DOCUMENT RESUME

ED 427 743 IR 019 294

AUTHOR Vantaggiato, Antonio
TITLE Automatic Exams and Course Administration on the Web.
PUB DATE 1998-11-00
NOTE 8p.; In: WebNet 98 World Conference of the WWW, Internet and Intranet Proceedings (3rd, Orlando, FL, November 7-12, 1998); see IR 019 231. Figures may not reproduce clearly.
PUB TYPE Reports - Descriptive (141) -- Speeches/Meeting Papers (150)
EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS *Computer Assisted Instruction; *Computer Assisted Testing; Computer Software Development; Courseware; *Distance Education; Expert Systems; Higher Education; Instructional Design; Instructional Materials; Material Development; Online Systems; Programming; *Test Construction; World Wide Web

ABSTRACT

This paper describes a system that provides faculty with automatic testing and grading online facilities, as well as course administration resources, using the World Wide Web as a common and ubiquitous interface. The system was built using the Common Lisp HTTP Server from MIT and its functionality to generate HTML "on the fly," which offers the appropriate level of interactivity and flexibility needed to design such systems. It allows instructors to create exams and to manage them online, and it includes automatic grading and immediate feedback to students. Also, the system allows for automated scores bookkeeping: students may check their grades at any time and so may their instructor. It is argued that the system might be a useful framework for enhancing Web-centered education, and that the CL-HTTP server is an ideal instrument for easily achieving this sort of interactive and automatic management. (Author/AEF)

Automatic Exams and Course Administration on the Web

Antonio Vantaggiato

Laboratory of Artificial Intelligence
Sacred Heart University, PO BOX 12383 San Juan, Puerto Rico 00914
Email: a_vantaggiat@uscac1.sagrado.edu

Abstract

This paper describes a system that provides faculty with automatic testing and grading on-line facilities, as well as course administration resources, using the World Wide Web as a common and ubiquitous interface. The system was built using the Common Lisp HTTP Server from MIT and its functionality to generate HTML on the fly, which offers the appropriate level of interactivity and flexibility needed to design such systems. It allows instructors to create exams and to manage them on-line, and it includes automatic grading and immediate feedback to students. Also, the system allows for automated scores book-keeping: students may check their grades at any time and so may do their instructor. After presenting the system, the conclusion of this paper argues that it might be a useful framework for enhancing Web-centered education. Also, it argues that the CL-HTTP server is an ideal instrument to easily achieve this sort of interactive and automatic management.

Keywords: Web Based Training, Courseware Development, Distance Education, Server Technology, Common Lisp.

1. Introduction

This paper describes a system that aims to provide faculty with automatic testing and grading on-line facilities, as well as course administration resources, using the World Wide Web as a common and ubiquitous interface. The system was built using the *Common Lisp HTTP Server* from MIT [Mallery 1994] (http://www.ai.mit.edu/projects/iiip/doc/cl-http/home-page.html) and its functionality to generate HTML on the fly. It focuses on allowing instructors to prepare and manage on-line exams, and to manage students' grades. The World Wide Web is being used in educational environments as both a medium for unprecedented communications opportunities and information sharing across geographical limitations, and as a substrate to enhance (or to reinvent) the pedagogical process of learning [Trentin 1996], [Brusilovsky et al. 1996]. Educational applications on the Web often need some management apparatus to better achieve their goals. Specifically, this work has been focused on the design of an instructors' framework for course development over the Web. While some interesting work exists on Web-based course and exam administration [Byrnes et al. 1994], [Gibson et al. 1995], [Goldberg et al. 1996], systems in this area appear to need that some task be completed off-line by the instructor or webmaster. The basic tenet of this work, on the other hand, has been to allow full on-line access to both exam creation and management.

The system described here is in fact able to help instructors with the on-line creation of multiple on-line exams and their automatic management, including immediate feedback to students. Its goals included allowing

instructors to also access students' grades and manage course pages. While this is still work in progress, the main goals have been achieved, and a prototype system is now finished and being tested.

After presenting the system in the main sections of this paper, its conclusion will argue that this might be a useful framework to enhance Web-centered education, and that the CL-HTTP server shows to be a formidable tool for the development of advanced interactive Web pages.

The basic characteristics of the system are:

- It is server-based. It appears critical, for security and other reasons, that an exam administrator does not reside on the client.
- It maintains a global status (as a list of objects) which is saved on the server.

Each connection to the exam server (from a student's client) opens a thread, so the system can easily keep track of each student's separate exam session. On each form through which a student submits his answers there is also a hidden field to carry local status information used by CL-HTTP to avoid collisions among threads.

A student's or instructor's status is kept by parameter passing protocols and by lambda (anonymous) functions, so no cookies are used (but could be). Student and instructor data are retrieved from persistent objects after authentication.

The system was developed using the CL-HTTP Server software, a full-fledged http server implemented and *immersed* in Common Lisp and CLOS (the Common Lisp Object System). CL-HTTP offers functionalities that were considered essential for the development of the system, namely:

- It is a full http server implemented as an extension of the Lisp language, thus offering great extensibility and flexibility by means of open object-oriented programming in Lisp.
- It contains a set of language extensions useful to generate HTML code. Also, the HTML authoring operators support higher levels of abstraction than pure HTML (enumerating items, for instance.) HTML language standards are transparent.
- It allows Java and JavaScript to be integrated into Lisp code.
- It is robust, and is being used extensively in AI and education research (for instance, in the InterBook project of P. Brusilovsky [Brusilovsky et al. 1996], government (The White House Publications System), and corporate intranet development [Davies and Davies 1997];
- It is freely available (including source code) and runs on a variety of platforms.

2. The on-line Exam Creation & Management System

2.1 Description

The system's main goals are:

- To provide instructors with a simple and easy on-line interface to create and manage on-line multiple-choice and subjective exams, with such features as automatic grading, book-keeping, and feedback to students;
- To let instructors and students check grades and class' statistics, both numerically and graphically (see [Fig. 4] and [Fig. 5]).

Moreover, the former goals are part of a more comprehensive objective:

- To provide instructors with an on-line framework to help them build and manage course material based on the Web. In order to achieve this, work has still to be done to:
    - Include tools to let instructors who are inexperienced with HTML to build, publish and maintain simple courses' homepages;
    - Edit already-made exams.

In this section the system's architecture will be explained, as well as its main functions. The On-line Exam Creation & Management System starts with a Control Page that is personal to each registered instructor. In it, all courses that have been set up by the instructor are shown, together with every exam that he made available on-line. This is achieved by retrieving the system's global status, which is a list of all active courses. Each course in the list is a composite CLOS object, as shown in class taxonomy of [Fig. 1].

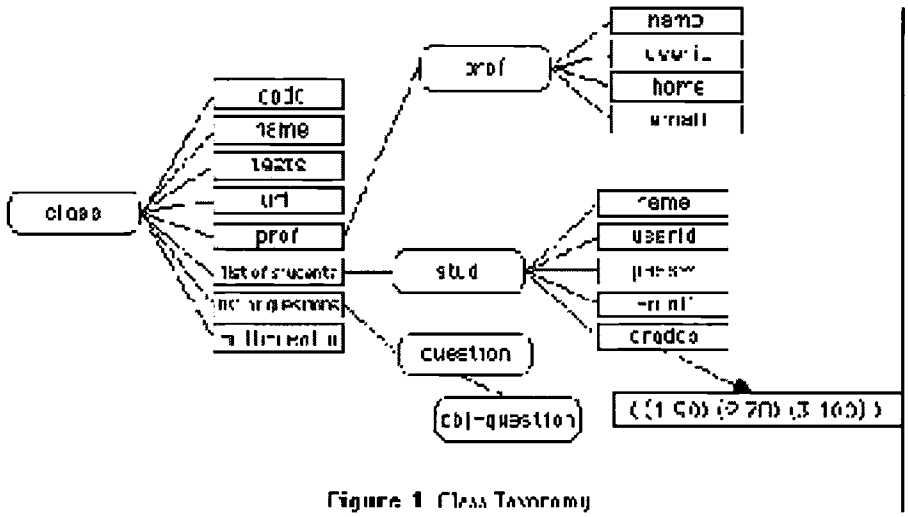

Figure 1 Class Taxonomy

In the main page the available choices are displayed by means of hyperlinks. [Fig. 2] displays the system's main page after a series of visits by an instructor. Through these links an instructor may:

- Create a new class, by defining -among other things- the students who attend the course, and assigning each of them a username and password. Sensitive data is not currently very secure, but CL-HTTP supports the MD5 digest method, so security should be easily upgraded.

- Prepare a new exam for a given course. This opens a form to enter questions (multiple-choice or subjective) and their correct answers.

- Check the exams that are already set. It opens a dynamic page which displays links towards existing exams' pages (owned by a specific instructor).

- Check or edit students' grades for a specific course. The instructor enters a course code and the server fetches her a table with students' grades and averages, built on the fly.

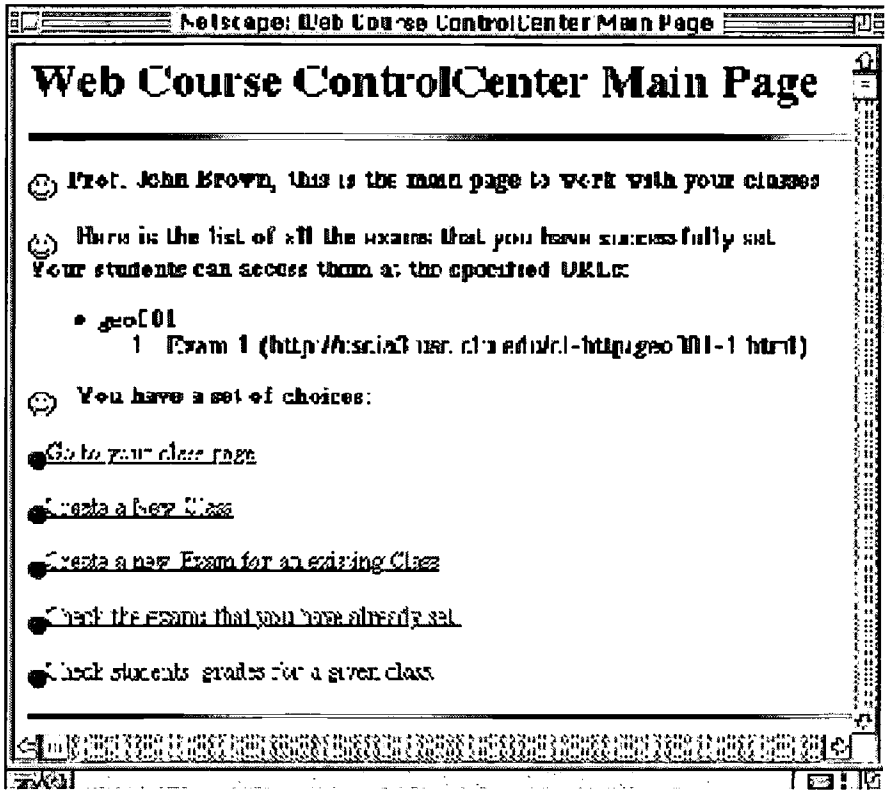

Figure 2 Main Page

2.2 Design & Implementation

The system's architecture, shown in [Fig. 3] (only the first three modules are actually displayed), is modular and simple. This means that it is easy to add new modules (exam editing, for instance) and to modify existing ones. In fact, one of the advantages of the chosen server architecture and Lisp/CLOS implementation, is that several new language constructs were defined which may be used directly by a designer who wishes to add or modify something in the system.

In [Fig. 3], boxes labeled *computed url* represent pages that are computed on the fly (and that depend on user input and global status), and that are *exported* (i.e., in CL-HTTP's terminology, the server is made aware of them), with CL-HTTP's method export-url. These urls have each two functions: a *form function* to generate the form associated with the page, and a *response function* that processes the form's results. In both cases a *lambda expression* is used (an anonymous function) to define an inlined function. Through these functions, parameters are passed to the form generator and the response functions, as a means of preserving the status between form generation and response, and to let them know what class, test number and set of questions the exam refers to.

As an example, an extract of code for `respond-to-exam-main` is shown next.

```
(defmethod respond-to-exam-main ((url url:http-form) stream query-alist)
<initialization...>
;; First associate local variables to form results (in association list query-
alist)
(bind-query-values (choices CLASS-CODE TESTNUMBER OBJECTIVE
                    ANSWERS SUBJECTIVE EXPLANATION computed-choices)
                   (url query-alist)
```

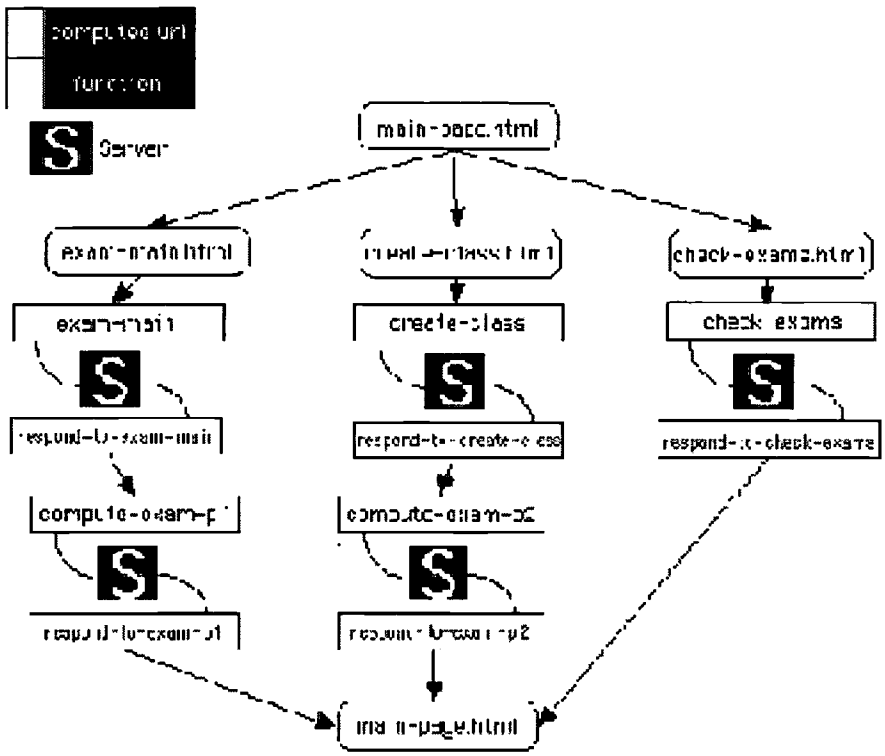

Figure 3: Exam Server's architecture

```
;; Then get the status information.
 (let* ((real-choices choices)
        (*computed-choices*   (read-from-armor-plated-string computed-choices))
        (classcode CLASS-CODE)
        (clas (get-class classcode))
        (data (collect-data-from-query-alist query-alist)
         (user-obj (current-user-object))) ;; gets info on authent'ed user
  (update-class classcode testnumber) ;;;; SAVE UPDATED CLASS!!!
;; Then proceed to next step, by setting up a new form. The two
;; lambda expressions are used to pass status parameters.
 (http:redirect-request *server*
      (export-url #u"/cl-http/exam-step1.html"
      :html-computed-form
      :form-function #'(lambda (url stream)
                         (compute-exam-p1 url stream data user-obj))
      :expiration '(:no-expiration-header)
      :response-function #'(lambda (url stream query-alist)
              (respond-to-exam-p1 url stream query-alist data user-obj))
      :private t :language :en :keywords '(:cl-http :webnet98)
      :documentation "Exam Administration.")))))
```

After collecting the data posted by the form, getting status information on the current authenticated user, and retrieving the course object (clas, which is immediately updated with the new information regarding the exam), the former method simply forces the server to issue a redirect-request to a new page, which is defined and exported on the spot.

The new page is defined with the CL-HTTP's primitive method `export-url`, which is passed all status information via parameters `query-alist`, `data` and `user-obj`. These are fetched to the new page's form

generation and response methods `compute-exam-p1` and `respond-to-exam-p1`, respectively through two lambda functions. Similar implementation has been used for the other methods.

These modules of the system allow an instructor to create an on-line exam, which is then administered by a separate module that retrieves the questions related to it, prepares an exam HTML page, and fetches it to the student's client. When the exam is created, a new url is exported, with its form generation and response methods. When the exam is to be served, these latter methods are activated, so that the exam gets fetched to the student and then graded. Immediate feedback is given to the student (by means of a reply page with his results, and an e-mail message with the same information), and to the instructor (with an e-mail message). Also, the student object is updated with the new grade and saved. A student who already took a certain exam is not allowed to repeat it.

[Fig. 4] and [Fig. 5] show examples of grades output in text and bar graph formats.

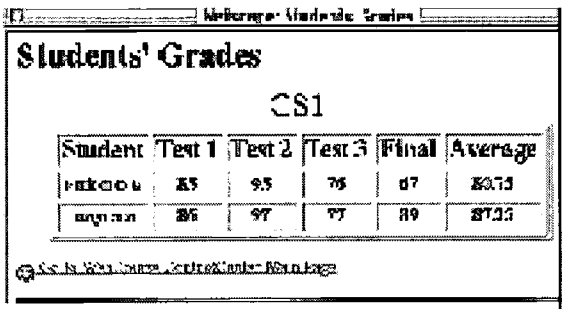

Figure 4: Course Grades

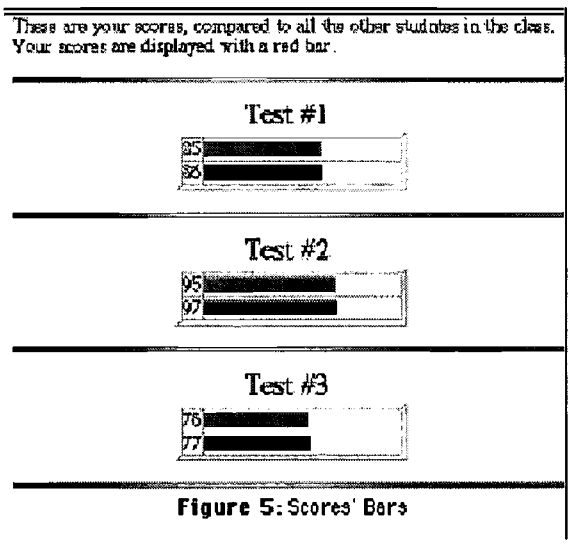

Figure 5: Scores' Bars

3. Conclusions and Future Work

In this paper two systems were presented that aim to enhance pedagogical experiences with Web-based tools. A big research effort has gone into the general theme of the World Wide Web and Education, and many researchers have produced very interesting Web systems that help enhance didactic experiences and help instructors design Web-based course contents [Goldberg et al. 1996], [Byrnes et al. 1994]. This work's aim is directed at assisting faculty in the management of on-line exams, and the book-keeping of students' evaluations, activities full of promise in both distance and traditional courses. This conclusion argues that the system is successful in that the design, tools and programming methodology all concurred to produce an effective (albeit prototypical) system, that is now being tested. Work is still needed to achieve the goal of a system fully able to support:

- Exam's questions editing;
- Instructor's design and publication of course-related Web pages.
- From the design point of view, work also needs to be done to:
- Verify the possible need to pass some computations to a client, in order to lessen the burden over the server;
- Use a more effective and efficient way to have persistent objects: A connection to an Object Database System might prove essential.

4. References

[Brusilovsky et al. 1996] Brusilovsky, P., Schwarz, E., & Weber, G. (1996), A Tool for Developing Adaptive Electronic Textbooks on WWW, WebNet 1996, Association for the Advancement of Computing in Education, Charlottesville, VA.

[Byrnes et al. 1994] Byrnes, R., Debreceny, R., & Gilmour, P (1995), The Development of a Multiple-Choice and True-False Testing Environment on the Web, *1st Australian World Wide Web Conference*, 1994, Ballina (Australia).

[Davies and Davies 1997] Davies, B, and Davies. V.B. (1997), Patching onto the Web: Common Lisp Hypermedia for the Intranet. *Communications of the ACM* 40 (5), May 1997.

[Gibson et al. 1995] Gibson, E., Brewer, P., Dholakia, A., Vouk, M., & Bitzer, D. (1995), A Comparative Analysis of Web-Based Testing and Evaluation Systems, *4th International WWW Conference,* 1995, Boston, MA.

[Goldberg et al. 1996] Goldberg, M.W., Salari, S. and Swoboda P. (1996), World Wide Web - Course tool: An environment for building WWW-based courses, *5th International WWW Conference,* 1996, Paris; Computer Networks and ISDN Systems 28 (7-11).

[Mallery 1994] Mallery, J.C. (1994), A Common Lisp Hypermedia Server, *First International WWW Conference,* 1994, Geneva.

[Trentin 1996] Trentin, G. (1996), Internet: Does It Really Bring Added Value to Education?, *International Journal of Educational Telecommunications 2(2/3).*

Aknowledgements

The Laboratory of Artificial Intelligence at Sacred Heart University is sponsored by the Johnson & Johnson Familiy of Companies.

U.S. Department of Education
Office of Educational Research and Improvement (OERI)
National Library of Education (NLE)
Educational Resources Information Center (ERIC)

NOTICE

REPRODUCTION BASIS

☒ This document is covered by a signed "Reproduction Release (Blanket) form (on file within the ERIC system), encompassing all or classes of documents from its source organization and, therefore, does not require a "Specific Document" Release form.

☐ This document is Federally-funded, or carries its own permission to reproduce, or is otherwise in the public domain and, therefore, may be reproduced by ERIC without a signed Reproduction Release form (either "Specific Document" or "Blanket").