

# Memo

**From:** Greg Sandell

**To:** Tim O'Brien

**Date:** Friday, April 25, 2005 (last updated)

**Subject:** Content Management Tool for [www.IESAbroad.org](http://www.IESAbroad.org)

## Contents

|   |   |
|---|---|
| Contents .....  | 1 |
| Purpose .....   | 1 |
| Terms .....   | 1 |
| Application Scope .....   | 2 |
| In scope .....  | 2 |
| Tentatively in Scope.....   | 2 |
| Also Desirable.....   | 2 |
| User Experience.....  | 3 |
| Software Architecture of the Public Website.....                    | 4 |
| XML Data Format Promotes Content Re-use.....                        | 4 |
| Decoupling Content & Design Simplifies Page Editing.....            | 5 |
| Centralizing Information into Metadata Eases the Task of a CMS..... | 5 |
| Caching Speeds Up Page Loading and Instant Changes.....             | 6 |
| Glossary .....  | 6 |
| Figures .....   | 8 |

## Purpose

This document describes a proposed software application for managing the content of the public website of IES Abroad. Such software can be called a Content Management System (**CMS**). The CMS would be written in-house by IES IT staff. The document covers a general description of the software capabilities from the user's perspective, and a somewhat more detailed description of the technical internals of the application.

## Terms

This document contains many terms and phrases known to IES Staff who have closely worked on the development of the IES Public website, or IT professionals, but may be less known to others. The meaning of words can be consulted from a glossary appearing at the end of this document. The first usage of a word that is included in the glossary is bolded.

The following terms will be used often in the document so they are covered here:

**Sculptor**: the company's internal code name for the project encompassing the Content Management Tool under discussion in this document.

**Whydah**: the company's internal code name for the IES Abroad public website (launched September 19, 2004).

## Application Scope

There is a general understanding at this point of what **Sculptor** will and will not do. Although this may be modified over time, the following is the current understanding.

### In scope

1. The ability to change any and all wording on the public website. There may be some exceptions to certain pages that require IT expertise to modify (such as the online application), or wording which is drawn from Empower.
2. The ability to modify the layout of the page by selecting from a predetermined list of template possibilities. For example, a page consisting of equal-width columns and a quicklinks box could be altered to column widths of 33% and 66% and an image quote box.
3. The ability to upload new images and PDFs that are referenced in the content.
4. The ability to change certain pieces of information globally. These items include: the company phone numbers (regular and toll free), the name and phone number of the Dean of Students, the contact email address for the Photo Contest, etc.
5. Role-based permissioning, controlling who has the capacity to make edits. An audit trail of what changes were done, by whom and at what time, will also be included.
6. Single-level workflows. Notification schemes by email that new material has been entered and is awaiting material.
7. The ability to promote content to the preview server or the production server.

### Tentatively in Scope

Although both of these are highly desirable features, they increase the complexity of the software dramatically.

### Also Desirable

1. Page-based Scheduling and Deferring: the ability to set target dates on an individual page basis. For example, entering alterations to both the Vienna and

Santiago landing pages, but setting different target dates for when they get promoted.

2. Rollbacks: after a promotion, the ability to undo a collection of edits and return the site to a previous state.
3. The ability to add new pages to the site. The primary scope of the application is to edit the content of existing pages. However, adding a new page is usually just a matter of adding blank generic page; once added, Sculptor will be enabled to edit the page.
4. The ability to create new template designs. The template designs were created by our design consultant Sherrie Hablitzel and are considered a fixed asset of Whydah. Sculptor will offer a selection of these templates to the user. However, it would be nice to create new designs.
5. The ability to modify decorative schemes. Decorative schemes encompass font color; background colors of table cells & features and headers images. Decorative schemes were created by our design consultant Sherrie Hablitzel, and are applied to broad groups of pages; for example, all pages for centers in Australia have fixed choices for font and background colors. However, it would be nice to create new designs.
6. Multi-level workflows in the approval and promotion process. It would be nice for Sculptor to be able to manage approvals by multiple individuals, email notifications and multiple versions of pages.

## User Experience

The following describes Sculptor's features from the user's perspective. All description should be considered to have "draft" status, subject to change as the project matures.

The application will be web-based, so activity will take place using a web browser.

The first screen the user would encounter would be a login page.

Subsequently, the majority of activity in Sculptor will be around editing individual pages. The experience of choosing the page to be edited will closely resemble the navigation of Whydah itself, that is, a cascading navigation box on the left. In addition there may be other tools for getting to desired pages quickly.

Once the desired page is reach for editing, the user would see a **WYSIWYG** editor for the main content area of a page. The content area of a page is shown circled in black in Figure 1 (see end of document).

The content area is the only part of the page that the user will be able to modify, with the exception of the title text in the header ("Santiago Field Trips" in the example above).

The editor will have a MS-Word type interface with all the familiar tools for marking up text (bolding, making lists, and so on). All the familiar features of using the cursor, selecting text, cutting and pasting will be available.

In addition, the interface will have customized buttons for:

1. Inserting feature items (such as Quicklinks Boxes) and altering template choices;
2. Turning selected text into links

Each page will have a COMMIT button, which will commit the user's current changes (assuming that the user has those privileges). Another button, VIEW IN SITE will allow the user to see the entire page as it will appear on the IES website (i.e. not just the content area).

As part of the commit process, each edit will all be validated for correctness. For the most part, the editor interface should prevent entry of any markup that does not adhere to the template rules, so validation will usually succeed. In other cases, the user may be told that there is something they must correct before continuing.

The ability to promote currently pending changes to preview or productions servers would be on a separate page. The ability to undo all pending changes will be possible as well.

## Software Architecture of the Public Website

Whydah is written as a servlet-based J2EE web application with Tomcat 5.x as the servlet container running under Java 1.4.x. An Apache 2 web server acts as a front layer to either forward requests to Tomcat or directly serve up static elements such as images and PDFs. Other technologies used in the website include JSP 1.2, JavaScript 1.2, CSS 1, XML, XSLT, Struts Framework 1.0, Apache Tiles Framework 1.1, Oracle, mysql 4.0, Apache Commons, Jdom 0.9, Saxon 1.4.1, and Xalan 1.2.

Three guiding principles drove the architecture decisions:

### XML Data Format Promotes Content Re-use

All content and metadata for the site are represented in XML. XML emphasizes interoperability and minimizes the effort of porting Whydah to a new technology platform should this be desired in the future. Much of the data in Whydah has value that reaches beyond the needs of the company site. For example, the data for fees, deadline and program dates, faculty and course curricula can easily be ported to a web service that can be read by other websites in the IES domain.

Whydah also draws data from several mysql and Oracle database resources, which are not in XML. However this data is adapted to XML so that internally, all data in Whydah is in XML.

## Decoupling Content & Design Simplifies Page Editing

Decoupling refers to separating components of a complex system so that each piece can be easily revised or replaced without affecting the operations of the other components. A simple example would be a car and its air conditioning unit. The Volvo company makes versions of the same car that runs on gasoline in North America and Diesel for other regions. However, this difference does not dictate any difference in what air conditioner can go in the car, because air conditioner units run on electricity, and cars need only to adopt a common electrical framework for supporting them. In other words, nothing about the air conditioner architecture is affected by whether the fuel architecture of the car is based on gasoline or diesel.

Decoupling is a sound practice in any software development, and principles for practicing good decoupling in Java and other object-oriented languages is well known. Whydah is written on many of these principles, but the details of this are not the focus of this document.

Of relevance to Sculptor is the decoupling of the content from the design. As covered before, all content in Whydah is in XML format. Design elements are distributed across many different components: JSP templates, Tiles pages, CSS stylesheets, XSL templates and Xpath queries. Both the content and design components are easily swappable according to the needs of the page.

For example, the text on the Vienna landing page could be changed to the text for the Nantes landing page by changing the Xpath query for the page; everything else would remain unaffected. Or the same page could have its decorative scheme switched from Austria to France simply by altering the setting of the CSS stylesheet component. The key point here is that nothing about the content has to change to accommodate a design change, and vice versa.

## Centralizing Information into Metadata Eases the Task of a CMS

Each of the components described above can be retrieved by a single key word, file path, or other collection of Strings. For example, the Xpath that determines which XML fragment is used is simply a String. The Xml and Xsl sources for a page are determined by file paths (although this will eventually be improved to a URL pointing to an XmlDB store). The JSP, CSS and Tiles sources are URIs understood by Struts.

Compiling this into one collection of data defines the metadata of the site. A snapshot of some of Whydah's metadata, in XML format, can be seen in figure 2. It shows the metadata for two of the site's pages, Sydney Field Trips and Parents Resources.<sup>1</sup> The task of Sculptor can be expressed with two simple facts: the content (text) of a page is edited by manipulating the file referenced in the metadata, and all facets of design are manipulated to altering the values of the remaining fields in the XML!

---

<sup>1</sup> URLs: <http://www.IESAbroad.org/sydneyFieldTrips.do> and <http://www.IESAbroad.org/parentsResources>.

### Caching Speeds Up Page Loading and Instant Changes

A caching scheme is at the heart of Whydah's data model. At server startup time, all metadata is loaded into application memory; each page on the site corresponds to a Java object known as a Resource. There are currently around 500 different pages on Whydah. This data is then iterated through to generate three strings that are cached for the lifetime of the application: the HTML of the content area, the HTML of the breadcrumbs, and the Javascript of the navigation on the left side. Figure 3 shows an application flow that represents what happens at startup time.

One powerful advantage of this is that pages load very fast because no database queries or XSL transforms need to be done for each page request. This also prevents taxing the servlet container with many calculations for each page request, leaving CPU cycles for the more demanding, non-cachable processes such as the Online Application. Overall the footprint of this cache is currently about 50 megabytes. With today's memory capacity, this is completely practical.

The page resources will have an API that can be manipulated from outside the application through a web service interface. This API will allow for manipulation of any of the metadata for a page, followed by a refreshing of the cached content for that page. The advantage here is that Sculptor can make on-the-fly alterations to content and design without necessitating a server restart.

### **Glossary**

**Decorative Scheme:** Choices of font color, table cell background colors of table cells and features; header images. These choices are dictated by a global design scheme affecting broad sections of the site. In the programs area, each parent country has a decorative scheme shared by all pages of all centers in that country. In the four "roles" sections of the site (Parents, Students, Alumni, Faculty) each role has its own scheme.

**Empower:** IES Abroad's Student Management System. This is a client-server Enterprise running on top of Oracle.

**CMS:** Content Management System. A tool for editing the content (text) on a website without the user having to directly manipulate HTML.

**Common page:** a page which appears to be an independent page in a section, but in fact shares the same content with other pages of the same type. For example, all centers share the same content for their Passport pages (compare Vienna Passport with Berlin Passport<sup>2</sup>). Although the two pages use the decorative scheme of their parent country, the content is in fact drawn from a single source.

---

<sup>2</sup> URLs: <http://www.iesabroad.org/viennaPassport.do> and <http://www.iesabroad.org/berlinPassport.do>.

**Feature:** elements on the right hand side of most pages, such as Image Quote and Quicklinks boxes.

**Image Quote Box:** a design element frequently used in Whydah. For example, see the right side of the page Vienna Internships.<sup>3</sup>

**Landing page:** the page at the top of the navigational tree for a given section. For example, the Australia page is the landing page for the all the centers in Australia, the Adelaide page is the landing page for the Adelaide center, and the Adelaide Student Life is the landing page for the Student Life section of the Adelaide center.<sup>4</sup>

**Quicklinks Box:** a design element frequently used in Whydah. For example, see the right side of the Vienna Housing Page.<sup>5</sup>

**Preview Server:** the computer hosting a version of the IES Abroad website meant for internal use only, showing pending changes ready for review.

**Production Server:** the computer hosting the IES Abroad public website viewable by the public (<http://www.IESAbroad.org>).

**Promote:** To cause new information to go up on a web server and be visible by a web browser.

**Template:** A fixed page design which can accommodate different text, decorative features and (in some cases) images. For example, Salamanca Housing page and Santiago Housing Page share the same template: a 66%-33% column ratio, an image quote and a caption.<sup>6</sup>

**WYSIWYG editor:** The letters stand for "What you see is what you get." For a word processor, this means that what the user sees on the screen during editing is exactly what it will look like when printed out. A WYSIWYG editor for a web page allows a user to edit a page with a view that is exactly like it will look like on the site.

---

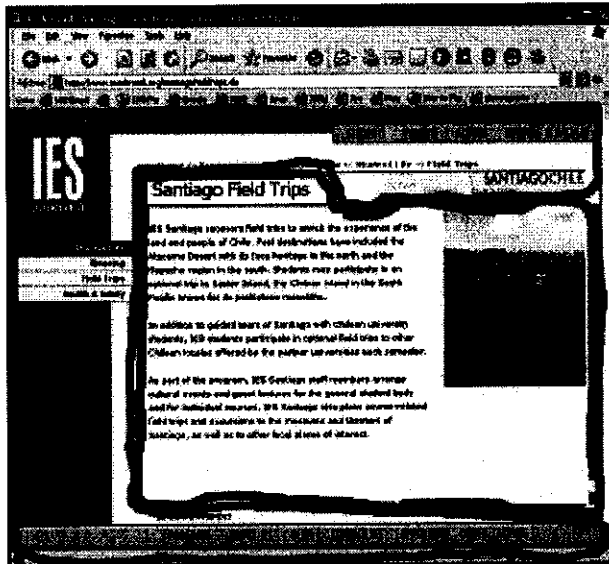
<sup>3</sup> URL: <http://www.iesabroad.org/viennaInternships.do>.

<sup>4</sup> URLs: <http://www.iesabroad.org/australia.do>, <http://www.iesabroad.org/adelaide.do>, and <http://www.iesabroad.org/adelaideStudentLife.do>.

<sup>5</sup> URL: <http://www.iesabroad.org/viennaHousing.do>.

<sup>6</sup> URLs: <http://www.iesabroad.org/salamancaHousing.do> and <http://www.iesabroad.org/santiagoHousing.do>.

## Figures



**Figure 1. Content Area.** The area in enclosed in black shows the content area that can be edited by Sculptor. The remaining area, enclosed in orange, are the areas determined by site navigation, templates and decoration scheme, and not modifiable by Sculptor.



```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <tree>
3   <resource action="sydneyFieldTrips">
4     <cssMap>
5       <uri>/pages/tiles/generic1/rolesNavTile.jsp</uri>
6       <uri>/pages/css/programsCSS.jsp?center=sydney</uri>
7     </cssMap>
8     <resourceClass>org.lesabroad.pageResources.PreLoadedPageResource</resourceClass>
9     <xpath>//entity[action='sydneyFieldTrips']</xpath>
10    <xslFile>res/xsl/templates/halfPara_imgQuote.xsl</xslFile>
11    <xmlFile>res/xml/programs/programs.xml</xmlFile>
12    <jspFile>/pages/templates/program.jsp</jspFile>
13    <pageTitle>Sydney Field Trips</pageTitle>
14    <navAction>sydneyFieldTrips</navAction>
15    <dataAction>sydneyFieldTrips</dataAction>
16  </resource>
17  <resource action="parentsResources">
18    <cssMap>
19      <uri>/pages/tiles/generic1/rolesNavTile.jsp?role=parents</uri>
20      <uri>/pages/css/rolesCSS.jsp?role=parents</uri>
21    </cssMap>
22    <resourceClass>org.lesabroad.pageResources.PreLoadedPageResource</resourceClass>
23    <xpath>//entity[action='parentsResources']</xpath>
24    <xslFile>res/xsl/templates/thirdListOfLinks_sidebar.xsl</xslFile>
25    <xmlFile>res/xml/roles/parents/parents.xml</xmlFile>
26    <jspFile>/pages/templates/rolesLinks.jsp</jspFile>
27    <pageTitle>Resources for Parents</pageTitle>
28    <navAction>parentsResources</navAction>
29    <dataAction>parentsResources</dataAction>
30  </resource>
31 </tree>

```

Figure 2. Metadata sample. A portion of the XML file the specifies all the metadata on the site. To edit the content of a page, Sculptor manipulates the file specified by the xmlFile element or the pageTitle element. To edit the design of a page, Sculptor manipulates the values of any of the other elements.

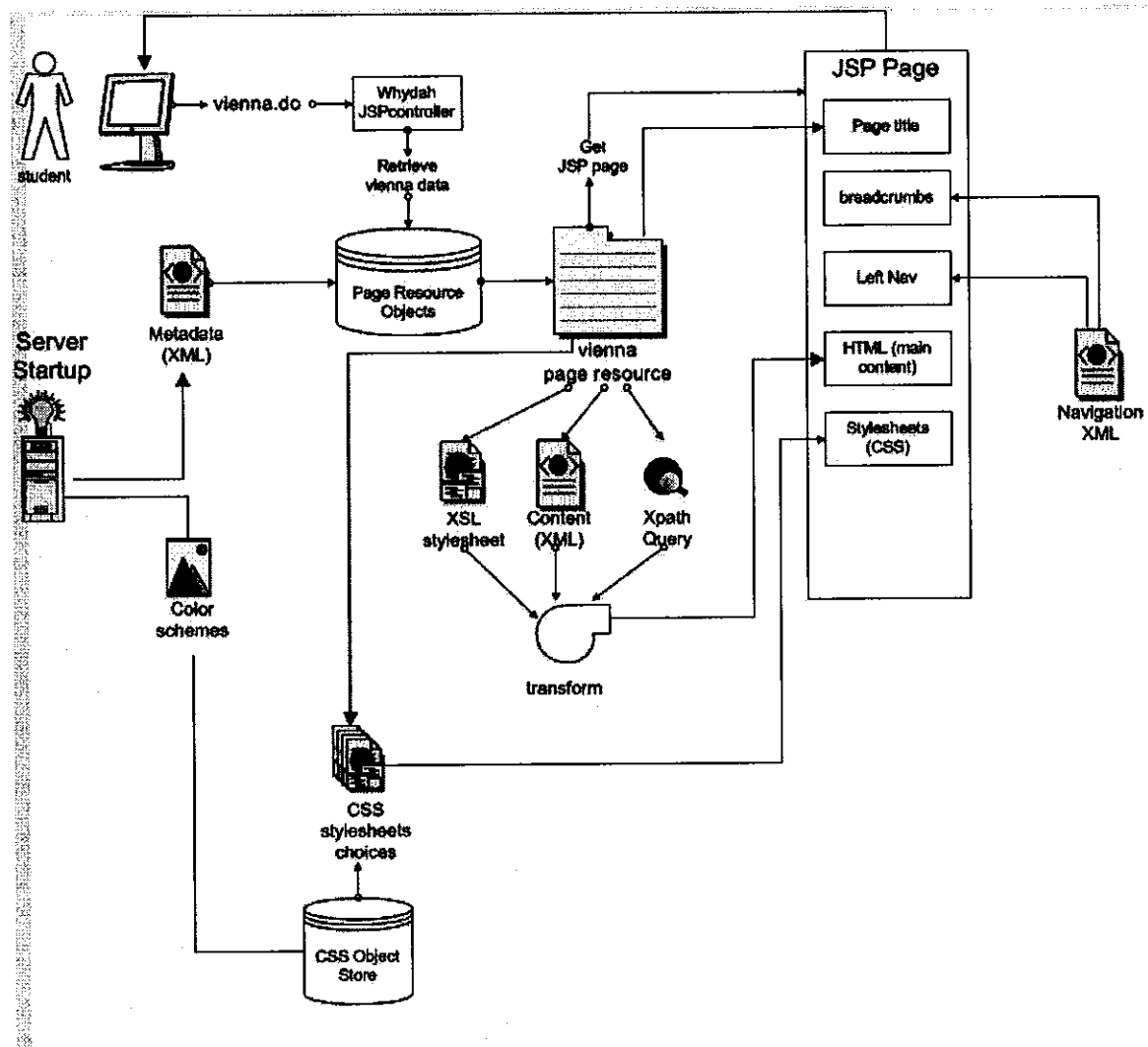


Figure 3. Whydah architecture showing the caching of metadata and HTML.