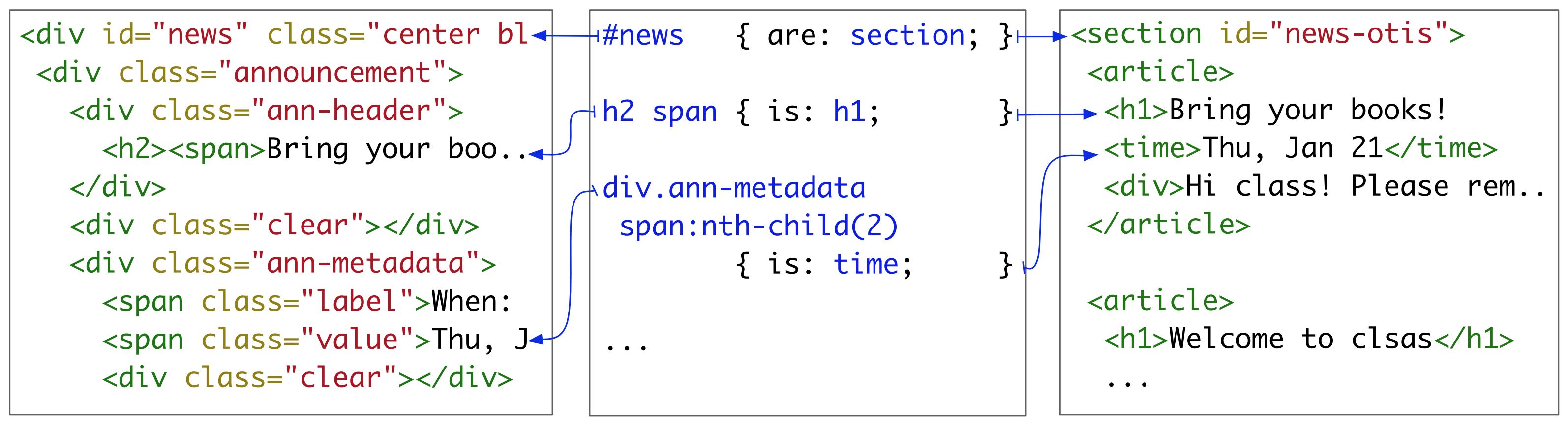
is	Defines two equivalent elements.
are	Defines two equivalent sets.
if-exist	Specifies boolean AND dependence.
if-nexist	Specifies boolean XOR dependence
recast	Specifies a point of transclusion.

Related Work

XSLT provides a way to specify the transformation of one tree into another, but requires users learn a programming language and commingle its commands with the source document.

HTML5 Components provide the technical capability to import and use HTML widgets but lacks the ability to talk about sets and conditionals and requires that widget definitions be explicitly written as a component, rather than a mockup.

Bricolage-style Retargeting [Kumar et al, CHI 2011] provides automatic retargeting, but exploits visual structure to do so and lacks a language to describe its transformation decisions. We are interested in documents that remove the visual structure completely and provide such a descriptive language.



```
<div id="news" class="center bl
 <div class="announcement">
   <div class="ann-header">
     <h2><span>Bring your boo..
   </div>
   <div class="clear"></div>
   <div class="ann-metadata">
     <span class="label">When:
     <span class="value">Thu, J
     <div class="clear"></div>
```

Design Mockup - Ordinary HTML

```
#news     { are: section; }

h2 span { is: h1;       }

div.ann-metadata
 span:nth-child(2)
        { is: time;     }

...
```

Cascading Tree Sheet

```
<section id="news-otis">
 <article>
  <h1>Bring your books!
  <time>Thu, Jan 21</time>
  <div>Hi class! Please rem..
 </article>

 <article>
  <h1>Welcome to clsas</h1>
  ...
```

Simple Content HTML

Content HTML

Simple HTML. Just enough structure to delineate content.

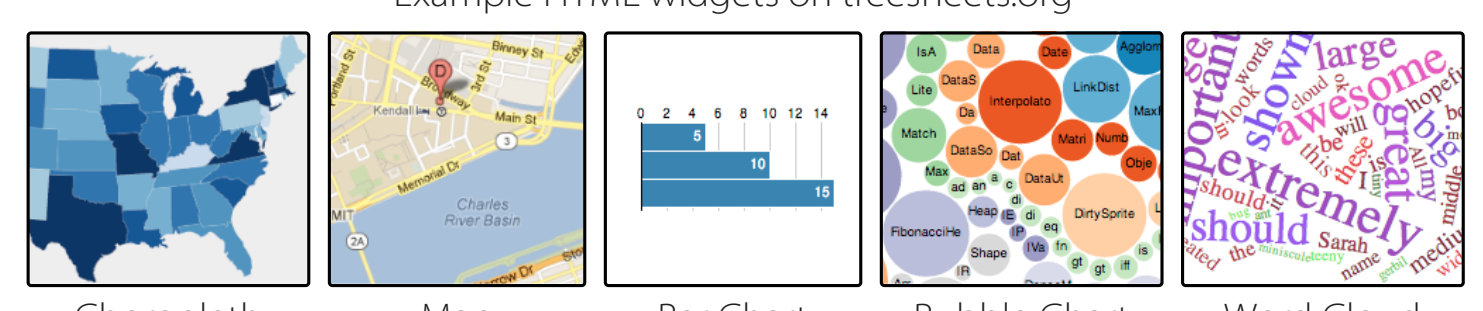
```
JSON                                   HTML
[                       collection    <ul>
  {                                     <li>
    name: "Spock",                        <h1>Spock</h1>
    position: "XO"                        <h2>First Officer</h2>
  },                                    </li>
 {                      object          <li>
   name: "Kirk",                          <h1>Kirk</h1>
   position: "Captain"  property          <h2>Captain</h2>
 }                                      </li>
]                                      </ul>
```

The HTML complexity required is the minimum necessary to achieve an isomorphic mapping to JSON.

HTML as Calling Convention

CTS enables simple HTML to become a calling convention into more complex layouts, widgets, and Javascript-laden functionality. By hand-authoring HTML and using the right CSS classes, the CTS engine transforms the HTML into a widget implementation, complete with Javascript.

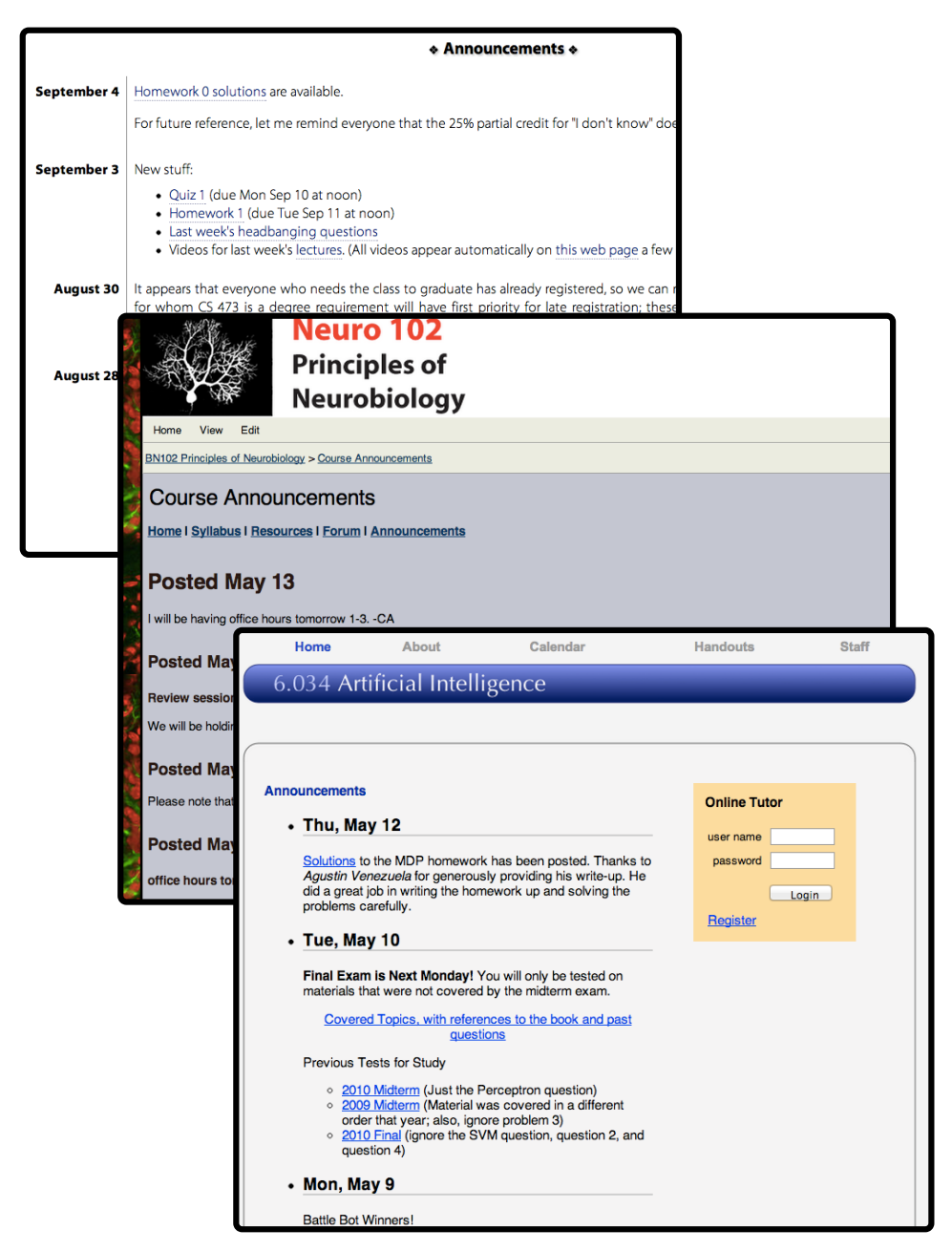
Example HTML widgets on treesheets.org



The user experience remains HTML, either WYSIWYG or raw source, and the content retargeting is performed by including a tree sheet just as one would include a style sheet.

4. Experiment

Dataset



Three course announcement pages from university courses.

Participants

Blah blah blah

Tasks

Copy
Locate and copy the the nth announcement and paste it into a text box.

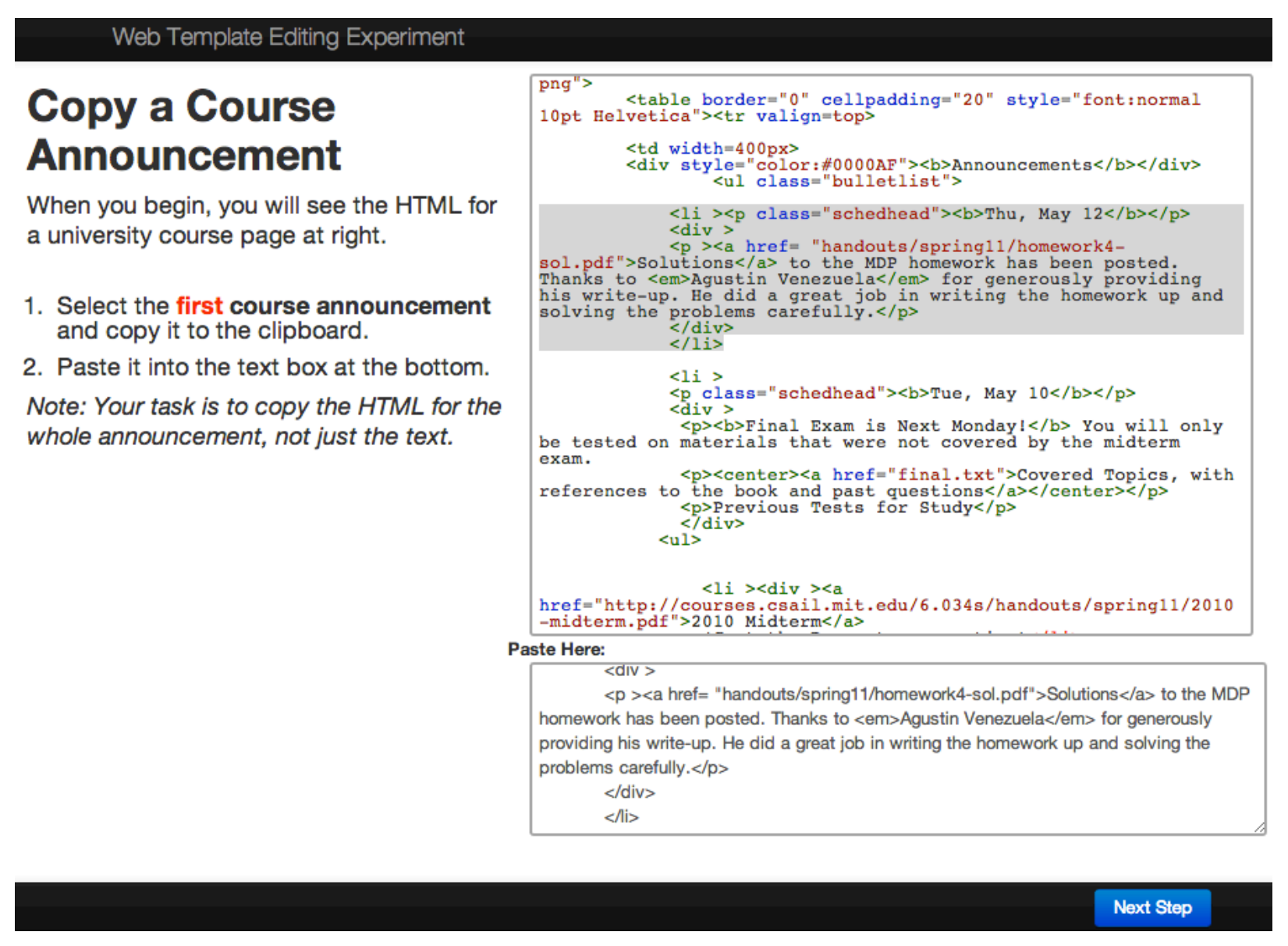
Paste
Locate the nth announcement and paste a pre-copied blob of HTML after it.

Edit
Locate the nth announcement and change its title to a pre-specified string.

Methods

HTML

Using the HTML as existed on the course page.



Web Template Editing Experiment

Copy a Course Announcement

When you begin, you will see the HTML for a university course page at right.

1. Select the **first course announcement** and copy it to the clipboard.
2. Paste it into the text box at the bottom.

Note: Your task is to copy the HTML for the whole announcement, not just the text.

```
png">
      <table border="0" cellpadding="20" style="font:normal
10pt Helvetica"><tr valign=top>

      <td width=400px>
      <div style="color:#0000AF"><b>Announcements</b></div>
          <ul class="bulletlist">

            <li ><p class="schedhead"><b>Thu, May 12</b></p>
            <div >
            <p ><a href= "handouts/spring11/homework4-
sol.pdf">Solutions</a> to the MDP homework has been posted.
Thanks to <em>Agustin Venezuela</em> for generously providing
his write-up. He did a great job in writing the homework up and
solving the problems carefully.</p>
            </div>
            </li>

            <li >
            <p class="schedhead"><b>Tue, May 10</b></p>
            <div >
             <p><b>Final Exam is Next Monday!</b> You will only
be tested on materials that were not covered by the midterm
exam.
             <p><center><a href="final.txt">Covered Topics, with
references to the book and past questions</a></center></p>
             <p>Previous Tests for Study</p>
             </div>
          <ul>

              <li ><div ><a
href="http://courses.csail.mit.edu/6.034s/handouts/spring11/2010
-midterm.pdf">2010 Midterm</a>
```

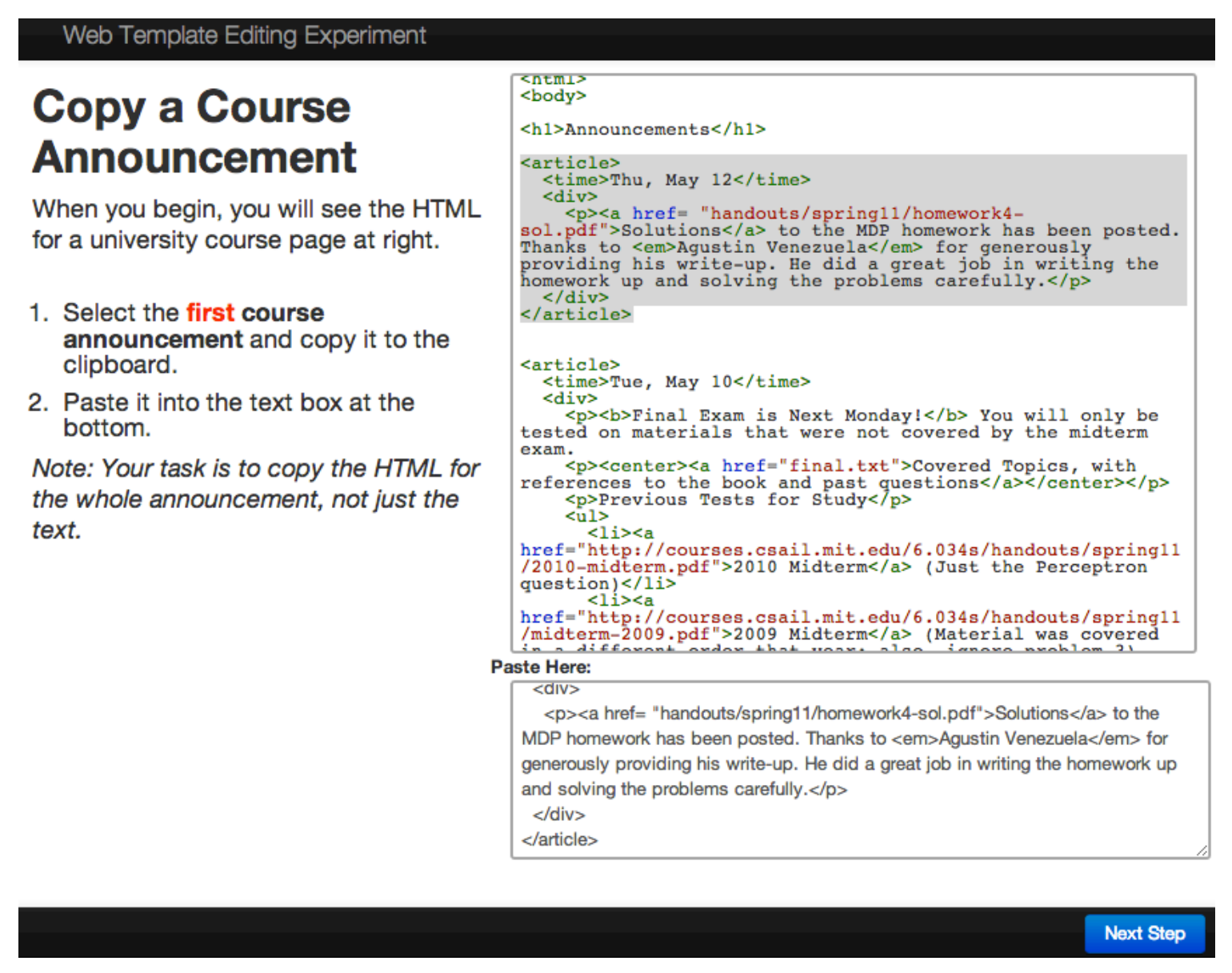
Paste Here:

```
        <div >
        <p ><a href= "handouts/spring11/homework4-sol.pdf">Solutions</a> to the MDP
homework has been posted. Thanks to <em>Agustin Venezuela</em> for generously
providing his write-up. He did a great job in writing the homework up and solving the
problems carefully.</p>
        </div>
        </li>
```

Next Step

HTML-Simple

Using a simplified content document containing only the minimal HTML structure necessary to delineate content items and fields.



Web Template Editing Experiment

Copy a Course Announcement

When you begin, you will see the HTML for a university course page at right.

1. Select the **first course announcement** and copy it to the clipboard.
2. Paste it into the text box at the bottom.

Note: Your task is to copy the HTML for the whole announcement, not just the text.

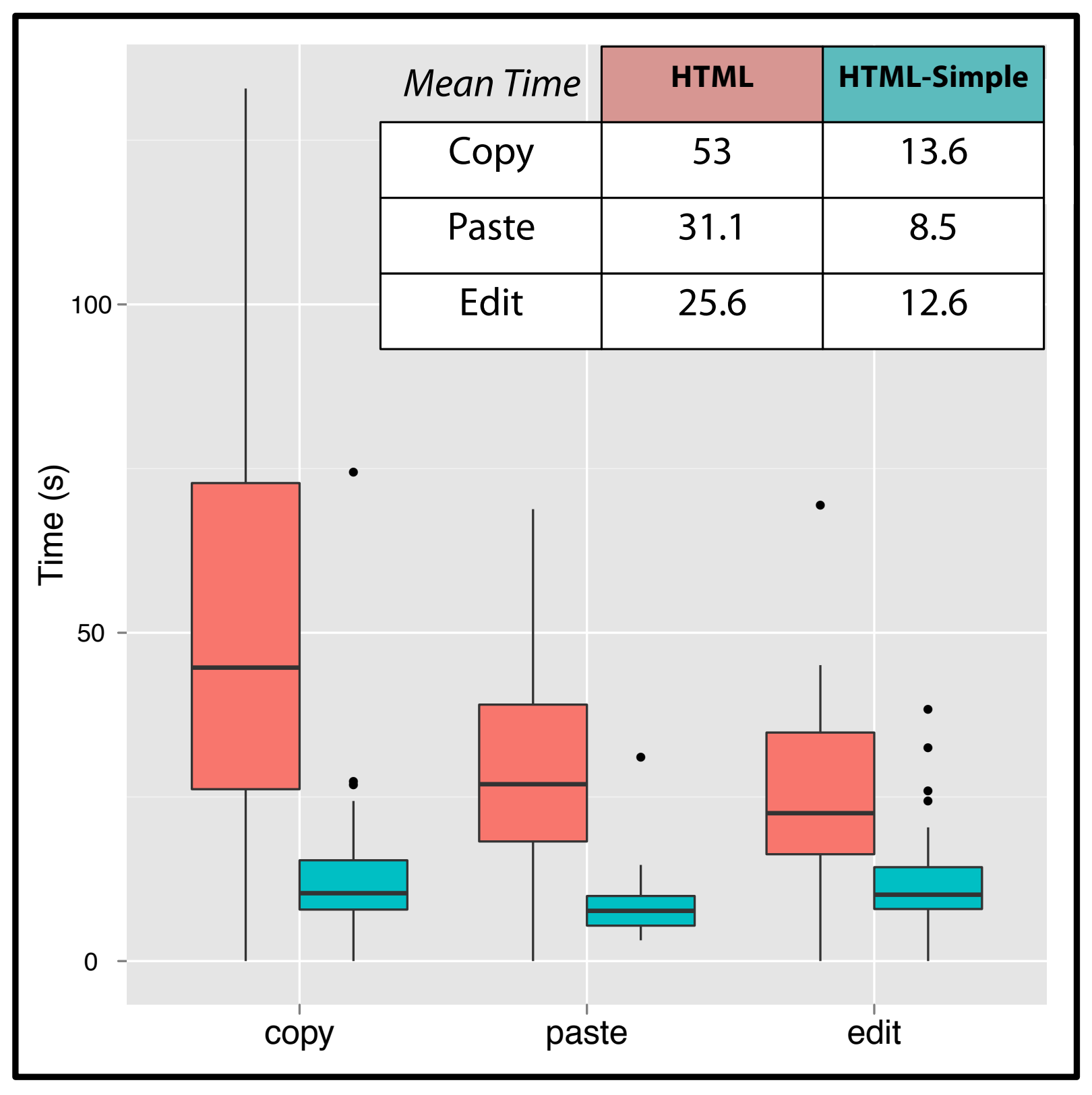
```
<html>
<body>

<h1>Announcements</h1>

<article>
  <time>Thu, May 12</time>
  <div>
    <p><a href= "handouts/spring11/homework4-
sol.pdf">Solutions</a> to the MDP homework has been posted.
Thanks to <em>Agustin Venezuela</em> for generously
providing his write-up. He did a great job in writing the
homework up and solving the problems carefully.</p>
  </div>
</article>

<article>
  <time>Tue, May 10</time>
  <div>
    <p><b>Final Exam is Next Monday!</b> You will only be
tested on materials that were not covered by the midterm
exam.
    <p><center><a href="final.txt">Covered Topics, with
references to the book and past questions</a></center></p>
    <p>Previous Tests for Study</p>
    <ul>
      <li><a
href="http://courses.csail.mit.edu/6.034s/handouts/spring11
/2010-midterm.pdf">2010 Midterm</a> (Just the Perceptron
question)</li>
      <li><a
href="http://courses.csail.mit.edu/6.034s/handouts/spring11
/midterm-2009.pdf">2009 Midterm</a> (Material was covered
```

Paste Here:

```
  <div>
    <p><a href= "handouts/spring11/homework4-sol.pdf">Solutions</a> to the
MDP homework has been posted. Thanks to <em>Agustin Venezuela</em> for
generously providing his write-up. He did a great job in writing the homework up
and solving the problems carefully.</p>
  </div>
</article>
```

Next Step

Results



Mean Time	HTML	HTML-Simple
Copy	53	13.6
Paste	31.1	8.5
Edit	25.6	12.6

5. Discussion

This study shows improvement for content editing. Is there similar improvement for design editing?

Do simplified source edits approach the usability of WYSIWYG editing?

At what point does using HTML as a calling convention (e.g., for widget parameters) require too much overhead to provide a benefit.

How much of a CMS is really just a wrapper around mockup grafting and friendly content editing operations?

Is this method of design and content grafting capable of replacing exiting procedural templates?

Could the same annotations used to graft content onto mockups be used to provide domain-independent CMS capabilities?