WikiCode: A Framework for Publicly Editable Web Applications

Michael H. Fischer, Loren Yu
Stanford University
Computer Science Department
Stanford, CA 94305-9035, USA
{mfischer, lorenyu}@cs.stanford.edu

ABSTRACT
Wikipedia and open source software represent a trend towards leveraging the capabilities of the masses through openly editable content. To facilitate the interaction with open source software, we have developed WikiCode, a framework for hosting web applications that anyone can easily edit in the browser. It enables developers to modify code without the initial costs setting up a development environment and checking out the code from a repository. WikiCode reduces the threshold to contribute, enabling more developers to become involved with building code, thus resulting in the development of a wider variety of features.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces. - Graphical user interfaces.

General terms: Design, Human Factors

Keywords: Wikipedia, Open Source, Online Developer Community, Browser-based Editor

INTRODUCTION
It is clear from Wikipedia that many people working together are able to produce large quantities of high quality encyclopedia articles. The closest parallel to Wikipedias in the software domain is in open source software. Just as anyone can edit articles in Wikipedia, anyone can edit open source code. One of the differences between contributing to a Wikipedia article and contributing to an open source project is the initial investment of time required to begin making the first edit. Given the current cost structure of open source projects, users intending to make small changes to the application have little incentive to invest the time to set up the development environment. By lowering the setup cost to making modifications to source code, more users will contribute to expanding and evolving the application.

We envision a world where editing open source code is as simple and effective as editing Wikipedia. As a first step in this direction, we developed WikiCode, an online frame-work that hosts web applications written in Python and HTML. Web applications hosted in Wikicode are fully editable by anyone. With a single click, users can begin editing the source code of a web application in a browser-based editor. Changes in the source code are immediately made live in the application. This interaction caters to the user group that likely has some knowledge about developing in Python, but does not have the dedication or time to check out the source tree for an open source project and become an active contributor.

TOOLS FOR HOSTING PUBLICLY EDITABLE WEB APPLICATIONS
In WikiCode, users edit web applications using a browser-based editing environment. Pages are stored in a database that contains every revision of every page. A WikiCode page can contain PyML—Python embedded in HTML—which generates dynamic pages. WikiCode can also host CSS and JavaScript files that can be linked to from other pages.

Since our implementation of WikiCode remains on a private network, we are able to neglect the security issues with executing user code. However, untrusted code can be sandboxed by executing it in a modified namespace using Python’s “exec…in…” statement. In this manner, code can be prevented from importing other modules or calling un-
safe built-in functions. Alternatively, lower level control is possible by overriding the __import__ method with a custom import method that only allows certain safe modules to be imported.

WikiCode users have the full Python language at their disposal. They also have access to special objects that WikiCode provides. The include function allows pages to incorporate other dynamically generated pages. Like Python’s import statement, the function imports the included page’s namespace, so that any variables, functions, and classes defined in the included page can be used. This allows users of WikiCode to create modules much as they would in a standalone Python application. Users can also provide an optional parameter that specifies the version of the page to import. This mechanism helps alleviate the versioning issues associated with backwards compatibility, especially in a system that changes often and without notice. A complete list of objects that WikiCode provides is listed in Figure 2:

<table>
<thead>
<tr>
<th>Object</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>include(page,ver)</td>
<td>Prints HTML dynamically generated by page and imports its namespace.</td>
</tr>
<tr>
<td>include_raw(page,ver)</td>
<td>Returns a string containing the source code of a page.</td>
</tr>
<tr>
<td>num_versions(page)</td>
<td>Returns the number of versions a page has.</td>
</tr>
<tr>
<td>page_exists(page)</td>
<td>Returns whether a page exists.</td>
</tr>
<tr>
<td>form</td>
<td>CGI form that is sent to the page using the GET or POST method.</td>
</tr>
</tbody>
</table>

Figure 2: Additional functions and variables provided by the WikiCode framework.

The WikiCode system is built around the core framework just described. For example, the HistoryViewer page, which makes all revisions of a file visible, is written in PyML and hosted in WikiCode, as are the editor and viewer pages. User-created editors can also be used to edit code. The default editor and viewer pages are read-only so that they cannot be broken and are always accessible. This default interface is consistent for all new users, making it easier to write relevant tutorials and help pages. The rest of the system, however, can be changed and evolved over time by the users of the system.

RELATED WORK
Recent research shows that lowering the barrier to entry in a system makes it possible for novice users to make simple changes that are of low risk to the community as a whole [1]. Gradually, through these inconsequential edits, novices become familiar with the details of the system. As a user accrues skills and confidence, they take a more central role in the community.

FUTURE WORK
Having lowered the barrier to contribute to code, we turn our attention to aiding specific types of contributions. The changes that users will make to web applications vary in their type and duration. The space of edit tasks includes small edits like simple changes to the HTML or larger edits like expanding current features (Error! Reference source not found.). Small edits should be simple to do while minimizing the risk of breaking the code. User interface modifications involve different interactions from code modifications. We are interested in seeing whether users will create and employ different editors in order to reduce errors through a simpler interface [2], improve efficiency by fitting specific tasks, or cater to user preferences [6].

In addition, we will further explore issues with backwards-compatibility and security. WikiCode is currently vulnerable to denial-of-service attacks since Python threads are non-preemptive and the interpreter does not register interrupts during certain “atomic” operations, such as when evaluating the expression 100**100**100. Once these critical issues are resolved, we plan to evaluate our research claims by releasing WikiCode to the public, initially seeding it with a simple but interesting web application.

REFERENCES