

## A LIGHTWEIGHT COLLABORATIVE TOOL TO SUPPORT DESIGN RESEARCH

Filippo A. Salustri

### Abstract

This paper proposes that *Wikis*, lightweight web-based collaborative tools, can provide substantive support to the design research community. The key feature of Wikis is that in principle anyone can edit content; this changes the dynamic of collaborative interactions in a way that generally facilitates the emergence of bodies of knowledge. Wikis are compared to other tools, including email lists and sophisticated content management systems (CMSs). It is found that Wikis offer a far richer functionality than email lists without the administrative overhead associated with full-blown CMSs. The basics of Wikis are explained, with particular emphasis on a new Wiki, called Xiki, being developed by the author expressly for the support of certain design research projects. Four projects are described: a dictionary of design terms and concepts, an annotated bibliography of design publications, a repository for design patterns, and a design “co-laboratory.” Each of these projects is explained insofar as they benefit by having Wiki functionality available. Based on the increasing popularity of Wikis generally, and the advantages provided by Xiki, the author expects that this technology will benefit design research, and is working to develop Xiki to meet this expectation.

*Keywords: web-based collaboration, design research, wiki, web authoring*

### 1 Introduction

There is ongoing debate (e.g. [1]) on the “scattered landscape” and “lack of common terminology” in design research. This is exacerbated by the fluid nature of design, which is necessary to allow design research to adapt to how design happens in the “real world”. Recently, a number of initiatives (like the Design Society’s Collaborative Workspace and the mailing list [phd-design@jiscmail.ac.uk](mailto:phd-design@jiscmail.ac.uk)) have begun to provide collaboration tools for design researchers. The author believes that there is room for other tools, and only time will tell which of them will emerge as “standards”.

Generally, researchers need tools that are simple to use, universally available, and flexible. Some tools, such as email lists, take a relatively “low-tech” approach, supporting relatively plain content (usually just text) and simple organisational mechanisms such as discussion threads and linear, searchable archives. Online discussion forums and blogs (an abbreviation of “web log”) provide richer content management with some logical structure (e.g. distinguishing between initial statements from follow-up comments). Collaborative web sites such as the Design Society’s Collaborative Workspace provide very rich content representations, but they do so by placing constraints on the arrangement and presentation of content; by virtue of their breadth of scope, such collaborative systems tend to be significantly more complicated to use. It is unclear at this time which approach provides the better mechanism for people to collaborate. Indeed, it may well be that individual preference could be the most significant determining factor in this matter. As far as the current author can tell, however, there has yet to be clear evidence beyond the anecdotal of this.

In this spirit, the author proposes another kind of collaborative tool: a *Wiki*. Wikis, originally developed by Ward Cunningham [2], are predominantly text-based, lightweight, and flexible platforms that allow participants to share information asynchronously through editable web pages. There are many Wikis, mostly opensource projects, which have been used in industrial, academic, and community settings. While they were originally intended as a collaboration tool, they can also be used as *personal information managers*. The author has studied Wikis for more than a year, and believes that they represent a valuable design research tool. The author is developing a Wiki, called Xiki, especially intended to support some aspects of design research. This paper will explain how Xiki works, with four examples of Wiki-based design research projects being developed by the author.

## 2 What is a Wiki?

A Wiki is a simple type of web-based server-side content management system (CMS). Implementations can range from 100-line Perl scripts (TinyWiki) to 10000+ line packages written in PHP driving an underlying MySQL relational database (TikiWiki). Wikipedia (<http://www.wikipedia.org>) provides a Wiki-based online encyclopaedia to which literally anyone can contribute, and currently contains over 750,000 pages in dozens of languages. There are even companies (e.g. <http://www.jotspot.com>) that offer Wiki functionality as a Web Service. Wikis separate layout and presentation of information from content and logical structuring of content. Layout and presentation are defined by the system itself; this assures all pages will have a consistent “look and feel” and navigational model. Content and logical structuring, on the other hand, is the responsibility of the user community. This provides the flexibility for Wikis to evolve and adapt to the needs of the users.

Wikis have a number of advantages over email lists or news groups. They allow content to be refined over time, for proper revision histories of documents to be maintained automatically, and for logically grouping similar information items. It can be extremely difficult to track the progress of online discussions via archives of email lists or news groups; a classic problem is that “Subject Lines” in email and news messages tend to remain unchanged even if the discussion itself changes completely. Wikis eliminate such problems while maintaining the capacity to let the entire user community contribute. Wikis can also be searched with more intelligent algorithms than can email list or news group archives.

In this regard, Wikis can be thought of as “free format” databases suitable even as *personal information managers*. Of particular importance to design research is that Wikis permit the structure of stored information to *emerge* through collaboration – this means that a body of knowledge does not have to be structured a-priori. Since different Wiki implementations achieve this functionality in different ways, a detailed explanation is not possible. However, in the sections below, the author will describe how Xiki provides these capabilities.

While there might appear to be significant social and legal aspects to Wiki-based collaboration, none have presented serious problems even though there are millions of Wiki pages currently on the Web. Perhaps the most common dysfunction in Wiki user communities has been called the “revert war,” wherein contributors with strongly held opposing beliefs will continually “revert” certain Wiki pages to previous versions that endorse their own perspectives, effectively erasing statements made by the opposing “faction.” This has given rise to the notion of “wiki etiquette.” A key rule of wiki etiquette is to avoid making categorical statements; for example, instead of writing “apples taste good,” one would be encouraged to write “many people find apples tasty.” Another common

rule is to never delete content, but rather move it to some other page. These simple rules have been found to prevent many confrontations between users [3,4].

Wiki pages can become disorganised quickly as more and more contributors add content. This has given rise to the practice of *refactoring*. Refactoring, originally a term of software engineering, denotes the practice of rewriting software that has slowly accreted many changes to be (a) more compact and efficient while (b) still providing all the functionality of the original version. Applied to Wikis, refactoring is the practice of replacing commentary and discussions with a properly composed summary and presentation of results. Usually, the original commentary and discussion is not deleted but rather placed in an associated Wiki page and linked back to the original. This gives users the choice of reviewing the original discussion or simply reading the summary and results. This is considered good practice, so that contributors to the original discussion can ensure that the representations made in the summary are consistent with the discussion that generated it.

The most commonly noted disadvantage of Wikis is that it is a *pull* rather than *push* technology. That is, Wikis require users to intentionally visit the appropriate pages to see recent changes and additions. Some people prefer to have such information brought to them, via email lists and newsgroups, instead of having to go to a particular site. This relates more to human preferences than to technological limitations. This problem can be addressed by implementing a notification service that emails users indicating which pages have been changed, possibly even indicating the nature of the change; the emails contain links that allow the recipient to go directly to the changed page. Some Wikis (e.g. TWiki) already implement this functionality.

### 3 Xiki Details

The author evaluated several different Wikis as part of a study of the technology, before deciding to implement a new one. Of the Wikis reviewed, the most interesting ones are very briefly noted below. The author limited his review to Wikis written purely in Perl, to help keep the administrative overhead low (at least lower than that required to support PHP and MySQL, or other very large packages). While this is by no means an all-inclusive review, the Wikis noted below do represent typical examples of what is currently available.

TWiki (<http://twiki.org>). TWiki supports a powerful authentication model, plug-ins, and multiple *webs* (collections of pages). However, it was primarily targeted at software developers and supported many functions of little use to the author. TWiki also has a complicated and somewhat inconsistent shorthand. Its implementation, at the time, was very difficult to understand.

Kwiki (<http://www.kwiki.org>). Kwiki has a very clean implementation that uses object-oriented programming style, but was its implementation architecture restricted possible additions that the author believed would be very important (e.g. extending the shorthand rules). It also performed quite slowly.

OddMuse (<http://www.oddmuse.org>). OddMuse is a very efficient though conventional implementation, with little support for plug-ins and extensions. It is very reliable for those not interested in application-specific functionality.

Though certain ideas were borrowed from all three of these implementations, the author realised that a new Wiki would be needed to support his research projects. Thus, he started to implement Xiki, which is currently at version 0.43.

Xiki, like most Wikis, has two main operating modes: browsing and editing.

With respect to browsing, Wikis appear much like any other web site: pages containing links that can be followed by any user. However, since Wiki pages are dynamically generated at “browse-time,” they also insert many interesting links that would otherwise require substantial content management. For example, most Wikis provide a link on each page that, when followed, generates a list all other pages that refer to the one (so-called *backlinks*). Most Wikis also provide links that give access to the revision history of the given page, including authorship and dates of the revisions. This allows tracing progress on specific pages easily and generating lists of “hot” (i.e. frequently edited) pages. Beyond these basics, Xiki provides a configurable menu system supporting various navigation actions. Figure 1 shows a typical browse screen for an authenticated contributor (authentication is described below).

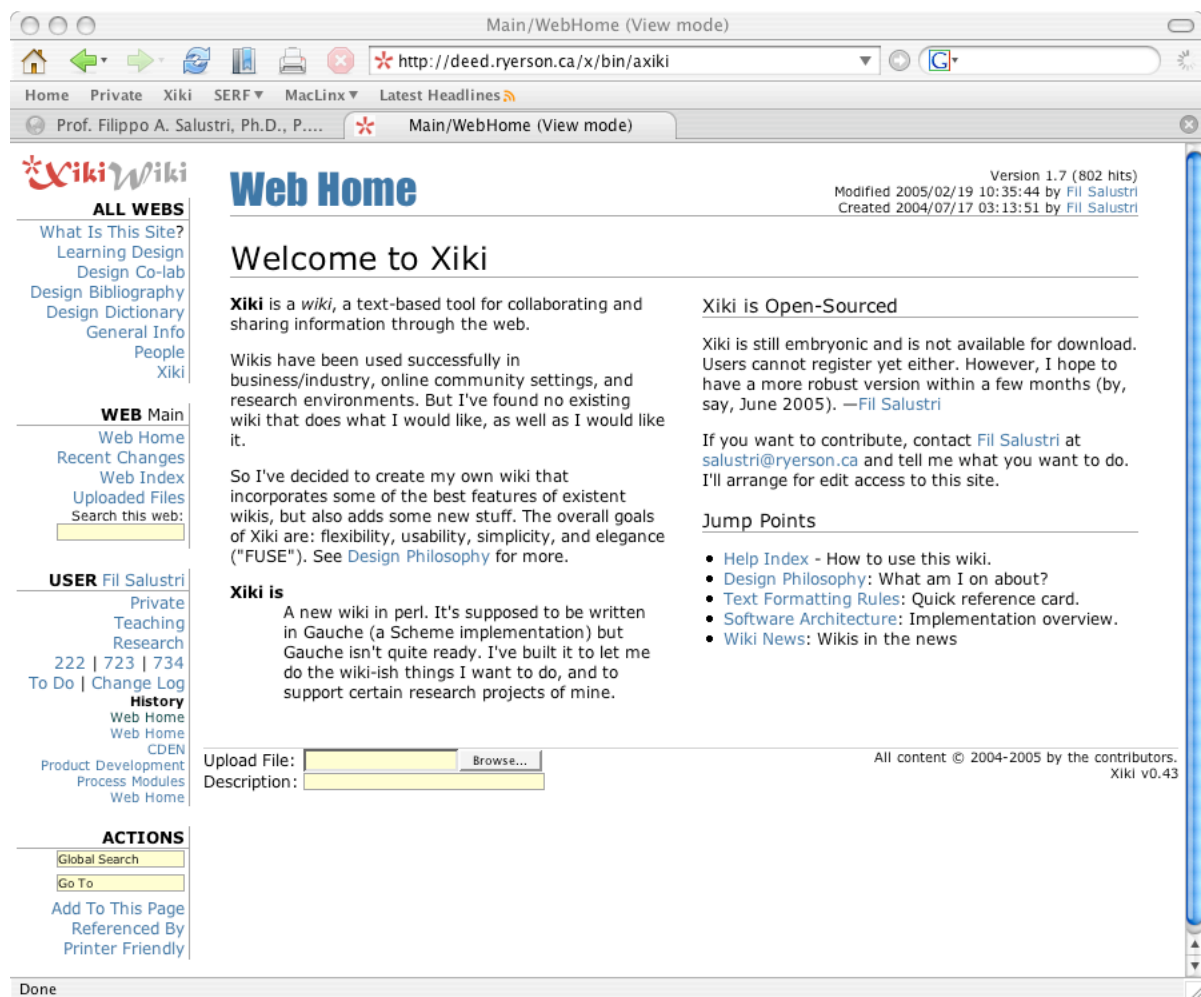


Figure 1. Screenshot of Xiki’s front page for an authenticated user (the author) under Firefox for Mac OS X.

The page title (“Web Home” in Figure 1) and the authorship information beside it are automatically generated. User-provided content appears below. The *sidebar* provides navigation links and actions. From top to bottom in the sidebar are: a list of currently available webs (described below, and configurable by the site administrator only), a list of special pages for the web Main (that can be configured by an administrator for that web or the site administrator), and finally a list of special pages specific to the given user (configurable by the user, the site administrator, or the administrator of the People web). The History portion of the user menu is simply added; one just adds the variable

\VisitHistory to one's configuration page. Xiki takes care of maintaining and updating this list.

The final section of the sidebar lists available actions. The `global search` box permits keyword searches in the content of *all* webs. The `go to` box attempts to find pages whose names match the keywords. If only one page matches, Xiki will go directly to that page. `Add To This Page` will start an edit session on the current page. `Referenced By` will display a list of all pages that refer to the current page. `Printer Friendly` creates a version of the page better suited to be printed out.

At the bottom of the page is a small form to allow authenticated users to upload "attachments" to a particular web. This allows users to upload and thus share images, other documents (e.g. MS Word documents, PDF files, etc.) A list of all uploaded files for a given web is available via the `Uploaded Files` link that is automatically generated in the WEB menu once at least one file has been uploaded.

Most Wikis keep all pages in a single flat namespace, so every page must be uniquely named. However, one of the author's requirements was to allow multiple largely disjoint collections information to be maintained by a single Wiki. This two-level hierarchy also provides an improved security model; for example, Xiki documentation, kept in its own web, is editable by a different group of users than, say, the `Learning` web. Therefore, Xiki, like TWiki, supports a two-level hierarchy: collections of pages relating to a single overarching theme are kept in a *web*, which is implemented as a directory. Some of the webs the author's own installation of Xiki include:

- `Main`: the general "front page" area for the installation
- `Xiki`: the development and documentation web for Xiki
- `Learning`: the web containing all the author's courseware and lecture material.
- `Design Bibliography`: a web implementing one of the author's current research projects – an annotated, searchable bibliography of papers on design.

In editing mode, Wikis exhibit unique behaviour, and in Xiki there are a number of special features. These are briefly described below.

Upon request (modulo authentication issues, which will be discussed later), Xiki downloads a plain text version of the content of a page to the client browser, in an HTML "textarea" box. The content, including logical mark-up, can then be edited by a user and saved (uploaded back to the Xiki server). A preview of what the page looks like when browsed appears below the editing area. The preview can be refreshed at the user's request. (The author has found this feature to be lacking in other wiki implementations.) Since textareas can contain only plain text, a "shorthand" notation is used to recognise content elements that should be treated especially. The author refers to this shorthand in Xiki as *Wikan*. For example, paragraphs are marked by a blank line, and text surrounded by asterisks, *\*like this\**, in the textarea will be rendered as bold text, **like this**, when it is browsed. Images (URLs ending in `.gif`, `.jpg`, and `.png`) are automatically "in-lined." Plain URLs are automatically converted to links. There are Wikan elements to support italics and fixed-width fonts, nested lists, headings, tables, multiple column formats, etc.

Beyond this relatively simple mark-up, Wikan automatically renders certain text patterns as links. Fundamental to all Wikis is a WikiWord, which is a series of capitalised words run together (e.g. `WeightedDecisionMatrix`). Digits and the colon count as lower-case letters. Xiki assumes that there is a page named `WeightedDecisionMatrix` and automatically renders occurrences of the string in text as a link to that page, whether the page

exists or not. Following a link to a nonexistent page causes the page to be created (in Xiki, modulo user authentication). This approach provides a simple process to add content. First, the user edits an existent page and adds a reference to some new page to be created, and saves the change. Then, in browsing mode, the user selects the newly created link, which starts an editing session on the new page.

Xiki extends this basic linking method in various ways. First, there are cases where non-WikiWords simply make the best labels for some pages (e.g. MEC222 is not a WikiWord, yet it is an obvious title for a “front page” for one of the courses that the author teaches). Xiki therefore supports `{string}` as a construct to force a more loosely defined label to be a page name. This is also quite useful for one of the author’s projects, a design dictionary, which will be discussed below.

Second, one can change how a particular WikiWord will be rendered by using `[PageName alternate title]`. In this case, a link is made to `PageName`, but the visible text associated with the link in browse mode is `alternate title`.

Xiki also supports *keywords* with the syntax `:KeyWord`. A keyword is rendered as a link that when selected produces a list of pages containing the keyword. This facilitates creating groups of related pages within a web. For example, the author keeps all pages containing lecture material for his design courses in the Learning web, but only some of these pages are appropriate for each design course the author teaches. The author adds the keyword `:CategoryMec723` to all pages for the course MEC723; a student in MEC723 can then get a list of only those pages relevant to MEC723 by following the keyword link. Keywords can be used to develop conventions for naming Xiki pages. For example, the author uses wikiwords starting with “On” to identify key topics of particular interest. Thus, a keyword `:OnMereotopology` would appear in pages relating directly or indirectly to mereotopology. The keyword is rendered as a link that produces a list of all pages that relate to mereotopology.

Finally, syntax like `[Sa102b]` is used to refer to a publication kept in the Design Bibliography web. Further details on this particular web are given below.

Other linking mechanisms are possible. For example, one might create a list of arbitrary key words and phrases, and have Xiki create links automatically on all occurrences of the keys, such that following the links will generate lists of all pages containing the keys. In time, the author will implement these forms.

Most Wikis do not allow raw HTML to be embedded in pages. However, Xiki does allow this, so long as the HTML tags use only lower-case letters, because in some circumstances it is just easier to be able to use “real” HTML. Xiki also creates a *table of contents* for a page dynamically, linking entries in the table to the appropriate section of the page.

Another significant feature of Xiki is the macro language that provides access to some of the underlying functions inside the implementation through definable variables and substitutions. Some Wikis (e.g. TWiki) provide similar functionality, but Xiki’s macro set is both smaller and targeted directly at supporting the author’s research projects.

Variables are defined with statements of the form `\def\variablename{text string value}`. Once defined, a variable is accessed as in `\variablename`. Macro functions are built in to Xiki, possibly as Perl plug-ins that are loaded at run-time, and are accessed with statements like `\functionname{text string arguments}`. In this case, a built in function `functionname` is run, with `text string arguments` as the argument. Arguments can be of any form generally. In cases where function arguments have

multiple values, a separate perl function, `Args2Hash`, is provided that parses the argument string according to the pattern `[VAR=VALUE; ]+`. The syntax is a combination of LaTeX and CSS. Parsing and evaluating macros is done before regular Wikan is translated to HTML for rendering, so the macros can contain and generate Wikan.

Variable values, function arguments, and the values returned from functions are parsed and evaluated recursively; thus, variables and functions can refer to other variables and functions. This is especially useful for implementing functions to, for example, search Xiki pages. Examples of Xiki's search capabilities include the following.

```
\search{link=Referenced By;keywords=\TheWeb/\TheTopic;} This call is rendered as a link labelled Referenced By that executes a search for the name of the current page when selected. \TheWeb and \TheTopic are Xiki predefined variables the values of which are the current web and page.
```

```
\search{form=Enter Keywords: ;webs=Research People;} This call renders a HTML form with the text Enter Keywords followed by a text box in which users can type keywords. Entering keywords and RETURN executes a search of just the Xiki webs Research and People.
```

```
\search{keywords:salustri;} Calling search with only keywords runs the search when the page containing the form is loaded, and the search result is inserted into the page in place of the form itself.
```

Xiki search queries are still quite simple; separate keywords are assumed to imply disjunction, and key phrases can be added by surrounding groups of words with double-quotes. For example, the query `engineering "design pattern"` has two terms with an implied OR between them. More complex query syntax will be added in the future. Case sensitivity is inferred from the query itself: a query containing only lower case letters triggers a case-insensitive search; if there is at least one upper case letter in the query, then the search will be case-sensitive.

As mentioned above, Xiki implements an authentication model that controls access to Xiki functionality. The WikiWord `GuestUser` is used to name unauthenticated users. Unauthenticated users see a link to `Login` instead of `Add To This Page`. Xiki currently uses only the conventional basic authentication provided by typical web servers like Apache. This delegates the entire login transaction to the web server.

Complete openness is a cornerstone of the Wiki philosophy – many Wikis, including Cunningham's original, allow editing by literally anyone, without any authentication at all. While the author agrees with the principle of openness, it does not follow that editing should be unauthenticated. In the author's view, authentication helps to ensure that contributors take responsibility for their content and get proper attribution of their contributions. As of this writing, Xiki is still not open to other users because it is still quite immature. However, by July 2005, registration will be implemented and anyone will be allowed to register. Users registering on Xiki will be assigned a WikiWord user name (e.g. `FilSalustri` for the author) and a home page in the `People` web. Authenticated users will be able to customise the behaviour of Xiki to their own preferences. They will be able to use a special syntactic form (e.g. `--FAS` in the author's case) that will be rendered as a "signature" (this is another unique feature of Xiki).

Authentication is based on listing particular *actions* that can be done by users and user groups. Only the site administrator can define a user group, with the form

`\group\AGroup{UserA UserB SubGroupC}`. Actions are typically View and Edit, but also include Search, Delete, etc. Permissions are granted at three levels: for the whole Xiki site, for a particular web, and for particular page. Two examples are:

`\allow\Edit{web=DesignTeam XikiAdmin}` This form enables editing for all pages in the current web for groups DesignTeam and XikiAdmin.

`\disallow\Delete{topic=FredFlintstone}` This form prevents user FredFlintstone from deleting the current page.

Xiki also supports revision control via the RCS system typically available on Unix/Linux platforms and freely available for other platforms. Every time that a user saves changes to a page, Xiki uses RCS to register the changes. Anyone who can browse a page can also display the revision history; anyone who can edit a topic page can also *revert* the page back to a previous version. This lets a user community quickly undo mistakes or malicious attempts to destroy Xiki contents. While one might think that these incidents are frequent in a system as open as a Wiki, experience has shown that this rarely happens [3,4].

## 4 Design research projects using Xiki

In this section, the author will describe four design research projects that he is developing that use Xiki.

### 4.1 Design dictionary

Design is an evolving, growing discipline. The language that we use to describe it must also evolve. However, for design researchers to communicate effectively, there must exist some way to describe key concepts and terms. The author does *not* intend here that design terminology can or should be prescriptively set. Instead, we need to be able to capture the “living language” as it is currently used, in a way that can be updated easily as our understanding of design changes and grows.

The goal of the design dictionary (aka *Ded*) is therefore to provide the means to represent the living language of design and to facilitate discussion about concepts and terms in design, so that we might all understand one another’s thoughts and insights more clearly. Inspiration for the operating model of *Ded* comes from Winchester’s recent book on the history of the Oxford English Dictionary [5], and has the following major elements:

**Data gathering:** individuals are encouraged to submit samples from the literature showing the use of design terms in context. Submissions for a given term are put into a page named for the term.

**Analysis:** online discussion occurs as contributors comment on the samples. Discussion can spread over multiple pages to cover specific aspects of the samples. For example, once other users are enabled on Xiki (expected by July 2005), the author expects term like *Design* and *Function* to generate an extensive collection of discussion pages.

**Definition:** based on the discussions, recorded online in the appropriate pages, a working definition is developed over time. A definition can include multiple *senses* of a term.

**Revision:** as new samples are contributed and further discussion occurs, the definition ought to be revised to capture the “living” evolution of the term and the concept it denotes. Revision should occur on a cyclical basis per the method outlined here, but the duration of a cycle cannot be predicted, and indeed could vary between terms.



Each term or concept is given its own page. The page constitutes a repository for alternative definitions, and a forum for collaborative discussion about it and related concepts. Citations to sources for background information and definitions can be easily embedded via the *annotated bibliography* (discussed below), creating links to the annotated sources online. Different terms in the dictionary are cross-referenced by simply writing the term (possibly in braces if the term does not form a proper WikiWord) and letting Xiki handle the formation of the appropriate links. Collateral discussions and content can be easily added to other pages and linked to the appropriate dictionary page by simply referencing its name. A letter index is provided in the sidebar (see Figure 2). Xiki generates the index and the number of entries for each letter dynamically. Documentation is provided to assist contributors follow the general framework for developing definitions (e.g. the *Contributors Guide*).

An authenticated Xiki user can edit existent definitions to add content for a term or concept. A new entry is created in typical Wiki fashion. For example, to add a new entry for the term *system*, one would edit the *S* index page, add a list item for *system*, and save the change. There will now be a link in the *S* index page that will start an edit session on a new page named *System*. The *Ded* default new page is designed to facilitate writing new entries.

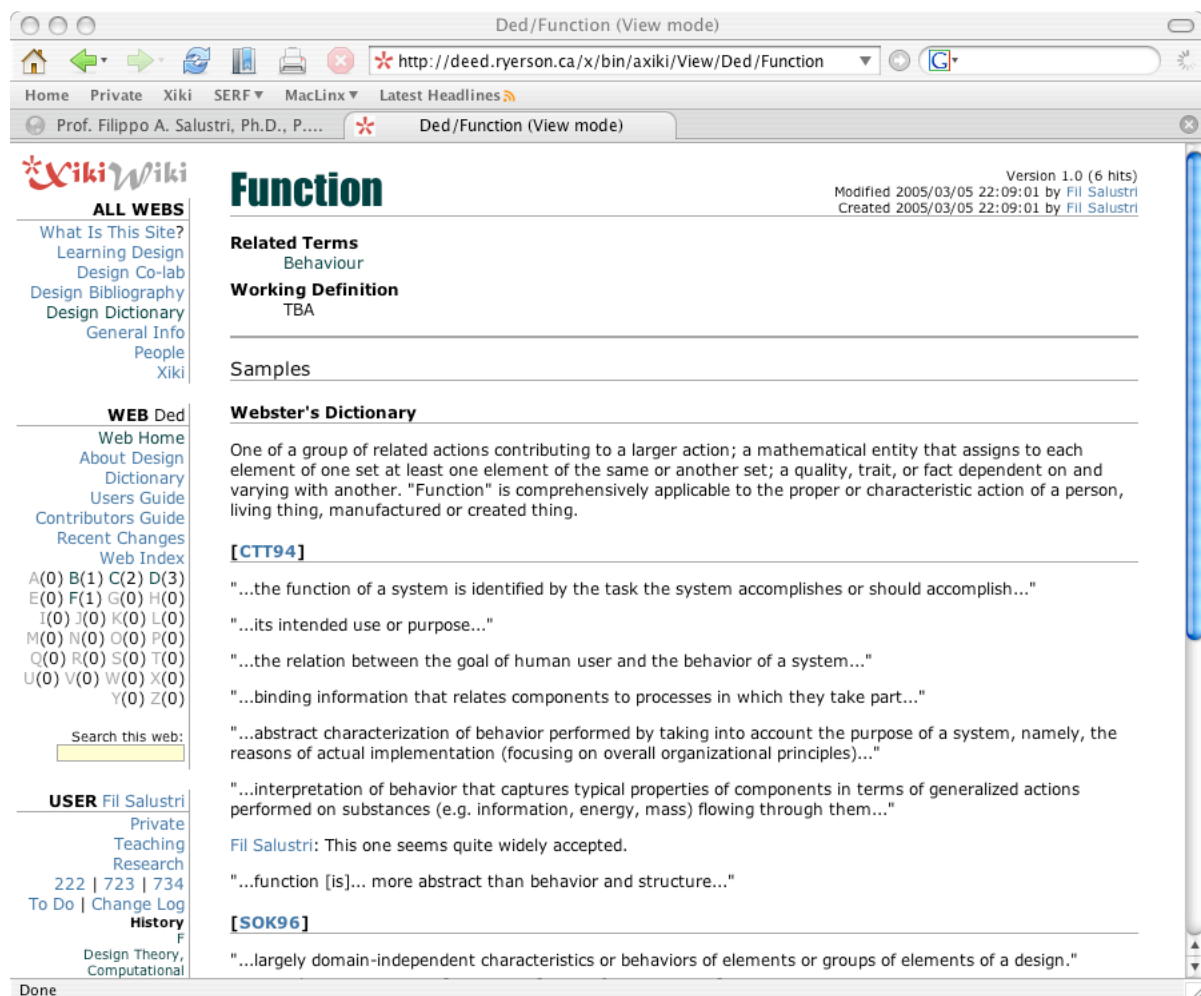


Figure 2: A sample term definition page from the Design Dictionary web.

## 4.2 Annotated bibliography of design research

Another Xiki-based project is an annotated bibliography of design-related publications, stored in the web *Abib*. More than just a listing of references, this part of Xiki facilitates providing links to downloadable copies of papers as well as forums for users to discuss each

publication. Each publication has a Xiki page dedicated to it. The citation format used is similar to the LaTeX *alpha* bibliography style – e.g. Sal02b. Xiki understands this format in square brackets as a reference to a page from the bibliography web and automatically turns such references into link to the corresponding page. Other bibliographies can be maintained in separate webs.

Creating new bibliographic entries is assisted by a web form embedded in the Xiki topic page called *NewReference* and a small CGI script that the form runs upon submission of the form. The form (see Figure 3) has separate fields for authors' names and year of publication; this information is needed to generate the citation label automatically. Upon form submission, an editing session is automatically started so that the user can add the paper's abstract and some notes about the paper. Of course, any authenticated user can add other information and cross-references in subsequent editing sessions.

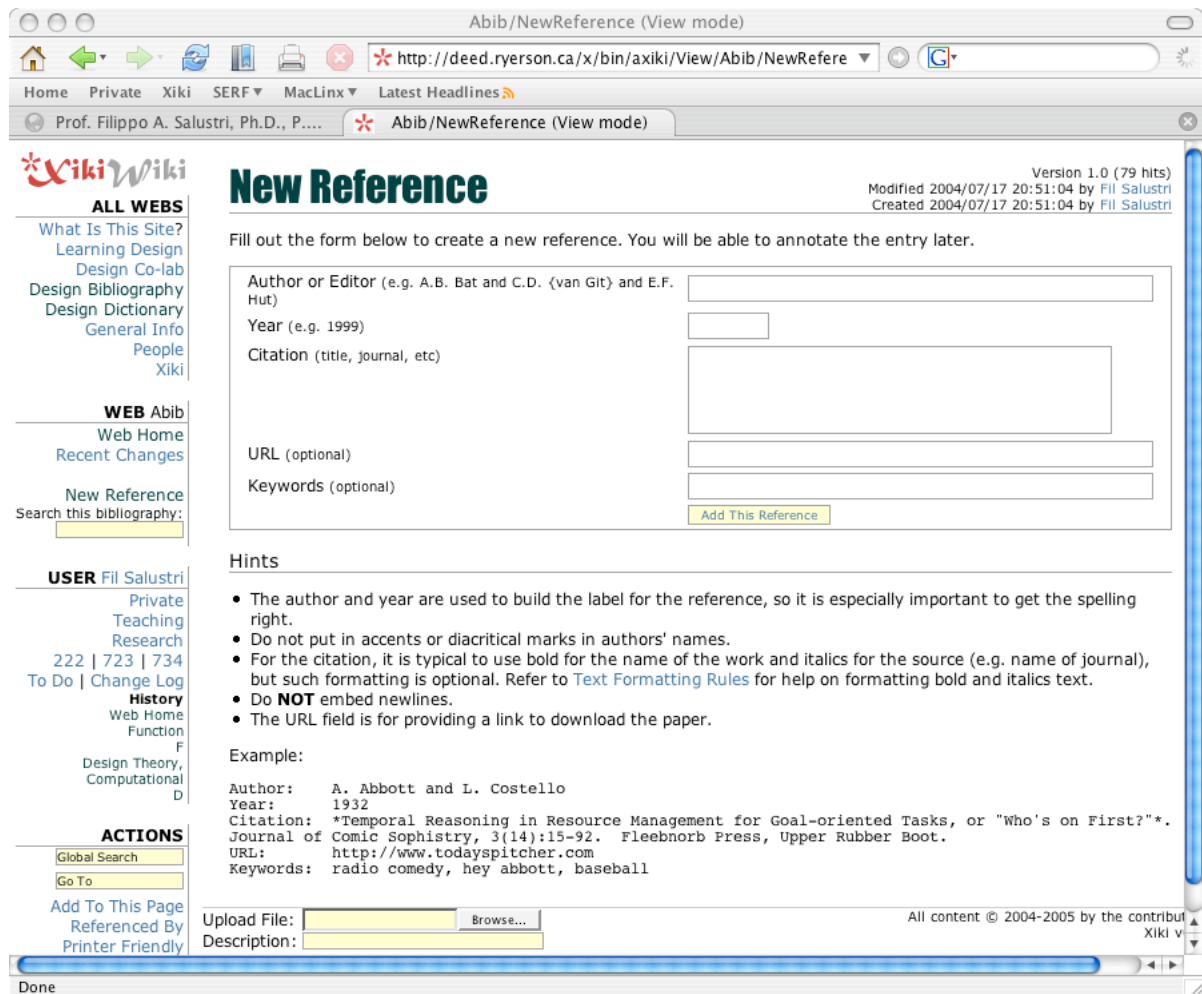


Figure 3: Form to create a new entry in the Design Bibliography web. Annotations and commentary are added after the initial entry is created, using the usual Xiki editing facilities.

Searching the bibliography is customised by a plug-in module. Typically, Xiki search results display the name of each page containing a match as a link, the number of times the keywords occur in each page, and a fragment of the page's content containing the first occurrence of the keywords. Xiki search code was implemented to support customisation on a per-web basis. In *Abib*, when a match is found, the part of the page containing the reference itself is reproduced entirely in the output; users then have complete bibliographic information available. Sample search results are in Figure 4.

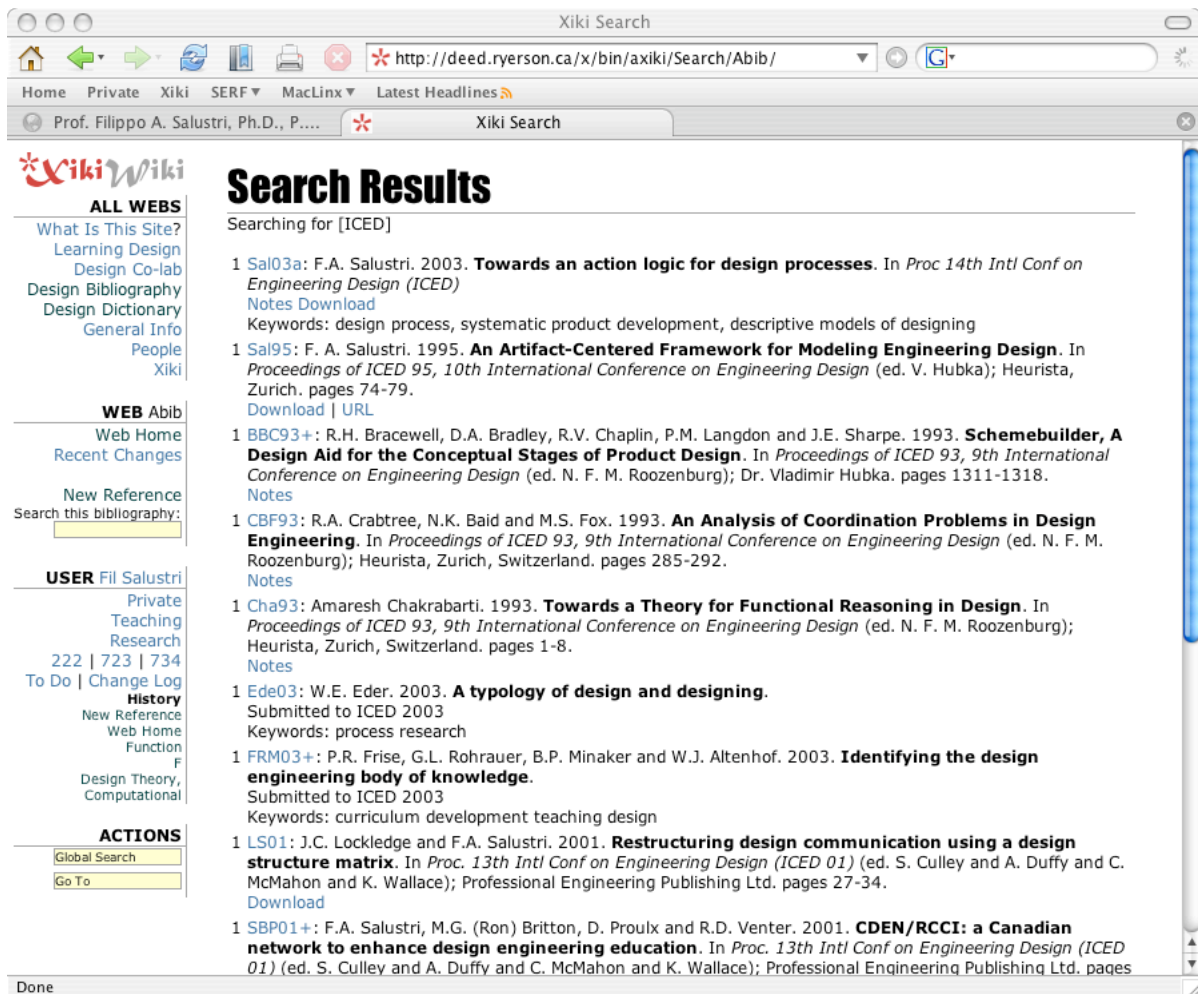


Figure 4: Typical search results from the Design Bibliography. Each entry includes the number of “hits” (number of times a search term appears in the entry), the ID label as a link that can be followed to the full, annotated entry, and a complete citation.

Of interest to the author is the ability to support an *open review* process for new publications. In a typical *double blind* reviewing processes, neither authors nor reviewers are aware of the others’ identity, and much of the reviewers’ commentaries are kept confidential. In open review, on the other hand, all information is made publicly available on the premise that such openness will promote more candid and meaningful discussion, lessen the responsibility on single reviewers, and protect against ethically suspect activities. Open reviewing has been gaining favour in a number of areas recently, including the Electronic Transactions on Artificial Intelligence; Nursing Research; the Journal of Interactive Media in Education; the Journal of Chemical Education; and the Turkish Journal of Neurological Sciences.

Abib can help here. A separate web could be set up for a conference or journal publication. The Abib labelling system can be adapted to suit the needs of the particular publication source because it is defined in a plug-in that can be replaced at run-time. Each submission would have a separate page with a link to a downloadable version of the paper. Xiki’s authentication model can be used to allow only the paper’s authors and the editor to upload new versions of the paper. The page itself would take on the appearance and function of a discussion forum in which reviewers and the author exchange thoughts and ideas about the paper. New drafts can be uploaded and linked to the page. Eventually, when an editor determines that the review process has run its course, the final version of the paper can be moved to a “proceedings” area where final papers appear. No modifications are necessary to the Xiki code itself to support open review, unless a labelling system different from the Abib

convention were used; all that would be required would be to customise the organisation of the open review web, which can be done simply by editing the appropriate Xiki pages.

### 4.3 Design pattern repository

The third project is a *design pattern repository*. Detailed in a paper submitted separately to ICED '05, this project seeks to apply Alexander's pattern language approach [6] to engineering design. Although quite successful in architecture, software design, and other fields, patterns have not yet been applied elsewhere. Patterns can be used to capture emergent *know how* about design research as well. The key feature of this project is the capacity to let a body of design know how *emerge* through collaborative online discussion.

Wikis are often used to render pattern languages. Each pattern would have its own page; the name of the pattern would be the name of the page. Occurrences of pattern names in other pages, in the form of wikiwords, would ensure automatic linking of patterns.

One use of patterns and Wikis currently being explored by the author is as a delivery mechanism in engineering education. The author is building a pattern language for an introductory course in mechanics of materials. For example, one pattern explains a `FreeBodyDiagram`, which in turn references and is referenced by other patterns, such as `BalancingMoments`. The resulting language should be sufficiently interlinked to create a true web of information that can be browsed by students in many different ways, depending on the order in which they follow links from one pattern to another.

### 4.4 A design co-laboratory

The final project being developed by the author is a *design co-laboratory*. This is the most "wide open" Xiki-based project. The `Dcl` web, which houses the co-laboratory, will provide design researchers a collaborative workspace to exchange ideas and information about design research. Pages will be dedicated to particular questions and issues. Participants will add the ideas, thoughts, and comments. Each page will have a designated editor in charge of maintaining the content, and refactoring content as appropriate.

It is difficult to imagine what structures, content, and user roles will emerge from the use of the co-laboratory. Indeed, the emergence of structure is one of the key features of Wikis. As structure emerges, refined and expanded functionality will likely be needed. This has been recognised as a valid approach to develop collaborative support software [7].

## 5 The future of Xiki

The author is planning many new features for Xiki. Two particularly relevant ones are discussed briefly here. One improvement will be to provide a facility for "free comments" to be added by *any* user. While authentication helps ensure proper attribution of content and *not* to limit access, the author acknowledges that some potential users may be discouraged from contributing by the authentication model. The free comment model, available in some other Wikis, allows someone to add a comment to a page without requiring the person to log in. The comment is typed into a text area which the page author can position wherever is most suitable in the page. Any other person can then browse the page, add a comment, and submit it, without "logging in" to the Wiki. Xiki will add the comment in an appropriate location automatically. This allows anyone to contribute, but limits such users from accessing the full Xiki editor. Attribution of comments can only be encouraged, but not enforced.

The second improvement the author will make is to let registered users sign up for an email notification service of recent changes. This improvement targets the preference of some people to have a “push” technology, in the hope to attract such users to Xiki webs such as the design dictionary or the co-laboratory. Users will be able to create a list of pages for which they wish to be advised on a regular (daily, weekly, or monthly) basis of changes. Since Xiki already tracks changes to pages, most of the infrastructure is already present. The author will create a separate program that can be run as a “cron job” (in Unix parlance). The program will then run automatically, scan the notification lists for all Xiki users, and dispatch appropriate email messages. The messages will have embedded links to the pertinent pages, so users receiving such messages will be able to access those pages directly.

## 6 Conclusions

All four of these projects discussed above will be accessible via the Web by the time of the conference. Xiki is available at <http://deed.ryerson.ca/x/bin/xiki> or by going to the author’s home page and following the appropriate link. Once Xiki is fully operational, the author and his graduate students will begin compiling information into these systems and experimenting with their usefulness. These projects, besides being useful tools in themselves, will demonstrate the usefulness of Wikis in design research settings.

### References

- [1] H. Birkhofer. 2004. There is nothing as practical as a good theory – an attempt to deal with the gap between design research and design practice. Proc Design 2004, paper #7.
- [2] B. Leuf and W. Cunningham. 2001. The Wiki Way. Addison-Wesley, Boston.
- [3] L. Aronsson. 2002. Operation of a large scale general purpose wiki website. Proc 6<sup>th</sup> Intl ICC/IFIP Conf on Electronic Publishing, pages 27-37.
- [4] D. Mattison. 2003. Quikiwiki, Swiki, Twiki, Zwiki, and the Plone Wars Wiki as a PIM and collaborative content tool. Info Today Online, April.
- [5] S. Winchester. 2003. The meaning of everything: the story of the Oxford English Dictionary. Oxford University Press.
- [6] C. Alexander, S. Ishikawa, and M. Silverstein. 1977. A pattern language: towns, buildings, construction. Oxford University Press, London.
- [7] M. Guzdial, J. Rick and B. Kerimbaev. 2000. Recognizing and supporting roles in CSCW. Proc CSCW 2000, pages 261-268.

Filippo A. Salustri  
Department of Mechanical and Industrial Engineering  
Ryerson University  
350 Victoria Street  
Toronto, ON, M5B 2K3, Canada  
tel: +001-416-979-5000 x7749  
fax +001-416-979-5265  
email: [salustri@ryerson.ca](mailto:salustri@ryerson.ca)  
<http://deed.ryerson.ca/~fil>