WebCHARLIE - A New Open-Source Web-Based Tool Used To Improve Mathematics Skills

Steven Barker

Buffalo State College

Abstract
A new web-based open-source computer-managed homework tool called webCHARLIE is described and compared to four competitors. WebCHARLIE is part of an asynchronous learning network (ALN) that has been used to help engineering technology students improve their mathematics skills. Lessons learned using webCHARLIE over the past six years are also discussed.

Introduction

WebCHARLIE is one component of an asynchronous learning network (ALN) with traditional HTML web pages separately providing the usual course information such as syllabus, bulletin board, tutorials, laboratory guidelines, and homework samples. WebCHARLIE is a tool used to deliver individualized homework and exam exercises, receive students' numerical answers, give instantaneous feedback, encourage students to find and fix their errors, discourage homework procrastination, enforce the college's policy of two hours of homework for each hour in class, and record usage log files which permit numerous assessments. Students learn more when they do homework that is assigned, collected and graded every week. The continuous availability (24/7) of webCHARLIE is a significant feature. WebCHARLIE helps some students with their conflicting goals; they like knowing exactly what is expected of them to succeed, and they also like the feeling of accomplishment after a successful course, but they also give homework a low priority in their busy schedules. It is very simple, do the homework and succeed, or avoid homework and fail. However, homework must not be forced on them. True choice is given to the students throughout the semester by computing two final course grades, one with and one without a homework component. The higher grade becomes the final course grade. Based on final homework scores, most students choose homework.

Using webCHARLIE to improve math skills is the most recent benefit, but that is not the reason it was created. It was created to grade homework given the model that most of our students learn primarily through homework that is assigned, collected, and graded every week.

History

CHARLIE (the predecessor of webCHARLIE) was born in the summer of 1996 after a frustrating year of manually grading homework. It became clear that a homework grader, other

than the instructor, would be required if students were going to learn the abstract course material. CHARLIE (an acronym for Computer-Managed Homework, Assignment and Response, Learning and Instruction Environment) was documented after its first year of use.[14] CHARLIE was developed on the college's mainframe computer, a DEC Alpha computer using the VMS operating system. Students reluctantly learned the VMS operating system commands that allowed them to read email and use CHARLIE. That computer did not permit access through popular internet browsers. Due to the anticipated demise of the DEC Alpha computer, CHARLIE had to be rewritten. A primary goal in the transition from CHARLIE to webCHARLIE was to maintain the large number of exercises (about 700) that had been developed.

During the summer of 2001, CHARLIE evolved into webCHARLIE which runs on common hardware using open source software (Linux, Apache, PHP4, PostgreSQL, C, and Octave). WebCHARLIE has two parts: 1) a homework generator based on C and used only by the teacher to create the homework exercises and insert the answers into the databases, and 2) an internet engine (delivery system) based on Apache, PHP4 and PostgreSQL. C is used in the webCHARLIE homework generator, but not in the webCHARLIE engine. The webCHARLIE engine is a simple extension of HTML web pages through Apache, PHP4, and PostgreSQL. The interaction among these components seems to happen quite easily and naturally. Hard copies of individual homework exercises are delivered to students during class. For the few students who requested extra copies because they could not come to class, or who misplaced their original hard copies, electronic versions were re-generated and posted on a course web page.[13] In addition, sample homework exercises with answers, were also posted.

WebCHARLIE is accessible from the internet using popular browsers. The course information component, which was a part of CHARLIE but not a part of webCHARLIE, is now provided externally through traditional HTML web pages.[13] An 'internet bulletin board' where the instructor posts timely information is very useful. The 'internet grade book' is the students' favorite web page. Some students use email to get help between classes. Email is effective because the instructor is willing to read email seven days a week.

Math-Skill Development

Our engineering technology students have relatively poor mathematics skills. A student once confessed, "I would have gone into engineering if my math skills had been stronger." Since we use a top-down approach in our control system courses (as in engineering), it is challenging when students have trouble isolating a variable in an algebraic equation. Improvement of students' mathematics (primarily algebra) skills has been a topic of special interest for several years.[5]

A simple solution is practice, lots of practice through homework, which many students have apparently avoided when they took algebra, college algebra, trigonometry, calculus I, and calculus II. According to the college catalog, students are expected to spend two hours preparing for each class hour. A careful enforcement of this policy without making the students feel oppressed was the underlying strategy.

Engineering technology students at BSC are required to take Analytical Methods (ENT300) - a required courses that traditionally focuses on differential equations after a short review of algebra and calculus. This author changed the focus to an algebra review with a shorter coverage of calculus and differential equations. After teaching ENT300 two times, it is clear that our students can improve their algebra skills if they have sufficient practice. A pre-test and post-test were used both semesters to measure the math-skill improvement. While not statistically significant due to the small number of students, the outcome is obvious - algebra skills of engineering technology students can be improved through practice. The primary factors that appear to have been effective include: 1) the nature of ENT300, 2) the will of the teacher, 3) the willingness of the students to submit to the huge amount of homework practice, and 4) some tool such as webCHARLIE that continuously monitors progress.

Hardware
The computer used to support webCHARLIE during Fall 2001 with fifty students in two courses was a modest 233 MHz Pentium II processor, with 64 MB of RAM, 4 GB IDE hard drive, 4x CDROM, and 10 Mbps network interface card. The computer was dedicated totally to webCHARLIE because of the uncertainties associated with the new system. However, the load was minimal with about four thousand hits during the semester, and disk requirements were small. The webCHARLIE homework generator for ENT300 required 1.5 MB, the webCHARLIE engine required 327 kB, and the webCHARLIE database for ENT300 (370 exercises, 970 numerical answers for 35 students) required 10 MB (1.7 MB compressed tarball). The internet connection was provided by the college. The modest hardware was more than adequate for the task. The server load approached 100% only when the instructor computed average scores at the end of the semester for the entire class of 35 students, most of whom had each submitted all 970 numerical answers. New homework exercises were added to the database during the semester rather than all at the beginning. Students could use webCHARLIE to compute their individual scores, but not for the whole class.

Software
The software supporting webCHARLIE in 2001 included Red Hat Linux 7.1 which contained the 2.4 kernel, Apache, firewall implemented with IPTABLES, PHP4, PostgreSQL, and Octave. Faculty users of webCHARLIE need not know anything about these software packages, unless changes to the webCHARLIE engine or new homework exercises are desired. All of these components are protected by the Open Source copyright,[12] and were also bundled with Red Hat 7.1 (and subsequently in Red Hat 7.2). Each one is described separately in the following sections. It is anticipated that webCHARLIE will be released as an open source product in the summer of 2002 after being used by another teacher this spring.

About 500 homework exercises have already been tested in webCHARLIE for two courses, Analytical Methods and Control Systems I, with another 200 exercises in two additional courses to be transferred from CHARLIE during the Spring 2002 semester. In addition, development of new exercises for an electronics course, is also planned. The webCHARLIE homework

development system used by the author, consists of the vi editor, C compiler, and basic Linux commands. While the software has no cost in terms of dollars, there is a significant cost in terms of time to develop new homework exercises: problem design, coding, testing, debugging, and figure development.

Linux

Perhaps everyone has heard of Linux by now, but just in case, the Linux Journal[11] says: "Linux, also known as GNU/Linux, is a free, UNIX-like operating system, developed originally for home PCs, but which now runs on a variety of platforms including PowerPC, Macintosh, Amiga, Atari, DEC Alpha, Sun Sparc, ARM, and many others. Linux aims for POSIX compliancy to maintain maximum compatibility with other UNIX-like systems. With millions of users worldwide, Linux is probably the most popular UNIX-like OS in the world."

Significant features of Linux include:

- free UNIX-like operating system,
- continually being developed,
- the recent 2.4 Linux kernel is simply amazing on all counts,
- fast and extensible,
- stability,
- ease of use,
- graphical interface option,
- software applications developed worldwide in addition to commercial developers, and
- natural networking.

After years of experience with Unix, the author enjoys using Linux for almost everything, including the occasional DOC-file translation. It should be possible to use webCHARLIE without learning Linux, but a Linux system administrator would still be required to set it up, perform routine backups, and monitor the security logs. Linux expertise would probably take a few years to build from scratch.

Apache

Apache is the most popular web server in the world.[10] It automatically handles all communications between webCHARLIE and internet users - students or faculty administrators. It requires a small amount of work to set it up - basically reading the installation file and making the suggested entries, plus a few webCHARLIE entries. It is perceived as a robust and thoroughly tested component that needs no attention after turning it on.

PHP4

PHP4 is a server scripting language[8] that seamlessly communicates with Apache to provide the dynamic nature of webCHARLIE web pages. It is more secure and much easier to implement than the older CGI bin techniques. Anyone familiar with HTML and the C programming language, will have a short learning curve to PHP4. Extensive PHP4 documentation, both online

and published, is available and world-wide use of it has exploded over the past few years. The first International PHP Conference occurred in November 2001.

PostgreSQL

PostgreSQL is a database language that is used by webCHARLIE to store correct answers, students' most-recent answers, number of attempts, due dates, and date and time of answer submissions. It also stores student names and passwords, and security logs of internet access. Officially, PostgreSQL is a "sophisticated Object-Relational DBMS, supporting almost all SQL constructs, including subselects, transactions, and user-defined types and functions." PostgreSQL is supported by published books and online documentation.[9] PostgreSQL, (a fourth-generation computer language, compared to FORTRAN, C, Perl, and C++ which are all third-generation languages) was easy for the author to learn and use with no prior SQL experience. PostgreSQL databases are easily accessible from PHP4, and are invisible to the students and faculty who interact with them. The databases are updated by ASCII PostgreSQL command files that are created when new webCHARLIE exercises are generated. A PostgreSQL processor executes the command files to update the databases.

Octave

Octave is not an essential component of webCHARLIE. It was included in webCHARLIE to demonstrate its huge potential, and also as an introduction to MATLAB, which is used in other courses. Octave may be thought of as powerful graphing calculator that can be securely and easily integrated into any dynamic web-based delivery system. Currently Octave is used in webCHARLIE to compute matrix inverses, determinants, and solutions of simultaneous linear equations. Webpage counters registered over a hundred hits from 35 students for each of these Octave functions.

Octave is a mature high-level language for numerical computations, which is functionally similar to MATLAB. It is easily integrated into dynamic web pages through the use of PHP4. The beta version of Octave (2.1.33) contains a very nice suite of control system functions not found in the stable version (2.0.16). Documentation of Octave[6] is thorough and currently has a world-wide following as evidenced by the activity on the Octave listserv.[7]

Student Interface

The student interface to webCHARLIE is easy and convenient through browsers such as Netscape, MS Internet Explorer, AOL, webTV, and Lynx. The student interface is a major advantage of webCHARLIE over CHARLIE. It has almost no learning curve since most students are already familiar with internet browsing. Some students who had experience with CHARLIE commented on how easy it is to use. The grading system discourages procrastination and guessing while encouraging accuracy, thoroughness, and complete answers. Students must also fix their errors in order to earn maximum credit. The grading system, a simulated tour of webCHARLIE, and everything the new webCHARLIE user needs to know is available from webCHARLIE's homepage.[13]

Administrative Interface
The administrative interface differs from the student interface in that it supports name and password changes, new homework generation, exercise due-date changes, access to internet-usage logs, and several statistics gathering features. It is as easy to use as the student interface. It is similar in appearance to the student interface and uses the same password-protection system. For extra security, it uses a second password that is managed by the Apache webserver. For more security, it uses an access log that is different from the student's access log. System administration, after developing homework exercises and after setting up the webCHARLIE engine and databases, does not require knowledge of Linux, C, PHP, postgreSQL, or Octave. However, making routine backups requires some knowledge of Linux. The first trial use by another faculty member is planned for the 2002 Spring semester.

Security
Protection of the server, hardware and software, cannot be over emphasized. Only authorized webCHARLIE users are allowed to use webCHARLIE. The server is totally dedicated to webCHARLIE. Students must not see the webCHARLIE databases, although their answers are written into the databases under the control of webCHARLIE. Faculty administrators need to see the databases occasionally, and also need to compile usage statistics. Only the teacher should compute the overall course average because it is the most resource-intensive operation.

Security measures include a firewall (implemented using IPTABLES) that restricts access to the computer, a password system for the students, and a double password system for faculty administrators.

The firewall, implemented with IPTABLES, is a major source of security. Setting up the firewall was very easy. Unwanted probing, which happens every day, is repelled by the firewall. IPTABLE packet statistics showed that hundreds of millions of packets were blocked by the firewall during the semester. The Linux tool/command (tcpdump) was also used to confirm proper operation of the firewall. WebCHARLIE also has a logging component that records IP address, dates, times, browser, usernames, and passwords of all webCHARLIE users.

Initial setup of the computer (almost any old computer will probably work), operating system (Linux), web server (Apache), and other software packages (PHP, postgreSQL, C, and Octave) requires an experienced administrator. Backups are also the responsibility of the computer administrator. Weekly backups using tarballs were simply copied to another computer through the internet using secure-shell copy. Telnet, FTP, and mail features were turned off. SSH provided the equivalent telnet and FTP functions. Total computer administration was provided by the author during its first semester of use. Changes to the webCHARLIE engine or to the database structure should not be attempted after the course begins.

In the event of a server crash such as a disk failure, a new disk (or new computer) could be installed within a few hours along with all new software including the webCHARLIE engine and

databases from the last backup. There was no notable security event or hardware failure during the first semester.

Choosing A Homework System

Several homework systems that are conceptually similar to webCHARLIE are available: CAPA, WeBWorK, Mallard, and CyberProf. Although this author has never used any of these systems, summaries based on online documentation are provided in the following sections.

All of these systems including webCHARLIE have several common elements including individualized numerical exercises for homework, quizzes, exams, and a grade book. They provide immediate student feedback, and may be considered a primary component of a larger ALN system. They report increased student learning, success, and acceptance.

Before summarizing each of these competitive systems, some features of webCHARLIE are listed. webCHARLIE is simple and small in size. The server is stable and secure. The operating memory requirements are small. Maintenance is minimal. It has an easy-to-use graphical interface. I has an efficient exercise-development system. It has the potential of addressing difficult pedagogical issues such as thinking-skill development. It has a high student-acceptance rate. It has a guaranteed low cost (zero purchase cost) because of the open-source components, which are extensively documented. It can run on inexpensive low-end hardware. Individual exercises are developed in C. The instructor of a course for which webCHARLIE exercises have already been developed does not need to know about the full software suite (Linux, Apache, C, PHP, postgreSQL, or Octave). New-course development and total administration requires knowledge of the full software suite. One shortfall may be the numerical answers without units, and no hints for wrong answers. Having used webCHARLIE (and CHARLIE) for six years, this instructor would not like to teach without it.

The following summaries of completive systems were derived from online documentation.

CAPA

Key features from CAPA's online documentation[1] include personalized assignments, quizzes, and examinations, which encourage students to discuss concepts. Problems for each student differ sufficiently to inhibit rote copying. There are a large variety of conceptual questions and quantitative problems. Students are given instant feedback and relevant hints via the internet. Students may correct errors without penalty prior to an assignment's due date. CAPA keeps track of students' participation and performance. Records are available in real time both to the instructor and to the individual student. Statistical tools and graphical displays facilitate assessment and course administration in addition to extensive documentation. CAPA was developed at Michigan State University, and was first used in a small (92 student) physics class in the Fall of 1992. Since then, it has been used by more than 100,000 students in astronomy, biochemistry, chemistry, mathematics, physics, botany, accelerator physics, human food and nutrition, family and child ecology, and computer science courses. Today it is used at many different universities.

One report about the use of CAPA[1] shows a strong statistically-significant positive relationship between success on homework and success on the final exam through a two-year project. An important outcome was that this network technology can significantly improve student achievement in large classes. CAPA helped students overcome some of the factors that contribute to students not achieving their goals, which include deficient preparation and lack of awareness, misconceptions about physics fundamentals, insufficient mathematical problem-solving skills, excessively demanding and difficult course schedules, students' perceptions of the quality of education, and emotional and physical well being (students who work to pay bills and students who fall behind).

CAPA requires a UNIX operating system, and has been exported to several different UNIXes including Linux. CAPA has two student interfaces: 1) internet web browser, and 2) telnet. The telnet interface requires a large server swap space (5 MB) per telnet session. The CAPA shell is free of charge via the General Public License (GNU). All demonstration problems, libraries, homework sets, are not covered under the GNU license. At least two CAPA libraries are available: a physics library for $300 and an astronomy library for $100.

WeBWorK

Key features from WeBWorK's online documentation[2] include an internet-based method for delivering homework problems, instant feedback on homework answers, individualized problem sets, no penalty for wrong answers before the due date, and homework assignments that are available as a typeset, downloadable postscript file. WeBWorK is based on CAPA, differing mainly in its use of a new generation of electronic technology. WeBWorK was developed at the University of Rochester, and was used initially in pre-calculus and calculus. WeBWorK is accessed using common internet browsers and requires a Unix server. It uses email to provide personal help.

The online documentation contains a comparison between CAPA and WeBWorK. The documentation suggests that the CAPA engine is a self-contained C program that requires internal changes for future improvements. In contrast, "WeBWorK knits together existing web programs, leveraging the work done by the programming community surrounding the internet, in order to accomplish the same goals as CAPA." Since homework-problem development requires so much effort, the WeBWorK approach appears to offer an advantage: WeBWorK uses Perl while CAPA uses a restricted set of CAPA functions. WeBWorK has used "commonly available web programs" to "exceed CAPA's capabilities" using a staff of four people in less than two years.

WeBWorK is free and has been used at several different colleges. They are currently (January 2002) getting feedback from others who use WeBWorK. WeBWorK currently runs on UNIX (including Linux) and also on NT and Mac servers.

Mallard

Key features from Mallard's online documentation[3] include a web based secure asynchronous instructional environment, a powerful, flexible, reliable, and easy-to-use tool that can enhance teaching efficiency and effectiveness. Mallard is equally useful for an (otherwise) traditionally structured course as for distance education. It can be used in virtually any subject. Mallard has a builtin feedback system, that provides extensive hints and immediate feedback. Online assistance is part of Mallard. Questions can be either randomly generated (if appropriate) or randomly selected. Mallard was developed at the University of Illinois at Urbana-Champaign (UIUC), and it has been used in 50 courses at UIUC in subjects that include aviation, engineering, foreign language, business, agriculture, and many more. Currently it is used in several other schools, including elementary schools.

Mallard is not free and the Mallard engine cannot be modified. The annual subscription fee ranges from $100 to $5000 based on number of users and number of servers.

The Mallard server is a UNIX computer (Linux was not listed). Several web servers are possible. It requires at least Perl 5.002 and Revision Control System (RCS).

CyberProf

Key features from CyberProf's online documentation[4] include interactive homework and quizzes, electronic bulletin board for communicating with peers and teaching assistants. The online grade book is used by course directors, instructors, and students. There are online lecture notes. CyberProf uses Perl and Java to develop animations, images, graphs, and equations as well as prose text. It uses an HTML Editor to create CyberProf problems. Two hints per exercise are provided in the form of simple prose text or a series of further questions. CyberProf evaluates strengths and weaknesses of students' knowledge and skills which permits students to direct their own learning. It uses special 'grading packages' for chemistry and computer science. Cyberprof was developed at the University of Illinois at Urbana-Champaign (UIUC), and has be used in Physics, Chemistry, Biology, Economics, HS Electronics and AP Physics. It has been used in Physics and Engineering at Purdue, Rutgers, Virginia Tech, and others. HTML, Java, and Perl are the primary development languages along with special CyberProf HTML-like tags. CyberProf includes a 'File Browser' (os.exe) for developing exercises that can be run on their server, and also includes online documentation pages that itemize the individual steps for creating CyberProf exercises.

The Beta test version of CyberProf was free in December 2001. Server requirements were not obvious, but they can be contacted for more information.

Lessons Learned

Collaboration among students is generally encouraged, but a problem has been noticed recently. Students who think that the main purpose is to feed webCHARLIE correct answers are tempted to search for 'easy' ways to get those answers. Some students learn from other students that all they have to do is to combine a few numbers in a certain order and the correct answer appears. Those students avoid learning the fundamental concepts, and probably expend more energy in

the process. This may explain how they were able to pass calculus without ever acquiring the prerequisite algebra skills.

WebCHARLIE is more manageable than CHARLIE because it only deals with homework and exams. The course-information part is handled separately through HTML web pages.

Most students prefer take-home webCHARLIE exams to in-class exams, presumably to avoid the stress typically associated with exams. For the past two semesters, take-home exams have been given that are three times the length of in-class exams (thirty exercises instead of ten). This strategy supports the course mission: mathematics-skill development through practice.

Deficiencies of webCHARLIE include a lack of 'smart' multiple choice answers, no automatic system of providing hints, and students need not provide units for their numerical answers. Students are asked to provide numerical answers that have specified units, because unit analysis has not been built into webCHARLIE. A 'smart' multiple choice feature is also desirable, where several correct answers would be randomly placed in the list, and a few wrong answers would be randomly selected from a much larger list. CHARLIE provided some hints such as significant figures, sign errors, and order of answers. These minor hints could easily be written into webCHARLIE, but a more sophisticated system would be better, such as tutorials on significant figures and hints requiring use of the textbook.

Type-setting output is needed in order to replace ASCII format (e.g. 3x^3+2x^2+x=9 is not the same as 3x^{3+2x^2+x}=9). In the words of one student who complained about this format; "I understand order-of-operator precedence, but this type of mistake is not my fault." Our students are discovering that their powerful calculators can solve almost anything, but they still need to understand the fundamentals of mathematics in order to formulate a correct input and interpret the output.

Faculty use of webCHARLIE occurs through an unsecured link. Perhaps a secure link should be added - probably through the existing PHP4 capabilities. We will pay close attention to the usage logs next semester to assess this situation.

Course evaluations at the end of the semester were favorable. There was no student revolt due the large amount of homework, although there was some 'groaning' throughout the semester. Some students appreciate a tough but fair course policy, and hopefully, they will benefit from their improved math skills. WebCHARLIE permits the instructor to be more of a 'learning facilitator' than 'teacher/lecturer.'

WebCHARLIE homework provides a level of accountability not typically inherent in traditional homework. This is particularly important for our students who, at this stage in their development, view homework as a chore rather than an opportunity to acquire useful skills.

WebCHARLIE was originally perceived as unsuitable for thinking-skill development. WebCHARLIE was perceived as primarily a homework enforcer that gave immediate feedback while simultaneously encouraging students to find and fix their mistakes. However, after observing how thinking skills are developed through CAPA, it is now believed that webCHARLIE can do the same, not by any new feature to the webCHARLIE engine, but through the design of each homework exercise. It follows that since homework-exercise development is the hardest part of webCHARLIE (most costly in terms of time and effort), adding a thinking-skill focus to each homework exercise would significantly increase development efforts. Several examples of exercises that promote thinking skills are provided in the CAPA documentation.[1]

Summary

This paper describes a new web-based open-source computer-managed homework tool called webCHARLIE. Four of webCHARLIE's competitors are also described. WebCHARLIE has been used to improve mathematics skills of engineering technology students, and it appears that it also has the potential of addressing other tough pedagogical issues such as thinking-skill development. This paper also focuses on the hardware and software that was used by webCHARLIE during its first semester during the Fall 2001 semester.

Bibliography

1. Kashy, E., Albertelli, G., Thoennessen, M., Tsai, Y., Morrissey, D., "CAPA," http://capa4.lite.msu.edu/homepage/
2. Arnie Pizer, A. and Gage, M, "WebWork," http://webwork.math.rochester.edu/
3. Brown, D.J., "Mallard, Asynchronous Learning on the Web," http://www.cen.uiuc.edu/Mallard/
4. Hubler, A, "CyberProf," http://www.howhy.com/home/
5. Barker, D., "Mathematics Skills Improvement in Engineering Technology Students," http://tech.buffalostate.edu/~barker/publications/asee01/index.html
6. Eaton, J.W., "Octave," http://www.octave.org/
7. Octave Mailing List, http://www.octave.org/help.html
8. PHP, http://www.php.net/
9. PostgreSQL, http://www.jp.postgresql.org/
10. Apache Web Server, http://www.apache.org/
11. Linux Journal, http://www.linuxjournal.com/
12. OSI http://www.opensource.org/
13. WebCHARLIE's Homepage, http://tech.buffalostate.edu
14. Barker, D., "CHARLIE: A Computer-Managed Homework, Assignment and Response, Learning and Instruction Environment," http://fie.engrng.pitt.edu/fie97/papers/1088.pdf

Biography

STEVEN BARKER is an Associate Professor in the Technology Department at the State University of New York College at Buffalo (Buffalo State College) who teaches electrical engineering technology courses. He teaches the control system and PLC courses, and develops educational tools, such as webCHARLIE. He is a member of IEEE and ASEE.