Designing A Web Application System For Paper Presentations and Evaluations (P.P.E)

As A Blackboard Architecture Project

Mohammed Assiri
Department of Computing and Software
McMaster University
assirims@mcmaster.ca

ABSTRACT

A web-based application is proposed to evaluate papers' presentations and reviews. The designing of the proposed application is based on the blackboard architecture. The system's requirements and description are explained, and the results of system's test are reported.

Keywords

Blackboard architecture; web-based application; evaluation system; PHP; MySQLi

1. INTRODUCTION

Software Design aims generally to reduce the complexity of software via decomposing its functionality, aspects, and environment into manageable parts. One of the initial phase in software design is the requirements specification (i.e. the high-level design), which is very formal. The requirements can be functional (specifying what the system must do), non-functional (e.g. legal and security requirements), and process (e.g. costs and risks).

High level of software design must include software architecture that address (partially or totally) some of the desired attributes based on the requirement specification. Each architecture re-presents, for instance, the elements' arrangements, connections, and control.

2. Blackboard Architecture:

Blackboard architecture ([1], [2], [3], [4] and [8]) is one of two categories of data centered architecture, where the software components are communicated via data store, which provides all mechanisms such as insertion, deletion, update, and retrieval. In blackboard architecture, the data store is active while its clients (i.e. knowledge sources) are passive.

The blackboard architecture (Figure 1) named from the classroom teaching and learning where each students and teacher holds a role of agent to solve a problem either parallel or independently. The blackboard architecture is decomposed into two major sub-systems; namely blackboard, which stores data, and knowledge sources, where domain-specific knowledge is stored. Nevertheless, another optionally sub-system is called controller (or also control), which initiates the blackboard and knowledge sources.

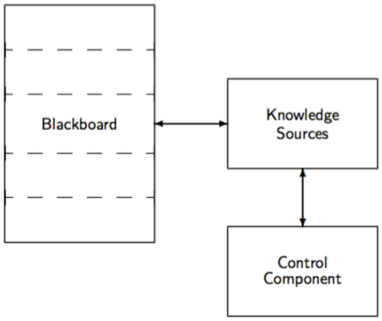

Figure 1: Components of Blackboard

3. System Requirements

3.1 Functional Requirements

- The application must allow the instructor to make changes to the processes.
- The application shall provide average scores and generate reports.
- The application shall be independent from individual students (i.e. absent students have no effect on the processes).
- The application shall allow only registered students to participate.
- The application shall allow selected students to access only selected type of data (e.g. a student can only see his/her received scores and feedback).
- The application shall be available only at selected times by the instructor.

3.2 Non-functional Requirements

3.2.1 Usability Requirements

- The application shall be easy for both instructors and students to use.
- The application shall be easy for both instructors and students to learn.

3.2.2 Performance Requirements

- The application shall have a maximum response time of three seconds for any process.
- The application must be accurate to calculate average scores.

3.2.3 Reliability and Availability

- The application shall be only available for the specified time by the instructor for some tasks.

3.2.4 Operational Requirements

- The application must be able to interact with any HTML browser.

3.2.5 Security Requirements

- Only the instructor can access and change all data.
- Only selected students can see selected data.

4. Paper Presentations and Evaluations System

Adopting the blackboard architecture is unusual enough for a web application [5]. However, the works in ([6] and [7]) were inspired to proposed a new web application called **Paper Presentations and Evaluations (P.P.E)**, which is based on the blackboard architecture.

The proposed system, (P.P.E.), is developed for serving some courses, such as "CAS 703", where students are trained to present and review some scientific papers. This mission is accomplished via (P.P.E) by two blackboard-based systems, namely student blackboard system and instructor blackboard system. The description and the implementation of these systems are given as follows:

4.1 Student Blackboard System

The student blackboard system (Figure 2) is composed of one control, one blackboard component, one database, and five agents.

4.1.1 Control

The students should enter into the system through the log-in system (i.e. control).

4.1.2 Blackboard

The student blackboard allows the students to react with different agents. Furthermore, the student blackboard retrieves data from the database and provides each agent with the appropriate data. Also, the student blackboard stores the processed data from agents into the database. Therefore, the agents have no direct connection to the database.

4.1.3 Paper Selection Agent

After the instructor has provided a reading list, the students are allowed to selects papers from this reading list via the paper selection agent, which requests data from the blackboard, process it, then returns the data to the blackboard to be stored into the database.

4.1.4 Presentations Evaluations and Report Agents

After the instructor assigns a paper for each student to present timely during a class, the audience (both the instructor and students) should evaluate both the delivered presentation and the presenter's performance through the presentations evaluations agent. Then, the presenter will eventually receive a scores and may also receive feedback upon completion of his/her presentation through the presentations report agent. Both agents receive and return data via the blackboard (i.e. the agents have no direct connection to the database).

4.1.5 Reviews Submission and Reports Agents

After the instructor assigned some papers for each student to review timely, each student should review papers by guidance of a giving review template through the reviews submission agent, which receive and return data to the blackboard with no direct connection to the database. Then, the students can review their reviews reports through the reviews reports agents after the instructor had evaluated and graded the students' reviews.

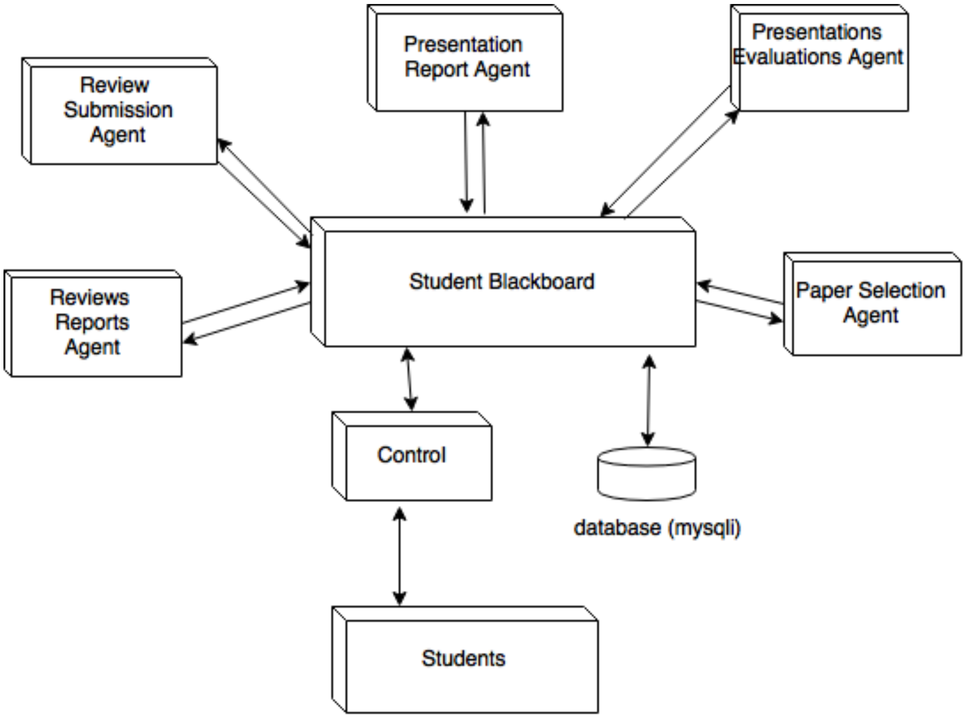

Figure 2: Student Blackboard

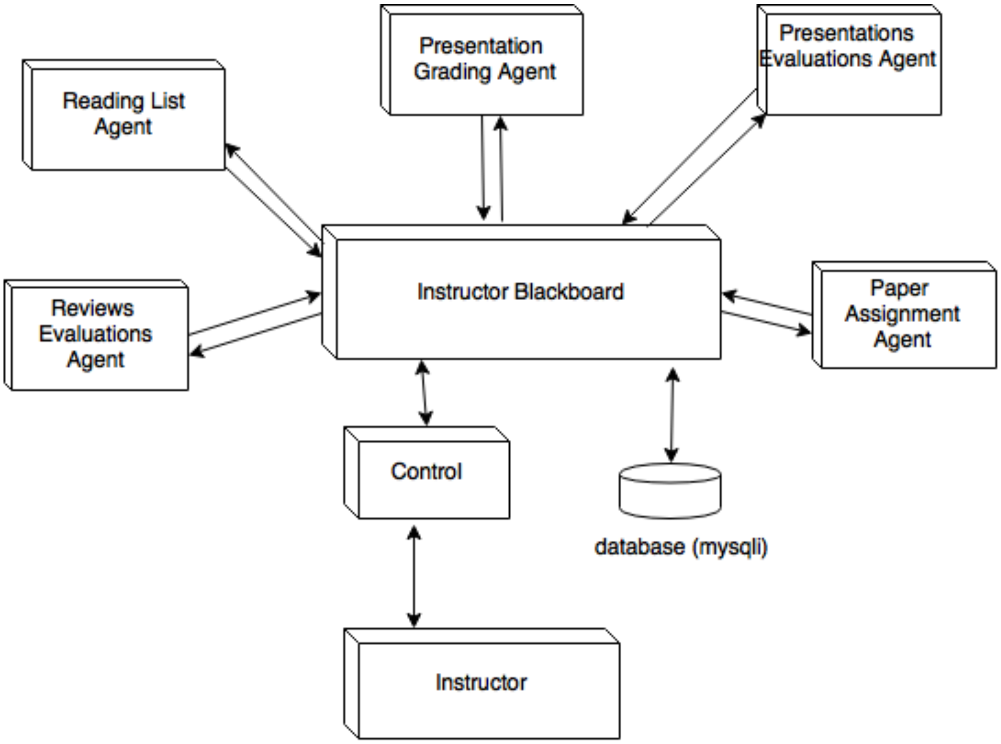

Figure 3: Instructor Blackboard

4.2 Instructions Blackboard System

Similar to the student blackboard system, the instructor blackboard system (Figure 3) is composed of one control, one blackboard component, one database, and five agents.

4.2.1 Control

The instructor enters into the system using the log-in system (i.e. control).

4.2.2 Blackboard

In the instructor blackboard, the instructor can request agents. As well, data from the database to agent are retrieved and stored via the blackboard.

4.2.3 Paper Assignment Agent

The instructor can assign papers to students via the paper assignment agent, which (as all agents) requests data from the blackboard, process it, then returns the data to the blackboard to be stored into the database.

4.2.4 Presentations Evaluations and Grading Agents

The instructor evaluates both the delivered presentation and the presenter's performance through the presentations evaluations agent. Then, the instructor also grads the presentation through the presentations grading agent. The agents have no direct connection to the database, only via the blackboard.

4.2.5 Reading List Agent

Through this agent, the instructor can provide a reading list to the student, where the data is processed to the database via the blackboard.

4.2.6 Reviews Evaluations Agents

The instructor can evaluate the submitted reviews through this agent that receives and passes the data though the blackboard.

5. Test Report

The reported results from testing the system are discussed below.

5.1 System Strengths

Developing the proposed system by PHP programming language along with adoption of some design principles results in the following features:

- The proposed system has a high degree of cohesion, where each agent (i.e. PHP file) performs only a set of closely related operations, which results in manageable, expandable, and maintainable tasks.
- The proposed system is easy to be extended to meet new requirements, where the instructor, for example, has the control of enabling presentations evaluations (i.e. certain tasks) without modifying the system's code (and the other tasks could be programmed the same).
- The proposed system reserves the least privilege principle by allowing specified privileges for specified subjects. As allowing a student to review its received feedback from instructor and other anonymous classmates.
- The proposed system allows students to review only their selected papers, and evaluate only the session-approved personation, which is the fail-safe defaults principle (explicit access to an object is required).
- The proposed system has a simple security mechanism (yet effective), where upon login-in the user's information is stored by using sessions (a way to store information in PHP) that are stored on server side. Instead of cookies that are stored on client side (less secure). This is an example of the economy of mechanism principle.
- Similarly, the proposed system destroys the stored data via session upon log-out, which gives safe and complete login-and-logout system

- Furthermore, the proposed system uses a random salting for password encryption, and pre-defined functions (by PHP main library) to encrypt and verify the password. Such a mechanism makes so-called rainbow tables useless.
- For the complete mediation principle (i.e. all accesses must be checked), the stored data via sessions are checked in all user's requests, when moving from page to another, for example.
- The proposed system provides well defined functions for validations that are re-usable with any HTML form integrated with PHP. For example, checking the validate of email address, or the length of input.
- By validation functions, the proposed system reduces the connection to database (i.e. a connection is requested when the input data are valid).
- The proposed system provides not only reviews submitting, but also reviews reports (similar for presentations)
- The proposed system gives the instructor full control (inserting, editing, deleting, and retrieval).
- The proposed system facilities the instructor's tasks by providing, for example, the accurate average of each personation's evaluations (i.e. the used equation is the total of submitted criteria divided by the number of submitted forms not the number of students, means absent students do not affect the calculation)
- The entire proposed system is mainly decorated by means of CSS files with no pictures, which makes it fast to load and browse.
- The proposed system successfully processes all user's input via means of pre-defined functions ("htmlentities" and "mysqli real escape string"), which keeps the system secured, and prevents SQL injection attack, for example.

6. System Weakness

Even though, the system has many features, it also has some limitations, as discussed below.

- The proposed system has incomplete implementation of validating input data (exactly in submitting reviews pages) due to the lack to time. However, the main validation functions as stated before are well defined and applicable for all inputs (means easy to fix).
- The login sub-system of the proposed system is considerably fast with no other way to slow it down. The problem of this point arises when a pre-programmed script is used to test thousands passwords in short time. The fix of this limitation could be done by adopting the capture-image validation, which is provided by may third-party developers.
- The proposed system has no HTML editor for writing comments. The easiest fix of this limitation can be done by adopting third-party editors.

7. Conclusion

In this paper, a web-based application to evaluate papers' presentations and reviews was presented. The blackboard architecture was adopted as the design architecture. Furthermore, the requirements and the results of system's test are discussed.

8. ACKNOWLEDGMENTS

The author acknowledges generous scientific support from Dr. Khedri starting from teaching, guidance, and including the allowance of using the course martials

9. REFERENCES

[1] Khedri R. Slides from a course on "Software Design", CAS Department, McMaster University, 2016.

[2] Hunt J., "Blackboard Architectures", JayDee Technology Ltd 27, 2002.

[3] Qian K. et al., "Software Architecture and Design Illuminated", Sudbury, Mass.: Jones and Bartlett Publishers, 2010, pp. 143-150.

[4] Butler G., "Blackboard Architecture", A course on "Software Design Methodologies", Department of Computer Science, Concordia University, 2015.

[5] Rudenko D. and Borisov A., "An Overview of Blackboard Architecture Application for Real Tasks", Scientific Proceedings of Riga Technical University, 2007.

[6] Metzner C., Cortez L., Chacin D., "Using A Blackboard Architecture In A Web Application", The Journal of Issues in Informing Science and Information Technology, Volume 2, 2005, pp. 743-756.

[7] Ortega-Arjona J. and Fernandez E., "The Secure Blackboard pattern", Procs. 15th Int. Conference on Pattern Languages of Programs (PLoP 2008).

[8] Corkill D., "Blackboard Systems", Blackboard Technology Group, Inc.