

# TiddlyWiki in Science Education

Franco Bagnoli

Dipartimento di Energetica & Centro Dinamiche Complesse

Università di Firenze, I-50139 Firenze, Italy

Email: franco.bagnoli@unifi.it

Peter Jipsen

Department of Mathematics and Computer Science

Chapman University, Orange, CA 92866, USA

Email: jipsen@chapman.edu

Andrea Sterbini

Dipartimento di Informatica

Università di Roma 1, I-00198 Roma, Italy

Email: sterbini@di.uniroma1.it

*Abstract*—We discuss here an ongoing experiment which uses ASciencePad [1], an adapted version of TiddlyWiki [2] as a support for teaching/learning physics (elementary and advanced) to engineering students.

ASciencePad is a “live math notebook” that is editable with a modern browser and includes a WYSIWYG editor with mathematical formulas and SVG plots. It allows also the inclusion of JavaScript simulations, that can be easily edited by users and therefore can be used as a playground for computational experiments. Our graph drawing extension allows the construction of graph maps of the learning paths to help the student navigate through the notebook topics.

This client-side tool is complemented by a wiki-based server-side system, for handling files, permissions, and integrating this tool with an existing distance-teaching framework. Our ASciencePad-based solution may be seen therefore as a “detachable wiki”.

Many of the present server-side tools, namely a system for generating and managing multiple-choice quizzes, are being ported to this new architecture.

*Index Terms*—E-learning, TiddlyWiki, Science teaching, Live notebook, Cooperative annotation.

## I. BACKGROUND

Teachers notes and students’ annotations and comments are usually two separate pieces of information. The former are mostly written and distributed in an immutable form, often in electronic form and using a web page as a distributing medium. Students’ annotations are generally taken on paper and rarely captured in digital form. Students’ comments

and questions are communicated and answered by voice, or using e-mail, web forms and chat forums, but seldom stored and/or converted to a structured form, like for instance a FAQ table.

This almost one-way communication is rather unsatisfactory, especially when teachers and students come from a very different background, like in our case of teaching physics for two master level courses in environmental engineering. Students’ feedback, possibly in the form of an edited frequently asked question (FAQ) list, is a didactically valuable tool.

In the past, we have experienced several forms of cooperative communications, mainly centered around the *wiki* concept [3]. As shown by the Wikipedia experience [4], the wiki-way may promote the elaboration of very high quality material through the self-organization of a group of users. However, the present wiki systems exhibit some drawbacks for our goals. In the first place, a WYSIWYG, mathematical-enabled wiki with math formulas, plots and charts is not yet available. We also wanted to include in the notes live computer experiments, in order to provide students with all the necessary tools for their study.

In the past we have developed a set of tools for overcoming these limitations, mainly using server-side techniques or Java applets, implemented on a widely available wiki system (TWiki [5]), which is used, at different levels, by some 200 teachers

and 5000 students in the Engineering Faculty in Florence [6]. The usage level, both of teachers and students, is however mainly limited to file sharing and institutional communications concerning courses. The system is considered unnecessarily complex and awkward to use, both due to the historical wiki “simplified” formatting rules and to server response delays.

Even worse, the requirement of a permanent Internet connection, a condition rarely fulfilled in a classroom or in a typical student’s house, is felt as a very displeasing constraint.

## II. INTRODUCTION

Teaching science, and especially physics, is traditionally done using chalk and blackboard. Notes are usually written in  $\LaTeX$  and distributed on paper.

However, this approach is far from being satisfactory. First of all, teaching is more effective using audio-visual support, like images, movies, animations, even though slide shows cannot completely replace handwriting. From the students’ point of view, there is the demand for complementing professor’s notes with their own annotations.

### A. Course notes

Teachers usually produce notes, using  $\LaTeX$ , or some WYSIWYG editor like Microsoft Word or OpenOffice Writer.  $\LaTeX$  notes are usually distributed in PDF form, and thus are not modifiable or easily annotated by students. In addition,  $\LaTeX$  is not well known among students. Microsoft Word, on the contrary, is widely available but the proprietary format is not easily portable and the mathematical rendering is quite poor. OpenOffice Writer or other WYSIWYG editors are good but not widespread. HTML is an open standard, very well documented, portable and easily modifiable by users. Standalone WYSIWYG HTML editors (like Microsoft Word) are however targeted to produce static pages, and may produce non-portable code. Recently, browser-based WYSIWYG editors have appeared, either as a replacement of textareas [7], [8], [9] in forums and wikis, or as a complete client-server application [10], [11]. On the other hand, direct editing of HTML source is not feasible. The present HTML implementations suffer from the lack of a standardized way of inserting mathematical formulas (MathML is not yet widely supported), but allow the

easy integration of multimedia and interactive tools (e.g. applets). WYSIWYG and  $\LaTeX$  documents can be easily converted to HTML, with results of varying quality.

In any case, the organization of notes is usually fixed, i.e. they have to be read linearly. Only the HTML version promotes the use of a *networked* structure of hyperlinks, which is however difficult to read and improve, due to the loss of standard search tools and *view organizers*.

### B. Multimedia Presentations

The two most widely used approaches to produce presentation slides in the scientific environment are: using a WYSIWYG editor such as Microsoft PowerPoint or OpenOffice Impress, or using  $\LaTeX$  with a suitable package [12], [13] in order to produce a pdf slide show, to be presented using Acrobat Reader or xpdf.

The first approach is more flexible, and allows the inclusion of several different media types, like for instance movies. Yet, it has portability problems, since presentation software (especially Microsoft PowerPoint) is known to behave inconsistently with respect to different versions and different architectures. It is also a closed format, with no guarantees to be maintained in the future. The OpenOffice format (among others) is open, but not widely adopted yet and still in its infancy.

The second approach is much more portable, since pdf is an open, widely adopted, standard. Writing slides with  $\LaTeX$  can be quicker than the writing a presentation if one uses a uniform layout, uses slides with a lot of mathematics and/or has already available the text as  $\LaTeX$  material. The output is rather good-looking and essentially platform-independent. It lacks some feature like fancy transition effects or animations/movies. It complements very well with printed (or pdf) notes.

Finally, one can use HTML, with a suitable JavaScript library and CSS support [14], to assemble slide shows that only require a browser to be seen. This approach may be complemented with JavaScript support for writing formulas, using jsMath [15] or AsciiMathML [16], and in general make use of all tools available to browsers, like SVG vector graphics (possibly generated using AsciiSVG), Flash or Java animations, etc.

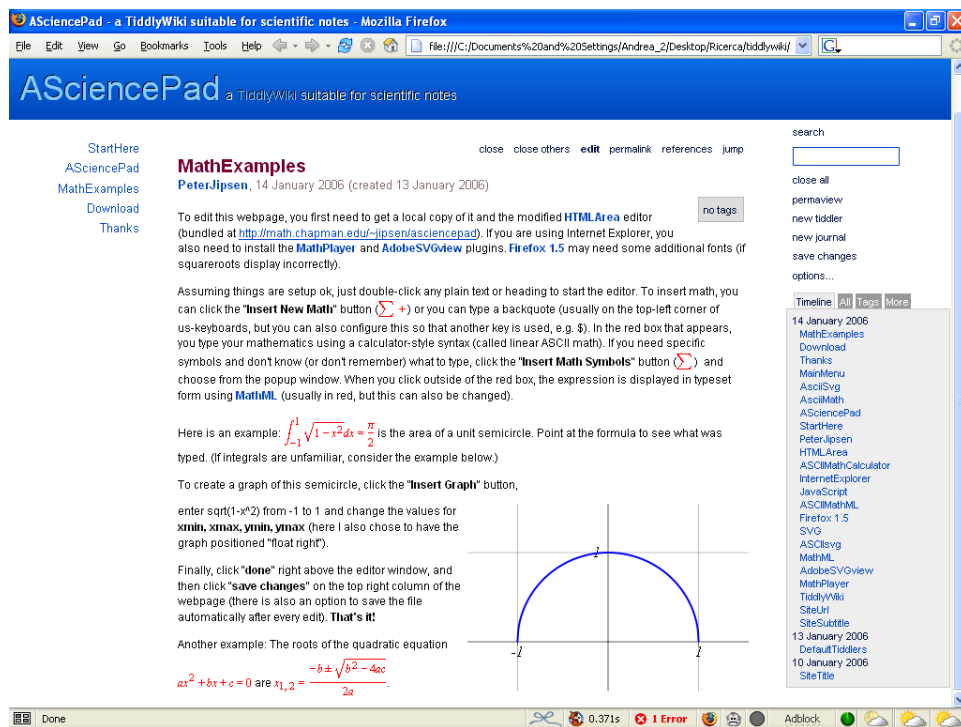


Fig. 1. ASciencePad: a TiddlyWiki with math formulae, SVG function plots and WYSIWYG editor

### C. Distance support

For distance teaching or distance support, the teaching material is usually made available for download from a web site. Many popular browsers allow one to open presentation or pdf files directly in the same windows as the web pages, thus making the transition almost invisible. For HTML presentations this step is trivial.

However, while these approaches are essentially specialized to presentation, they are not easy to use in cooperative work or discussion. Students cannot “naturally” annotate slides with mathematical comments, drawings, etc., reorganize the information flow and integrate it with other material like they are used to do with notes taken by hand.

### D. Web and human interface

A rather different approach for distance teaching goes through interacting web tools, either specialized for e-learning [17] or more generic, like blog, forums, chat rooms. One of the most promising tools is a wiki system, i.e. a web site in which pages may be directly edited in the browser using a simple syntax, and converted to HTML by a rendering engine. This approach has been made widely popular by Wikipedia. It is particularly suited for

collaborations, and may be complemented by other tools like polls, Java drawings, etc. It is also quite simple to transform a wiki page to a slide show, and some wiki systems (TWiki) implement a custom server-side plugin for this task. The advantage of this approach is that it allows a more *democratic* interaction between teacher and students.

However, the present editing interfaces are rather cumbersome. The editing interface generally implies the switching to a different view of the material, and the writer may easily “lose the point”. Even when sectional editing is used to edit just part of a page (like in Wikipedia and TWiki), this approach is not fluid enough (being generally implemented as an http server transaction) and implies a *context switching* that does not allow one to edit the section while looking at the surrounding text. This problem can be solved by using inline sectional editing, like in Quirksmode [18], possibly complemented by a WYSIWYG HTML editor. Another promising on-line approach would be an Ajax-based WYSIWYG page editor.

Yet, not all students have a permanent internet connection, and normally they would like to study on their laptop while they are offline, so the tools/web pages should be downloadable/detachable.

### E. Learning by Experimenting

Physics is better learnt by experiments, either of “physical” or computational nature (numerical experiments or animations). The best experiments are those that allow the student to tweak the parameters and the implementation itself, while *show only* experiments add little to the lesson.

Support for experiments may be in the form of written “how-to”, in order to allow students to replicate existing experiments at home or in a lab, or may be in the form of a *computational environment*, ranging from generic programming language with suitable libraries, to more specialized environments.

A HTML-based how-to may take advantage of the presence of movies and of simulations. Computational experiments preferably should not make use of a specialized environment, and should allow students to put their hands into the code. Finally, all experiments can take advantage of a collaborative environment like a chat room (possibly audio/visual), forums and FAQ compilations.

### F. Summary of requirements

In summary: the ideal tool for multimedia, interactive, distance learning of scientific topics includes:

- Support for multiple media (rich text, images, movies, animations, sound...).
- Visual instruments for rearranging views.
- Mathematical support.
- Drawing, chart, plots.
- Custom and modifiable animations.
- Support for presentation, notes and minutes.
- Collaboration, annotation and sharing of comments.
- Off-line and personal consultation/elaboration.
- Support for computational experiments.
- Fluid interface for editing, rearranging etc.

In the following, we report the results of some ongoing experiments done using ASciencePad/TiddlyWiki in teaching physics. The experiments in progress include teaching elementary mechanics and thermodynamics, and intermediate statistical mechanics and Diffusive Processes in university level courses at the Engineering Faculty in the University of Florence.

## III. THE TIDDLYWIKI APPROACH

An innovative approach has been suggested by the introduction of Jeremy Ruston’s Tiddly-

Wiki “wiki-in-a-page”, a *nonlinear personal notebook* [2]. TiddlyWiki is originally a personal HTML/XML notebook whose interface, with editing and navigation is provided by JavaScript. It is organized as a sort of rolodex classifier, whose cards (called “tiddlers”) can be opened, closed or edited inside the browser. It runs on all major modern browsers, is open software and is actively maintained and developed. A plugin mechanism allows the easy extension with new tools, like calendars, toolbars, etc., using the JavaScript language.

It appears like a standard HTML page, that can be downloaded to the personal computer and then accessed later while off-line. Changes are saved locally in the HTML file. The page provides a consultation framework with full-text search and organizational instruments. A plugin mechanism allows the easy extension with new tools, like calendars, toolbars, etc, using the JavaScript language.

Images and other objects (like applets for instance) could in principle reside inside the html page using the Data URL scheme [19]. This solution however is limited to the Firefox browser [20], implies the generation of large files and prevents the caching of images and applets. We therefore choose a multi-file approach, striping CSS and JavaScript resources (and images) off the HTML file.

Since all elaborations are performed client-side, the consultation/annotation is extremely fluid and the server stops being an interaction/performance bottleneck. There is no standard mechanism yet for uploading the material to a web repository, but several adaptations have been designed to implement this phase.

The innovative points of this approach are:

- The possibility of personal views and reorganizations of material.
- Personal notes, organized either per-topic or scheduled per-lesson.
- Off-line elaboration.
- Support for HTML objects.
- WYSIWYG edit of the page.
- Large number of plugins to extend the notebook functionalities.
- Easy access to JavaScript code to program small computational experiments with SVG-based display.

### A. Anatomy of a TiddlyWiki

Technically, a TiddlyWiki is just an HTML page with a rather large JavaScript section that takes care of displaying all the contents, and provides the interactive tools for their manipulation. The actual text of the page is not immediately visible. It is stored in a set of invisible DIV elements, called “tiddlers”. The content of the DIV tags is wiki text, i.e. text with a simple markup language, similar to (old) emails. When the user clicks on a tiddler name to show its content, the JavaScript rendering machine translates the wiki text into HTML. The text may also contain macro fragments that trigger the actions of subroutines. Other tiddlers are interpreted as a stylesheet or a JavaScript plugin.

When the user asks for a tiddler to be edited, this is replaced inline by a form, and the user is presented with the original text. The interaction with the system is very fluid, as the rendering framework is updated dynamically by the JavaScript rendering machinery.

Several navigation tools are already available to browse the tiddlers. The user can do full-text searches, or can browse the history of last changes. The system also lists orphans and missing tiddlers, i.e. tiddlers that are not hyperlinked to other ones, and links pointing to non-existent targets. Tiddlers can also be “tagged” using keywords that may serve to classify topics, so that the user can do tag-based navigation. By leveraging the structure of hyperlinks and tags we are developing tools that show which topics are pre/post-requisites of a given topic, thus building a learning path guidance system to the student, as shown in Figure 2.

### B. Client/server interaction and coordination

To enable collaboration and sharing of personal notes, a server-side repository of the TiddlyWiki notes is needed. In fact, we consider a TiddlyWiki as a “detachable Wiki” that allows off-line edit and collaboration among different users.

Our initial design uses TWiki just as a content manager, to keep track of the files needed by a TiddlyWiki stored in the system, and to allow uploading and unpacking of student-annotated notes. The personal notes added by the student would be collected as a separate TiddlyWiki attached to the student’s personal page. Next time a student gets the notes, his personal annotations are included in

the files downloaded. The student’s notes can be either “private” or “public”, the first ones being viewable only by the student, while the others are made for public consumption, and thus can then be seen/downloaded by all users, or can be collected and manipulated by the teacher to produce a FAQ or to enhance the course notes.

The last missing feature of this approach is the way of handling the updates of the JavaScript code, CSS stylesheets, images and of teacher’s notes, and how to feedback communication (questions).

The ASciencePad separation of JavaScript code and CSS stylesheets from the HTML file, and the separate storage of images make the updating procedure of these topics trivial.

For content updating, we have adopted an existing plugin for the selective inclusion and dump of tiddlers, and used the existing TWiki as a file server for communication.

necessario per forza! This extensions will make it possible to “detach” a set of TWiki pages as a TiddlyWiki, or to upload a dump of TiddlyWiki tiddlers to a set of TWiki pages.

Finally, we are porting a system for the management of mathematical multiple-choice quizzes, that uses TWiki as a user interface, to TiddlyWiki. The advantages of this are, for the teacher, the possibility of writing and checking new problems when off-line, and for students the opportunity of using the quizzes for self-assessment. Moreover, quizzes can be used as a navigational guidance system reinforcing the student’s notions by introducing self-assessment questions that link either to explanations of a wrong answer or to the following lesson.

The wiki server allows us to cooperatively exchange annotations, questions and suggestions among students and with the teacher. Moreover, this integration makes it possible to include our randomly-generated parametric quizzes [6] or to generate a TiddlyWiki notebook through the automatic selection of topics tailored to specific students [21]. The system selects coherent sets of pages where all the prerequisites of the course’s target pages are present. This (recursive) selection process stops when the pages known by the student are reached. Redundant pages (pages that present the same concepts of other pages) are also selected to enrich the presentation variability of the course. An optional optimization step can be applied to remove some redundant pages and obtain a course that

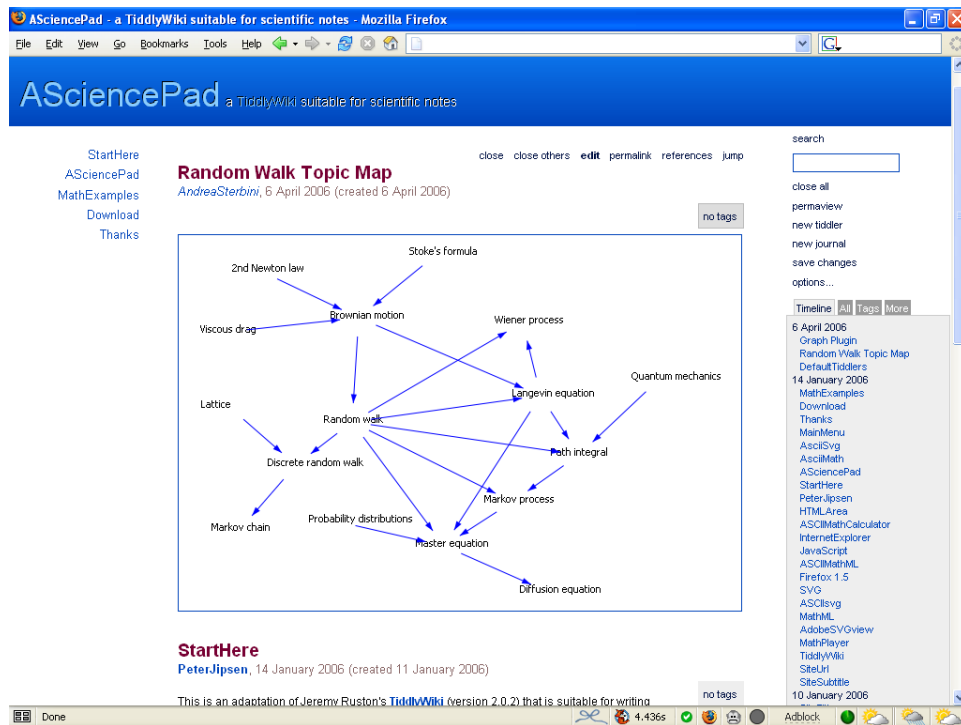


Fig. 2. The concept graph around the “Random walk” tiddler.

minimizes one of several possible weight functions, e.g. to obtain: a fast course (minimum total time), an overall easier course (minimum total difficulty), a course without too difficult explanations (minimum maximum difficulty), etc. To help the student navigating the notebook we have developed a graph map tool showing the course learning paths induced by the *prerequisite* relation, see Figure 2.

#### IV. OUTCOME

The response by selected users has been enthusiastic: the interface is fluid and responsive, and also quite usable and portable. Moreover, since it runs locally in the browser, it may be carried away, even on a memory stick, edited and used as a support tool in classrooms without requiring a permanent Internet connection. It also makes it very easy to use JavaScript as a computational language: it is sufficient to enter the JavaScript code in a “tiddler” and mark it as an executable tiddler using a special tag.

We surveyed user’s opinion and suggestions for two selected physics courses, one at elementary level (first year) and the other at higher level (first year of the second-level degree, corresponding to fourth year). These courses used, in the past years,

the TWiki pages for the exchange of teaching material and collection of questions.

The collected opinions and the usage patterns are much more promising than with the “older” interface. The students and the teacher particularly appreciated the possibility of downloading the material, with the non-linear way of reading typical of a wiki. We still have no data about the acceptance of the mechanism for cooperative collection of feedback and questions.

#### V. CONCLUSIONS

We are developing a physics “live notebook” for creating course notes that students can tweak and interact with to increase their understanding of physics.

We are completing a cooperative version of the notebook, by making it a “detachable wiki” of a set of pages of a standard Wiki server so that it’s easy to share the student notes and to coordinate the notebook editing.

We are integrating a course configuration tool (together with appropriate navigation aids) to build personal notebooks depending on the student’s initial knowledge and the course target knowledge.

Although quite preliminary, it seems that the strategy of local storage plus occasional commu-

nication, and the dynamical character of TiddlyWiki favors its acceptance by students and teachers. When extended with mathematical notation, numerical and graphing tools, TiddlyWiki may be used as a live, dynamical and upgradeable textbook. This approach complements well with traditional wikis. Currently we are using our system on an experimental basis in two physics courses for engineers at the University of Florence.

- [19] Data URI scheme, <http://en.wikipedia.org/wiki/Data:URL>
- [20] FireFox browser, <http://www.mozillaitalia.org/firefox>
- [21] A. Sterbini, M. Temperini, "A logical framework for course configuration in e-learning", Proc. *International Conference on Information Technology Based Higher Education and Training (ITHET) 2003*, Marrakech, Morocco, 2003.

## REFERENCES

- [1] P. Jipsen, *ASciencePad: A TiddlyWiki for science notes*, <http://www.chapman.edu/~jipsen/asciencepad/asciencepad.html>
- [2] Jeremy Ruston's TiddlyWiki, <http://www.tiddlywiki.com>
- [3] Bo Leuf and Ward Cunningham, *The wiki way* (Addison-Wesley, NY, 2001)
- [4] Wikipedia, the cooperative encyclopedia, <http://www.wikipedia.org>
- [5] TWiki, <http://www.twiki.org>
- [6] F. Bagnoli, F. Franci, and A. Sterbini *WebTeach: an Integrated Web-based Cooperative Environment for Distance Teaching Proceedings of 14th International Conference on Software Engineering and Knowledge Engineering (SEKE 2002)*, ACM-SIGSOFT, **519** (2002);  
F. Bagnoli, F. Franci, and A. Sterbini *WebTeach: Web Tools for Teachers and Students*, Proceedings of Frontiers in Education 02, Boston USA (2002);  
F. Bagnoli, F. Franci, and A. Sterbini *E-Learning With WebTeach: A Prototype System For Exam Delivery And On-Line Teaching* Proceedings of International Conference on Education and Information Systems: Technologies and Applications (EISTA '03) Orlando, USA, (2003);  
F. Bagnoli, F. Franci, F. Mugelli and A. Sterbini, *WebTeach in practice: the entrance test to the Engineering Faculty in Florence*, Proceedings of The IASTED International Conference on WEB-BASED EDUCATION (WBE 2004) Innsbruck, Austria (2004).
- [7] HTMLArea, <http://dynarch.com/mishoo>, <http://www.dynarch.com/projects/htmlarea>
- [8] Xinha, <http://xinha.python-hosting.com>
- [9] FCKeditor, <http://www.fckeditor.net/>
- [10] Writely, <http://www2.writely.com>
- [11] AjaxWrite, <http://www.ajaxwrite.com>
- [12] Beamer, <http://latex-beamer.sourceforge.net>
- [13] TeXPower, <http://texpower.sourceforge.net>
- [14] S5, <http://www.meyerweb.com/eric/tools/s5>
- [15] jsMath, <http://www.math.union.edu/~dpvc/jsMath/>
- [16] P. Jipsen, *ASCIIMathML: Math on the web for everyone*, <http://www.chapman.edu/~jipsen/asciimath.html>
- [17] Moodle, <http://moodle.org>
- [18] Quirksmode, <http://www.quirkmode.org/dom/cms.html>