

Coursebuilder: reusable multimodal elearning resource editor

WenChen Hol

Centre for Academic Development, University of Auckland

w.hol@auckland.ac.nz

Craig Housley

Centre for Academic Development, University of Auckland

c.housley@auckland.ac.nz

Abstract

CourseBuilder is an online tool for developing educational websites. The software is based on templates and elements devised by learning designers and academics to meet the needs of students from a range of disciplines.

Wyswig, multimodal, reuse and enable the delivery of seamless learning experience are its special characters. This article will provide an overview about this tool.

For detail information or author's previous presentation can be found in the appendix.

1. Introduction

Goal:

To help people create content that looks good and easy to learn.

Initiative:

It aims to provide a tool to help academics build courses like those developed during projects e.g. Biblical Hebrew Vocabulary, Nursing704, Marketing course.

Concept:

Using card-making or scrap-booking techniques to provide examples, this tool provides packages that contain nice paper, stamps, tools so that the user can create resources as good as the samples.

What's our approach?

Reuse: web tricks and learning design

Open: xml, template driven, easy to modify and add new designs.

2. How could it help improve the quality of learning

Look good:

Use book as an example. This tool aims at enabling academics to create eye catching ebook without the need to touch html/css/javascript.

Backed by very flexible technology, this tool can easily bring in lately design mojo to the use of general people. Transforming your teaching resource into e-scrapbooking like resource within this tool is simply some clicks and choice.

Easy to learn:

The book can be multimodal for text, image, media presentation together, along with many other designs(slide show, tab, accordion, show/hide)--enable teacher transform their face to face teaching script into content. Content could look concise but also very rich; with teachers' thought/explanation/notes/tips etc, highlighted or embedded within the terminology, text, image, link, or media file in the content.

Teacher can constantly improve the teaching resource whenever they think of or when students asked. Answers to students query can be simply embedded within the context, not only by chat, discussion or announcement; every students, now or future, could easily see the extra-information, tips, notes, answers of questions that people asked before, within the context.

Put says in the book:

Comments/questions/discussions/brainstorming activities/student notes etc can be embedded within content. Teacher can convey some of their face to face teaching sequences within the book; what people said could be read for readers now and future, within the context and one click away.

3. Advantages of this tool

Multimodal editors:

For Image or text/link annotation (Mind Map, Interface, Flow Chart, Products, photograph, eBook, supplementary information, for web, file, media resources, note etc) with text, image, link, media files all together. Help students learn faster. A page can contain rich information than with a traditional design and still look concise.

Media rich form/quiz editor:

Build your own questionnaires to get students to engage in brainstorming, poll, pre/post program survey, feedback, comment, sharing information, or quiz etc.

With it, course content can become a blog and with own made questionnaires for reader to feedback so to students needs and improve the course overtime. All the relevant information generated through the community will be closely embedded (one click away) within the learning resource.

Maths equation:

For science subject, equation writing is very important. no need to remember 300 latex symbols, combos or formats or hand craft the complex mathML etc. User can create/update/move/style an equation in this tool.

Reuse content, web and learning design:

Backed by very flexible technology, this tool can easily bring in lately design mojo to the use of general people. And because xml file approach, copy content/designs across course(s) is very easy (a filter to replace course id, page id etc); custom css, web, learning design, quizzes, media resources etc can be reused. All the efforts can be scaled up, not just a one off luxury.

This allows us to bring out good e-designs within our courses, blogs or journal articles about successful cases, put together as exemplar pages for teachers as inspiration or source to clone from.

Course resources can be reused and customized across similar discipline; for example Enggen199(Self-study English Grammar Module for Engineering students) can be cloned and change the content of quizzes to meet the purpose of English for Medical science or Business school etc.

A well designed course could become an exemplar for other departmental courses to clone from.

As time goes by, more and more of designer's mojo, lately web tricks and good learning designs, resources can be delivered to general users.

Hybrid course structure:

A course can be tree structure, wiki and graphic (topic map) format all together.


Pagination:

Sequential content, accordion, tab glider etc designs, allow many resources (quiz, chat, forum, note, flash application, media etc) together in one page but still look concise, important points are close in space, help prevent deep dive learning, and deliver seamless learning experience. It is like studying with resources, media, quizzes, widget, teacher's tip, students talk and communication together inside a page.

Allow embed other resources:

Resources created by developer, other tool or download from share content provider etc can be embedded easily. For example: quizzes from hot potato; articulate presentation, flash application that needs images, style sheet, multimedia, javascript, xml etc all together. This feature enables developers to work with this tool without changing the source code.


4. Template development process



The desired outcome.

```
<div class="gradientHeading">$heading</div>
<p>
  $description
</p>
```

Content template. The template engine is sophisticated enough from just simple place holder or to the very complex like the show survey status template.



Form template, for user interface input

```
<idivice>...
  <idivice_name>gradientHeadingFreeText</idivice_name>
  <idivice_shortcode>gradientHeadingFreeText</idivice_shortcode>
  <heading>$heading</heading>
  <description>$description</description>
  <submitIdentifier>$submitIdentifier</submitIdentifier>
  <subid>$subid</subid>
</idivice>
```

XML template, to define the data storage, just like a table in database.

5. Template samples

Accordion xml template

```
<idevice>
  <idevice_name>Accordion Freetext</idevice_name>
  <idevice_shortcode>accordion</idevice_shortcode>
  <heading>${heading}</heading>
    <idevice_icon>${getVar( 'idevice_icon', '' )}</idevice_icon>
  <headingBgColor>${headingBgColor}</headingBgColor>
    <borderColor>${getVar( 'borderColor', '#999' )}</borderColor>
    <headerColor>${getVar( 'headerColor', '#111' )}</headerColor>
    <headerFontSize>${getVar( 'headerFontSize', '' )}</headerFontSize>
    <expandOnStart>${getVar( 'expandOnStart', 'no' )}</expandOnStart>
  <description>${description}</description>
  <subunitidentifier>${subunitidentifier}</subunitidentifier>
  <subid>${subid}</subid>
</idevice>
```

Accordion content template

```
#from tag_stripper import strip as tagstrip
#set $headingId = $tagstrip( $heading.replace( ' ', '' ) ).replace( ' ', '' )
#set $idevice_icon = ${getVar( 'idevice_icon', '' )}
#set $headerFontSize = ${getVar( 'headerFontSize', '' )}
#if $idevice_icon!="
  #set $heading = '<img src="" + $coursetype + '/icons/' + $idevice_icon
+ "" border="0" alt="" hspace="6" width="40" height="40" align="absmiddle"
/>' + $heading
#end if
#set $headerColor = ${getVar( 'headerColor', "#111" )}

#if not $varExists( 'subid' ) or $subid=="
<div id="content${subunitidentifier}_${subid}" class="clear"></div>
<script language="javascript" src="$
{ coursetype }/javascript/accordion.js"></script>
#end if
#if not $varExists( '$headingBgColor' ) or $headingBgColor=="
#set $headingBgColor=""
#end if
#if not $varExists( 'subid' ) or $subid=="
<dl id="$${subunitidentifier}_${subid}" style="border:1px solid
#999;padding:2px 2px 0 2px;">
#end if

  <dt id="ac_${subunitidentifier}_${subid}" class="accordion_toggle"
style="background-color:$headingBgColor;">
    $heading
  </dt>
  <dd class="accordion_content hideShowDesc">
    #if $varExists( 'subid' ) and $subid!="
```

```

    __adminLink__
    #end if
    <div id="$headingId">
    $description
    </div>
    #if $varExists( 'subsubnodeText' ) and $subsubnodeText<>""
        $subsubnodeText
    #end if
    #if $varExists('sub_accordion_idevices_option') and $sub_accordion_ide-
vices_option<>""
    <div class="adminLink">
    <div class="ideviceOption">
    $sub_accordion_idevices_option
    </div>
    </div>
    #end if
    <div style="clear:both"></div>
    </dd>

    ##the part for displaying in print
    <div class="caseStudyHidden">
        $description
    #if $varExists( 'subsubnodeText' ) and $subsubnodeText<>""
        $subsubnodeText
    #end if
    <div style="clear:both"></div>
    </div>
    ##

#if not $varExists( 'subid' ) or $subid=="
    #if $varExists( 'subnodeText' ) and $subnodeText<>""
        $subnodeText
    #end if
    <div class="item" style="clear:both"></div>
    #if $varExists('accordion_idevices_option') and $accordion_idevices_op-
tion<>""
    <div class="adminLink">
    <div class="ideviceOption">
    $accordion_idevices_option
    </div>
    </div>
    #end if

</dl>
<script>
var Accordion_${subunitidentifier}_${subid} = new Accordion('${subunitidentifi-
er}_${subid}');
#if $getVar( 'expandOnStart', '' )== 'no'
Accordion_${subunitidentifier}_${subid}.activePanel.next().hide();
#end if
</script>
<style>
.accordion_toggle{

```

```
        border-color:$getVar( 'borderColor', '#999' );
        margin-bottom: 2px;
    }
    .accordion_content {
        margin:10px;
    }
    .accordion_toggle {
        #if $headerFontSize=="large"
        height:40px;
        font-size:large;
        line-height:40px;
        #end if
        color:$headerColor;
    }
</style>
#end if
```

Appendix

1. Coursebuilder Demonstration:
<http://cad.auckland.ac.nz/courses/12/publish/1/index.html>
2. Coursebuilder overview:
http://flexiblelearning.auckland.ac.nz/coursebuilder_overview/index.html
3. Sand box: <http://cad.auckland.ac.nz/coursebuilder>
4. Cheetah template: <http://www.cheetahtemplate.org/>
5. Sample links:
CALM - Computer Assisted Learning for the Mind <http://calm.auckland.ac.nz>
Business Information Skills Online:
http://www.flexiblelearning.auckland.ac.nz/business_information_skills/
English Enrichment for First Year Engineering Students
<http://www.flexiblelearning.auckland.ac.nz/enggen199/>