DISTANT COOPERATIVE SOFTWARE DEVELOPMENT FOR RESEARCH AND EDUCATION: THREE YEARS OF EXPERIENCE

Michel Buffa, Peter Sander, Jean-Claude Grattarola,
{buffa | sander| grattarola@i3s.unice.fr}
Mainline group¹, I3S Laboratory, Sophia-Antipolis, France

KEYWORDS: Cooperative tools, international cooperation, distant work and software development, distant project management.

Abstract
This paper describes several experiments with cooperative distant work over the internet, principally oriented to collaborative management of software development projects.

INTRODUCTION
In this paper we will first describe the experiments with collaborative software development and project management in our research group, emphasizing the tools used during these last three years.

In the second part of this paper, we describe two distant collaborative courses 1) between students of the IUP Miage of the University of Nice (France) and students from the University of Mannheim (Germany) in 2001-2003, and 2) between the same students from Nice and students from IUP Miage of Bordeaux (France). These courses involved a semester project where students worked together remotely. The experience gained from our research group experiment’s has been reused in these courses, but many improvements both on the tools and on the methodology have been achieved in order to improve their efficiency.

COLLABORATIVE DISTANT WORK BETWEEN DISTANT PARTICIPANTS
The goal of the European TRIAL-SOLUTION project² was to develop prototype tools for slicing, indexing, manipulating and delivering scientific books. Once in the proper e-book format, books could then be accessed via the web. During that project (1999-2001), we worked closely with researchers from the University of Koblenz and end-users from the Technical University of Chemnitz. Koblenz developed the server software that gave remote access to books while we developed the authoring tool for associating meta-data with the sliced material. Chemnitz had final say after acceptance-testing of the tools.

Our needs
The need for better tools became apparent once we started to work closely with our German partners on software development and testing. It was evident that we needed a versioning tool for the software development (CVS³ was the tool we were using already), but we were also looking for a collaborative tool we could use for bug reporting, for writing reports remotely and collaboratively, for managing the project advancement, for following software milestones, for keeping a trace of our meetings, e.g., who decided what and who is going to do it. We needed a tool for keeping our knowledge up to date on every aspect of the project.

¹ http://mainline.essi.fr
² “The Trial Solution European Project”, http://www.trial-solution.de/
³ “Concurrent Version System (CVS), the open standard for version control”, http://www.cvshome.org/
Industrial tools existed, e.g., Lotus Notes or Microsoft Project, but they were expensive, used proprietary technology, did not run on every system, and were not easily able to go through firewalls. Popular public domain tools like the BSCW\(^4\) or WebDAV collaborative web server (see Whitehead et al. 1999) were just not satisfactory from our point of view: they allowed people to post messages and attach documents in a hierarchical way, but the GUI was not very user friendly, and it was lacking features we felt were necessary, principally a satisfactory versioning system. At that time, a tool was called Twiki was brought to our attention - a free, open source implementation of the Wiki concept\(^5\). TWiki is not in itself a project management tool, it’s just a web server where all pages are editable by everybody, using a web browser. Twiki quickly became the tool for everyday use. We set up some seed TWiki web pages for the TRIAL SOLUTION project - “Open Discussion”, “Specifications”, “Software Status”, “Unit Test Results”, “Download Area”, “Resources”, “Documentation” - and people from Nice and Koblenz soon started to update them remotely.

In the next sections we will briefly describe the well-known CVS versioning tool, present the TWiki tool and explain how we used them! We give our opinion on their usability. Other experiences with TWiki have been reported by (Raygan and Green 2002).

**CVS for Secure Collaborative software development**

When several people work on the same software development project, version management poses problems. The answer is in versioning management tools like Concurrent Versioning System (CVS), Visual Source Safe, etc. These tools are very popular in industry and in academic research, but strangely are not commonly used in teaching.

**TWIKI**

**What is TWiki**

We usually think that the Internet makes remote collaboration easy. Email for example, is a wonderful tool, with mailing lists, usenet newsgroups, forums, etc. There are also proprietary systems like Lotus Notes. Each of these tools is dedicated to a particular domain, each has good and bad points.

The Wiki system proposes a new method of collaboration. The author, programmer Ward Cunningham, chose the name “Wiki Wiki” ( “fast” in Hawaiian), to describe the experiment he set up for the Portland Pattern Repository\(^6\). Using his own terms: Wiki Wiki is “an informal set of web pages that anybody can edit remotely, using any web browser” (see Cunningham et al. 2001). These pages are hosted in a database and managed through a set of CGI scripts\(^7\). The system creates hyperlinks between pages in a very simple, original way. Furthermore, all pages are indexed, making searches easy.

The original Wiki Wiki system spawned many clones\(^8\). Mattison (2003) published a very interesting survey of the Wiki clones. Some are Open Source, others are free, yet others are commercial tools, or specialized for bug reporting, etc. TWiki (see Buffa 2002) is one of these clones, designed as a Wiki evolution for those working in corporate environments (see Thoeny 2000). Originally developed by Peter Thoeny when he worked for TakeFive, a software development company, this industrial-strength wiki clone is distributed under the Gnu Public

---

\(^4\) “Basic Support for Cooperative Work (BSCW)”, http://bscw.gmd.de/
\(^5\) “TWiki, a web-based collaboration platform”, http://www.twiki.org
\(^7\) “The Common Gateway Interface (CGI)”, http://hoohoo.ncsa.uiuc.edu/cgi/
\(^8\) TWiki vs other Wikis, on the official TWiki web site : http://twiki.org/cgi-bin/view/Codev/TWikiVsOtherProducts
License. As we will see, it is well suited for intranet/extranet applications and for remote project development. Furthermore, many plugins are available for extending TWiki features.

With a classical web server, you can publish and share information across the Internet, but TWiki gives each participant the possibility to remotely edit the web pages using any web browser. A technical review of TWiki’s implementation, along with a report of TWiki’s usage has also been published by Raygan and Green (2002).

**TWiki features**

The features that we have found the most useful are:

- Users can register with the tool and may ask to be notified by email when the web pages they are interested in are updated.
- All pages are indexed and a search engine is integrated into the tool.
- Statistics on any web page are available, both for browsed and for edited pages, for example, “Who was the most prolific updater this month? What page is the most popular?”
- Documents of any size can be “attached” to a web page (zip files, word documents, UML diagrams, etc.). If these documents are pictures, they may be displayed in the page.
- It is possible to check the most recent modifications to any web pages - who changed what and when. (In fact, all the history of the page can be consulted, since its creation).
- Access control is very easy to set up remotely,
- Extensibility via plugins, including pluggable look and feels.
- It is open source and easy to customize.
- It includes a very powerful template system for automatic page template generation.
- And of course, it is based on remote editing of the web pages.

**TWiki is customizable**

**Plugins.**

The standard TWiki distribution comes with a default look and feel. Many plugins are available from the official TWiki site[^9], providing many extra features like LDAP connection, chart generators, spreadsheet functionality in the tables, etc., along with many different look and feels.

**Open architecture.**

As discussed later on in this paper, we have written several scripts that use TWiki’s template mechanism and open architecture. These were needed for collaborative projects that involved many students from different locations (even from different countries).

**TWiki in our research group’s activity**

All these characteristics met our needs quite well during collaborative work with partners on the Trial Solution research project. Figure 1 shows the status page of our software in development (near the mid-point of the project). Parts are edited by different partners. The first two columns on the left correspond to parts of the specification document and other columns indicate the status of the software (implemented, tested, or validated).

In our research group TWiki has been used for project management, and was rapidly extended to become the host technology of our public and private web sites. This means various project partners can remotely edit relevant parts of our website, and group members can edit the entire website from anywhere with nothing more than a simple web browser.

[^9]: “TWiki plugins repository”, http://twiki.org/cgi-bin/view/Plugins/WebHome
TWiki has become a keystone for coordinating our research activity.

Alternatives to the TWiki tool
Even today, it seems that TWiki remains in a class of its own compared to various Wiki dialects. A survey of TWiki vs other Wikis is available on the TWiki official site\(^{10}\), and another very relevant paper by Mattison (2003) compares WebLogs, Wikis and other collaborative tools, along with interviews of the authors of these tools.

Other popular collaborative tools share some features with TWiki, like the popular PHPnuke\(^{11}\) or SPIP\(^{12}\) which are more targeted for managing community portals or online magazines. Both are based on the PHP/MySQL combo (contrarily to TWiki which is written 100% as Perl CGI scripts).

Many companies\(^{13}\) have Wikis in their intranet/extranet, e.g., INTEL, Hewlett Packard, SAP. Microsoft SharePoint Team Services seems to play the same role as TWiki in the industrial world, where open tools can arouse distrust. Users who shared experience both on TWiki and SharePoint have very strong positions\(^{14}\) in favour of TWiki. A rather interesting comparison between SharePoint TS and other commercial tools has been published by Clyman (2002). The author of this paper points out that SharePoint does not provide version control itself. Its document collaboration capabilities vary depending on file type and the applications installed on your machine. If you want to mark up an Office document, you’re expected to use Office’s annotation capabilities. Furthermore, for updating remotely a Word document, he had to download it entirely.

---

\(^{10}\) TWiki vs other Wikis, on the official TWiki web site: http://twiki.org/cgi-bin/view/Codev/TWikiVsOtherProducts

\(^{11}\) PHPnuke home page: http://www.phpnuke.org/

\(^{12}\) SPIP Home page: http://www.spip.net/fr

\(^{13}\) "TWiki Success Stories", http://twiki.org/cgi-bin/view/Main/TWikiSuccessStories

\(^{14}\) TWiki vs Microsoft Sharepoint, on the official TWiki web site: http://twiki.org/cgi-bin/view/Codev/SharePointVsTWiki
COLLABORATIVE STUDENT PROJECTS: “TELE-SEMINARS”

History
TWiki and CVS have been the main tools we used in a joint tele-seminar program set up in collaboration with Professor Wolfgang Effelsberg from the University of Mannheim, involving his students and students from the IUP Miage of Nice (France). Note that “tele-seminar” here is not used in the sense of “seminar-based video-conferencing.”

Motivation
During their professional lives, today’s students will be facing both international cooperation and competition with partners from different countries and languages. These new trends of globalization of the professional environment as well as the rapid spread of new information and communication technologies make it necessary to adapt the students’ curricula to international requirements.

2001-2002: NICE-MANNHEIM EXPERIMENT

Introduction
The international tele-seminar presented in this paper is an agreement between the chair of Praktische Informatik IV at the University of Mannheim, Germany and the IUP MIAGE at the University of Nice Sophia-Antipolis, France. Both institutes have been cooperating for over ten years in the context of the European ERASMUS-SOCRATES program and the programs of the Franco-German Institute for Higher Studies. They also signed an agreement on a Mannheim-Nice double diploma. In order to increase the number of students able to participate in such international activities, we have been working on new ideas to allow a larger number of students to study and work on modules common to both curricula. The “Tele-seminar on e-commerce” involved 24 students (i.e., 12 students of either institute) and six teachers in bi-national groups with the design, implementation, and presentation of a specific application in the field of e-commerce, which is common to both curricula.

The proposed methodology for this tele-seminar was the following:
- To work in an international group at remote locations (3 students in Mannheim and 3 students in Nice), and thus to work at distance, using Internet-based communication tools such as:
  - electronic messaging (asynchronous),
  - computer-based tools for cooperative work (asynchronous), and
  - videoconferencing (synchronous).
- Use the English language as the common language, which is a foreign language for all participants.

Details of this experiment has been published by Effelsberg et al. (2002).

Organization of the Tele-Seminar
The initial planning of the joint Mannheim-Nice tele-seminar took place starting a year in advance. Former experience with remote cooperation (though only nation-wide) taught us that it is indispensable that the participants get in personal touch with each other before they start working on a project. This personal aspect is even more important in our case of an international cooperation in a foreign language. Therefore, an initial introductory week with all participants was scheduled in Nice. As the counterpart, the concluding week with the presentation of the results was scheduled for Mannheim. In between, five groups of 5-6 students were defined and each assigned to a specific project.
Introductory week in Nice
During the tele-seminar’s introductory week in September 2001, the students got a first impression of the other’s respective level of English and technical skills, but also — and this certainly was the most important point — they got to know each other personally.

At the end of this initial week, mixed groups were formed and a work plan for the semester to come was allotted to each.

Computer-supported Cooperative Work
During the four and a half months of the winter semester with its computer-supported cooperative work at distance, our students were initiated to both asynchronous and synchronous Internet tools documents exchange, they agreed upon design specifications, discussed the distribution of the work load, set deadlines for milestones and deliverables, etc.

The TWiki Web-based collaboration platform for the storage and retrieval of documents. Students were asked to store their meeting protocols, design specifications, and project status reports on this Web server. They used it as well to upload and download large files rather than sending large email attachments. TWiki extended statistics features have been used for project evaluation: Figure 1 shows a typical TWiki web page (this screenshot comes from the Nice-Bordeaux experiment described later).

The CVS versioning tool for collaborative software development: a CVS server was running on the same computer that hosted TWiki, as well as a MySQL database server. Software development was been done in several languages including Java, PHP, C++, etc.

SSH and FTP: the server was located in Nice. For security reasons, students could only log in using the SSH protocol. Once logged they could install or configure software they needed for their projects. Students could only use the ftp client from the server itself.

Weekly videoconferencing. Both Mannheim and Nice had seminar rooms equipped with videoconferencing facilities (over IP). While the groups preferred email and the collaboration platform TWiki for discussing technical issues, the weekly videoconferences served primarily for organization of the teamwork: the group members agreed deadlines and discussed the work plan of each member.

Figure 2 A typical TWiki page, using the Koala Skin look & feel
Concluding Week in Mannheim
After the tele-collaboration phase, the entire group and the professors met again for a full week at the University of Mannheim. In order to preserve the project results and to share the experience, a website was set up containing the technical reports, the digital presentations, and software downloads of all the five projects.

EXPERIENCES

From a social point of view
The students participating at the joint Mannheim-Nice tele-seminar experienced industry-like teamwork.

Problems encountered were:

- Different languages - communication was made more difficult both by the uncertainty of speaking a foreign language and by different interpretations of words.
- The German students were used to working more independently than the French students who generally prefer a precise list of specifications by their supervisors.
- Both French and the German group members had other obligations during different periods of the ongoing semester, which sometimes made it difficult to stick to the milestones.

However, the two days of final preparation during the concluding week were very important for catching up all the difficulties encountered during the semester. Students’ final results were excellent. Thus, the participants of our novel bi-national tele-seminar not only learned the technical capability demanded by their respective projects, but they also learned about international project management.

From a technical point of view
-TWiki was a great tool. Remote editing and versioning gave us a lot of flexibility. The “natural” syntax became quickly familiar to non-Wikiers, and HTML inclusion was a plus (even if discouraged by TWiki’s authors). Since CVS was keeping track of all modifications on the TWiki web pages, usage statistics have been a very good indicator of the student’s behavior during the project. Traceability is important when using a collaborative tool in academic experiments.
-Many users wished they could edit TWiki pages under a local text editor (emacs for example). While using an HTML text area widget in a web page may be very practical for small updates, this becomes frustrating for larger works.
-Automatic email notification was also very useful. Everyday, all project members and the tutors got an email with links to pages that had recently been updated, including a five-line extract of the modifications for each page. TWiki defines so-called “webs” which correspond to a set of pages grouped under the same tab in the GUI (with the Koala Skin, it may look different with other look and feels). We created a web for each group of students working on the same subject. Unfortunately, email notification in TWiki was not fine grained enough. Students and tutors got an email each time a web was updated (any of its pages), in other words, email notification was only available at a web-level, not at a single page level. This lack of granularity was sometimes frustrating.

15 http://www.informatik.uni-mannheim.de/informatik/pi4/stud/veranstaltungen/ws200102/seminar_MA_Nice/
-Video-conferencing did not work all the time. When the network was overloaded, video conferencing simply did not work well enough to be usable.

-Security was an ongoing concern. The system administrators were paranoid and only allowed an ftp server to run on one machine in Nice and in Mannheim. SSH was the only way to log into the machine, which was isolated from all other networks (see Barret and Silverman 2000 for more info on SSH). Students had to write client software that talked with the database server located on the secure computer and this became somewhat difficult to configure.

-Students accounts had to be created by hand by one of the tutors in France, and a lot of redundancy appeared: three accounts, one each for CVS, SSH and TWiki had to be created for each student (and this does not include accounts in MySQL or PostgreSQL databases). We need to find a way to generate different accounts automatically.

-Format problems. Students wrote their reports remotely and it was a pleasure for the tutors to see the advancement of their work on TWiki. But in the end, the sponsors (Accenture France and Accenture Germany) asked for reports in Microsoft Word format - students had to cut/paste the text from TWiki and reformat everything in Word! Another point was that all TWiki pages are generated dynamically. The HTML export module in TWiki was really limited - it could generate HTML pages, but only on the server’s disk. When our colleagues from Mannheim asked for a CD with all the work done by the students, the administrator had to collect them by hand, along with the documents attached to the pages, which are located in a complex hierarchy somewhere else on the disk. TWiki is a great tool but is not open to the Microsoft world, which is the rule in industry.

-Instant messengers like MSN or ICQ were heavily used by the students, which we had not anticipated. Moreover, none of the tutors used these tools, which are very popular among the students. It was one of the best ways for them to keep in touch with their partners and to initiate a video or audio conferencing session.

Conclusion
We have briefly presented the set-up of a bi-national tele-seminar between Mannheim and Nice as a novel form of student exchange in an international context. Not only does this allow combining both technical core competencies of the different German and French approaches in computer sciences, but also the students experienced the challenges and strengths of international cooperation at remote locations. The participants have experienced the technical contents of the seminar, communication in a foreign language, learned new tools for tele-cooperation, and what project work in “real-life” is about. This demonstrates the power of the tele-seminar approach and allows defining a whole new class of curriculum elements.

All the problems encountered have been addressed in the next Nice-Bordeaux tele-seminar experiment in 2002-2003, described below.

2002-2003 NICE-BORDEAUX EXPERIMENT

Introduction
The next year the crisis in the new technology domain plus the ENRON scandal made us lose our principal industrial sponsor. We did our best to repeat the tele-seminar experience, this time with the IUP Miage of the University of Bordeaux (France). This work was also closely related to the “international e-Mi@ge project”16 and prototyped the future “year project” module of this digital campus.

16 “Campus numérique International e-Mi@ge”, http://www.u-picardie.fr/~cochard/IEM/
The principle was the same, but this time we tried to make improvements, mainly technical ones, based on the conclusions of the previous experience we had with Mannheim. Furthermore, our new sponsors (La Poste, Hewlett Packard) asked us to make students work on collaborative tools, not only with collaborative tools.\footnote{“Projet collaboratif Nice-Bordeaux”, in french, http://miageprojet.unice.fr/twiki/bin/view/Bordeaux/WebHome}

**Collaborative work on collaborative tools**

One project consisted in evaluating the main collaborative tools available at that time, both commercial and public domain. The results of this study can be consulted online.\footnote{S.Lachaux, C.Pastorino and al. « Les outils de travail collaboratifs », in french, http://miageprojet.unice.fr/twiki/bin/view/Bordeaux/OutilsCooperatifs}

Another of the projects consisted in writing a cookbook, in case another teaching program should wish to set up student projects involving collaborative distant work.\footnote{“Tutoriaux pour l’utilisation de produits collaboratifs TWiki, VPN, SSH”, in french, http://miageprojet.unice.fr/twiki/bin/view/Bordeaux/OutilsCooperatifs}

As usual in academic institutions, free, public domain tools are preferred.

**Technical improvements**

*Security problems.* The students addressed a very common problem with university networks: a highly secure configuration prevents students working easily with resources and services located behind the firewall, especially if they are themselves outside of the local network. At best, any computer is only accessible with SSH, and most of the ports used by services like CVS, mail, etc. are masked and not accessible from the outside. For distributed groups of students, this is very annoying. Several solutions have been evaluated, including Virtual Private Networks (VPN, see Pena and Evans 2000) and SSH tunnelling techniques (see Barret and Silverman 2000). We decided that letting students install their own servers on their own personal machines was not an acceptable solution as we would then lose all traceability of the student’s work.

SSH tunnelling seems more flexible and easier to set up than VPNs. The principle consists in sending packets not directly to the target computer, as they will be blocked by the firewall, but to encapsulate them into a tunnel that can go through the firewall, i.e., using a protocol that can go through the firewall. Of course, this operation requires the remote user to be logged onto the secured machine (using ssh), and may need some software configuration on the client machines.

*Automated student account creation.* In order to avoid the repetitive task that consists in creating student accounts by hand for CVS, for TWiki, for SSH access etc., students wrote several scripts and templates for TWiki. With these scripts, groups of student accounts are created automatically from a web page. Students can register by themselves and no further complex administrator commands are needed. Of course students can register only once, and only during the registration period.

**TWiki templates and scripts for automating repetitive tasks**

TWiki’s open architecture makes its customization rather easy in Perl. Figure 3 shows the organization of an installed TWiki.

*Perl scripts* in the bin directory are called to generate TWiki pages on the fly (most of the time, this is done by the *view.pl* script). *The scripts may load TWiki templates.* As an illustration, the *register.tmpl* template may be loaded by the *view.pl* script in order to display the HTML form for user registration. Once the user has entered her personal data and pressed the Ok button, the *register.pl* script is called. It will take care of the parameters the user entered and register the user, then load another template for creating a confirmation page. Scripts are written in Perl and behave like standard CGI-scripts.

\footnote{“Projet collaboratif Nice-Bordeaux”, in french, http://miageprojet.unice.fr/twiki/bin/view/Bordeaux/WebHome}
Templates are just HTML + TWiki variables and macros. The variables can be used in templates for building Web pages. This format separates the presentation from the Perl script and lets anyone who knows HTML modify the page design easily. Consider Listing 1, which shows a simple template used by the view script. TWiki scripts automatically substitute values for placeholder variables. The %TOPIC% variable is replaced with the page name, just as the %TEXT% variable is replaced with the body text. This layout means users can change the look of the page without modifying the Perl source. In this way, each TWiki Web can have its own template files.

Variables are defined within the TWiki pages, using the set instruction. For example, the instruction

```
SET GROUPNAME=Mainline
```

placed in a TWiki page will define a variable named GROUPNAME. In any TWiki page or template on the same Web, it will be possible to use %GROUPNAME%. It will be expanded on the fly when the page or template is called by a script. When one installs TWiki, it comes with a comprehensive set of predefined variables.

Macros like %REVISION% are defined in Plugin code.

```html
<TITLE> TWiki . %WEB% . %TOPIC% </TITLE>
<TBODY bgcolor="#ffffff">
<br>
%TEXT%
<br clear=all>
</BODY></HTML>
```

Listing 1: An example of TWiki template that uses variables and macros

---

20 Variable and macro visibility is the web (a web has a name and represents a set of pages). TWiki proposes also a way to define global variables, which are visible on all webs.
Automatic creation of skeleton pages for each project

Each group of students had to write a set of TWiki pages in the Nice-Mannheim experiment. These included:

- Home page for their project, that contained all important data about their project, emails of the participants, tutors, etc. plus links to other pages described below.
- A page that will describe all the meetings they had, with a brief description of “today’s work”, participants and decided tasks, including its responsible.
- A page dedicated to all the ongoing tasks.
- A page for participants' reports.
- A page for all the resources used (links, references).
- A page with a download area to hold the archives of the software developed, presentations, interesting documents, etc.
- An open discussion page.

During the life of the project, all these pages will evolve, will hold links to other pages the students will create remotely, etc. But they serve as a good starting point for their projects: a skeleton web site organization.

After a group has been registered, scripts are automatically executed to create these pages\(^\text{21}\) default content to explain what we expect on each page, links to TWiki and SSH tutorials (including how to set up tunnelling), etc. These scripts also perform extra operations described below.

**Automatic setting of TWiki access restriction.** At the same time as the skeleton pages are created, access rights are also configured: creation of a group of users for the project (students + tutors), and settings for allowing only this group to edit the project’s TWiki pages (read access is free to everyone).

**Publication of many tutorials.** To introduce the various tools more rapidly, we wrote several tutorials and FAQs\(^\text{22}\).

**Experience**

This second experience involving 24 students (12 from Nice and 12 from Bordeaux) was a real success. The language barrier did not exist this time (French students had some difficulties with the English language during the Nice-Mannheim experiment) and many technical improvements were achieved which will now be leveraged into other projects. Our research team has already used some of the student work for its TWiki-based web site. TWiki and its script/template based architecture is a very flexible tool that proved to be very usable for managing remote student projects. The only sticking points that remain are lack of support for Microsoft tools, and the poor import/export modules that are proposed so far (as third party plugins). Academic institutions may not be overly concerned by Microsoft tool integration, but this is a serious problem for industrial partners. This year we will repeat the tele-seminar with the IUP Miage of Bordeaux, with nearly the same subjects dealing with collaborative tools. But this time, one of our industrial sponsors has politely requested that we test Microsoft SharePoint which proposes (at least on paper) collaborative access to Microsoft Office 2003 components (Word, Excel, etc.).

\(^{21}\) Pages in TWiki are simple .txt files located in the data directory, see Figure 3 TWiki architecture.

\(^{22}\) “Tutoriaux pour l’utilisation de produits collaboratifs TWiki, VPN, SSH”, in french, see also: H. Kopitzki, C.Clément, M.Moreno and al., “Cookbook for setting up TWiki and CVS”, http://miageprojet.unice.fr/twiki/bin/view/Bordeaux/SCRapport
We will keep improving the free public domain tools we have used so far, and in parallel we will be testing Microsoft products.

CONCLUSION

During the first experiment (Nice-Mannheim), it appeared that user training is required. Students only had a one-hour training period but quickly started to learn the tools on the fly. With the technical improvements we have made during the last experiment, and with the tutorials that are today available to our students, we expect that both issues have been addressed.

TWiki appeared to be a very valuable tool for academic and Microsoft-independent Internet collaboration. It comes with an open community of developers and its simple engine allows non-technical people to participate and bring fresh ideas. We constantly discovered new uses for this great tool and as we slowly became power users, we learned how to write scripts to suit our needs. We doubt proprietary tools would provide such a flexibility.

A recent survey by Paquet (2003) about “wikis and knowledge sharing” shows that more and more people are using it - a revolution perhaps?

REFERENCES

  Barrett & Silverman / O'Reilly  2000.
BUFFA (2002). Michel Buffa, “Présentation du logiciel TWiki”, in French, 2002,
  http://miageprojet.unice.fr/twiki/bin/view/TWiki/TWikiHistorique
  SharePoint, PC Magazine online, May, 7, 2002,
  http://www.pcmag.com/print_article/0,3048,a=25166,00.asp
PAQUET (2003). Sébastien Paquet, “Seb's "wikis and knowledge sharing" survey: Results”,
  University of Montréal, January 2003,