

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Smart Web Based Programming Contests Management Tool

Ahmed Bentiba, Mohamed J. Zemerly and Mohamed Al Mansoori
*Khalifa University of Science, Technology and Research, Sharjah
United Arab Emirates*

1. Introduction

Serious programming languages competitions world wide use the famous *Programming Contest Control System*, PC², developed at the California State University, Sacramento (CSUS, 2009)

We were involved during several years in local and regional ACM International Collegiate Programming (ICPC). We have participated at all levels: contestant, coaches and chief judge (KUSTAR, 2006 & 2007).

We found that PC² system has many limitations and weaknesses. PC² interface is limited to English language. It is mainly used for programming language contests and it is used in local area competitions in one center for example. Although it can be used in different centers, it cannot be used behind firewalls: "in a multi-site contest, every machine running a PC² server must be able to communicate via TCP/IP with the machines running PC² servers at every other site. In particular, there must not be any firewalls which prohibit these communication paths; the system will not operate if this communication is blocked" (CSUS, 2009).

PC² system is restricted only to programming contest types and it is limited to Pascal C/C++ and Java. Recently, Pascal has been dropped as a World Finals Language (CSUS, 2009).

For these reasons and many others, we decided to design an alternative to this system. We called it Wide Area Contests System, WACS (see Figures 1 and 2) to allow people from different cities or countries to participate in different contests not only programming language contests.



Fig. 1. WACS Login Page

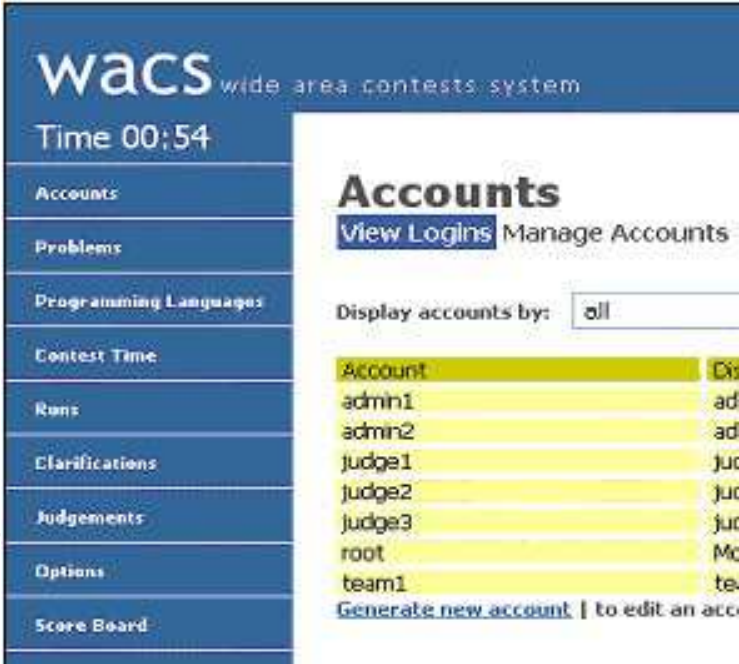


Fig. 2. WACS Web Interface

WACS is developed using different web languages and technologies such as PHP (Lerdorf, 1995), CSS (Lie, 1994; McFarland, 2006), JavaScript (Eich, 2005; Goodman & Morrison, 2004) and Ajax (Garrett, 2005; Zakas et al., 2007). It uses MySQL relational database system (Widenius & Axmark, 1994; Schwartz, 2008) to store its information rather than plain files. It is then more flexible and easier to add modules to WACS than to PC². Connecting to a contest on WACS is just as easy as to browse web pages. All what teams need is the contest's web address, login names and passwords. Figure 3 shows WACS web admin page.

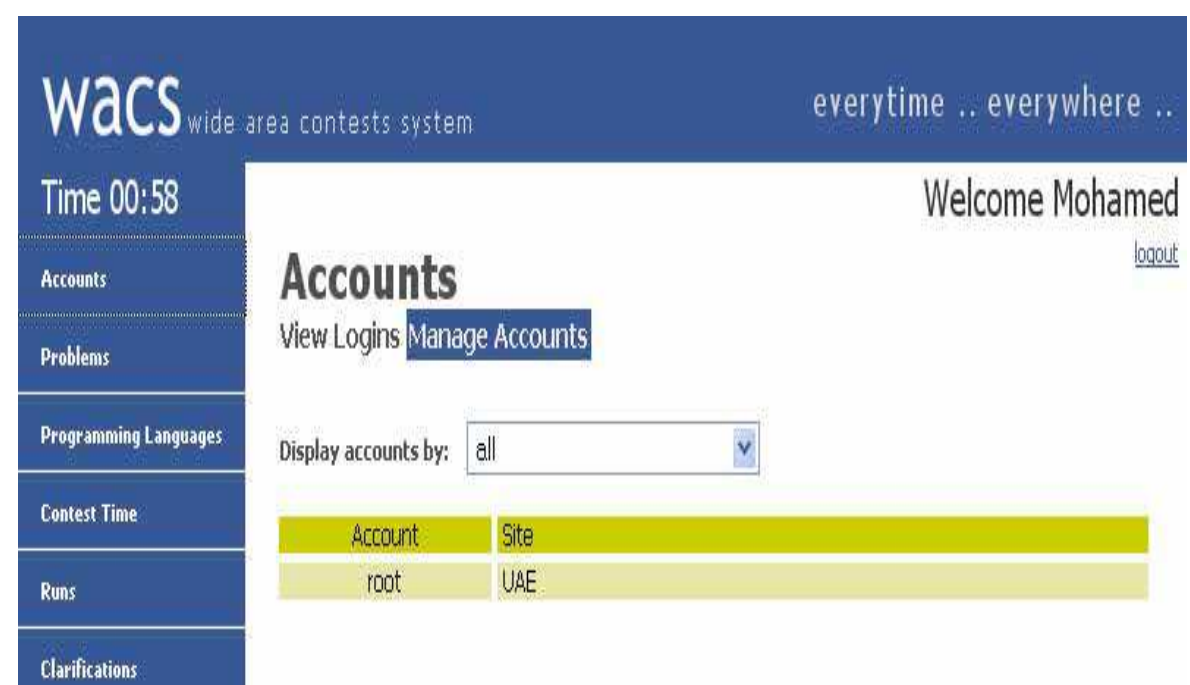


Fig. 3. WACS Admin Page

2. System Description

Although the original idea in developing the WACS system was to allow teams from different places to participate in different contests, WACS is implemented with a focus on programming languages contests as in Figure 4. However, other contest types, such as math, physics, Arabic, history and so forth can be easily plugged to the system as long as they are written using web languages and technologies.

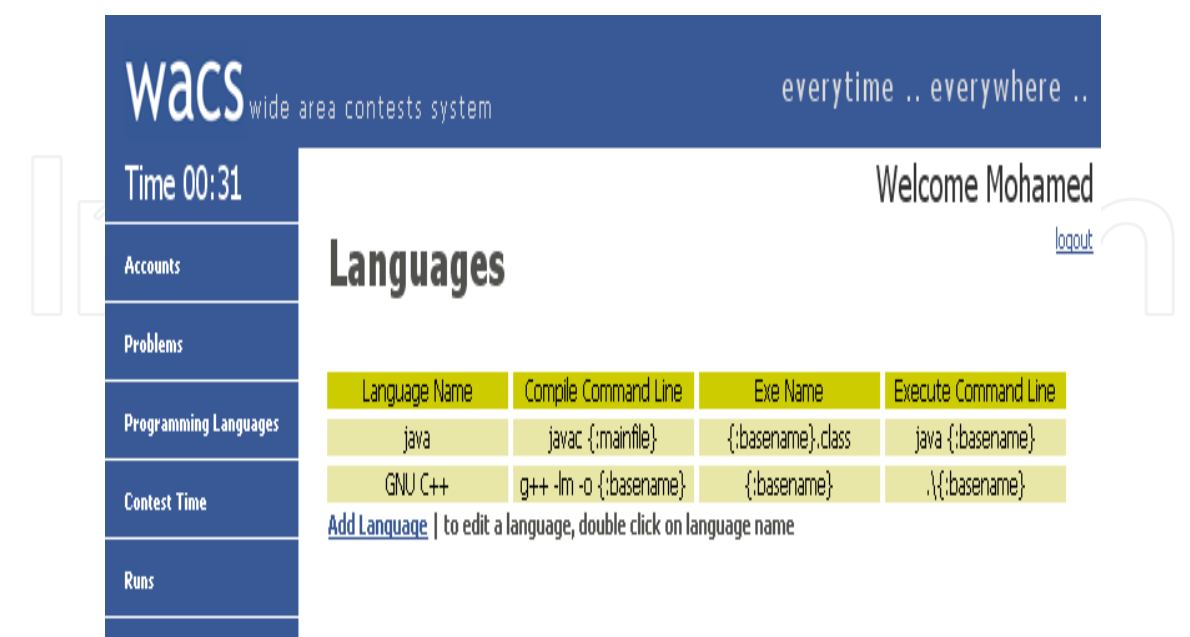


Fig. 4. Computer Languages

The system has three categories of users, which are administrator, judges and contestants or teams. The administrator has full control for the system. He/she is responsible for:

1. adding, updating and deleting accounts,
2. adding, updating and deleting problems or questions to solve,
3. adding or modifying programming languages to the system,
4. set the contest time, etc.

The system allows the contestants (teams) to interact with the judges during the contest through messages to request clarification and submit their answers to the problems. The judges can communicate with each team through messages to answer their requests. The judges can read the programs codes sent by teams and they can run them to evaluate them. The WACS system is able to run each team program code. Finally, the system provides a scoreboard that shows the results during the contest period with automatic update using the Ajax programming language without any manual page reload, as it is the case in PC². Ajax is originally considered an acronym for Asynchronous JavaScript + XML, the term is now used simply to encompass all the technologies that allow a browser to communicate (CSUS, 2009).

3. System Architecture

The WACS system is divided into two major parts: the base of the system and the contest type. The base of the system is where the WACS system allows people in:

- the same room or different rooms,
- in the same building or in different buildings,
- in the same city or in different cities,
- and in the same country or in different countries

to participate in a contest. It consists of an Apache web server (Apache, 2009) and a MYSQL database server (Schwartz et al., 2008) with PHP engine (Converse & Park, 2002).

On the client side, each educational institution can participate with one or more teams. Each team is composed of three students supervised by one or more coaches.

The contest itself is designed and supervised by the contest steering committee. During the contest, the responsibility is shifted from the administrators to the judges. The administrator will provide the judges with the technical support they need to run the competition smoothly. Figure 5 shows the main parts of the system's components.

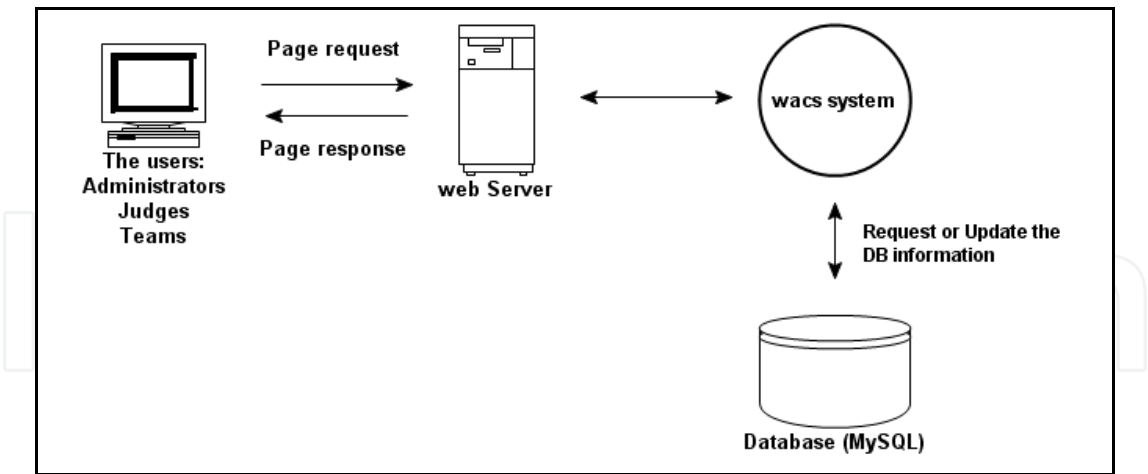


Fig. 5. WACS Components

Users communicate and interact with each other through the web server. Every category of users has a different web page to access the WACS. Any action performed in WACS is recorded within the MySQL database system, such as logins, submitting problems’ solutions, etc.. During the contest, each team can interact with the judges through the WACS system only by submitting their solutions and they may ask for clarifications or if they have any doubt regarding any contest question. The judges also can send contestants feedbacks or special messages through WACS only. There is no physical contact between teams and judges during the contest and between contestants and their coaches. The block diagram in Figure 6 shows the interactions between the three categories of users in the system.

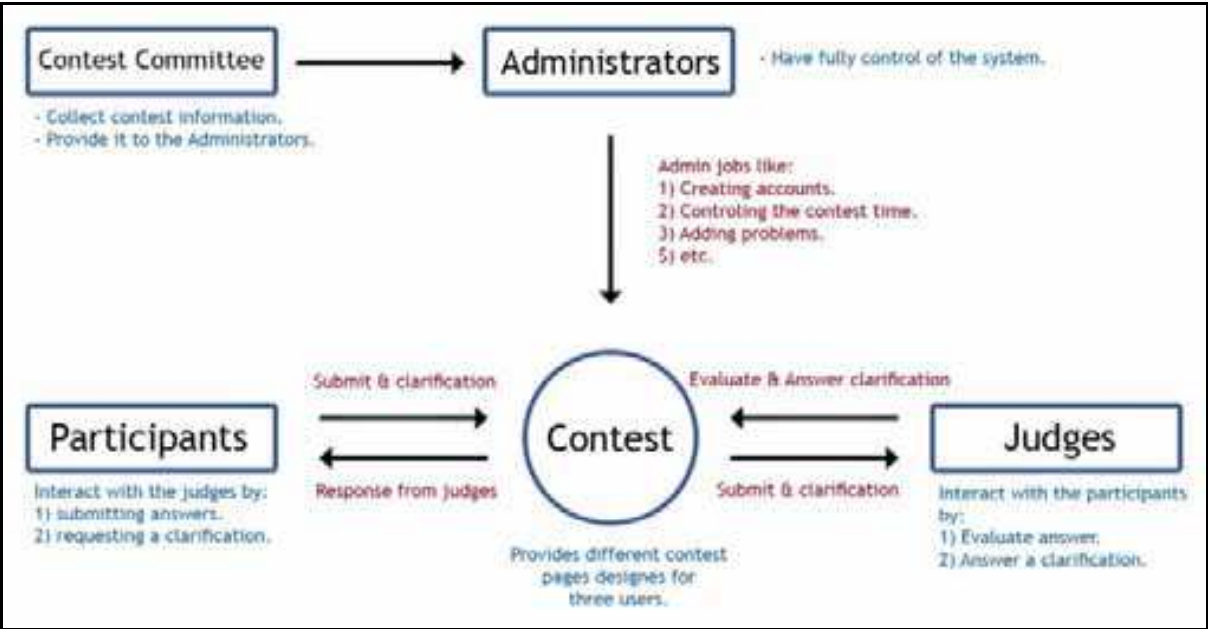


Fig. 6. Interactions between WACS users.

Figure 7 shows the entity relationship between various WACS tables. The details of these tables are not covered in this chapter.

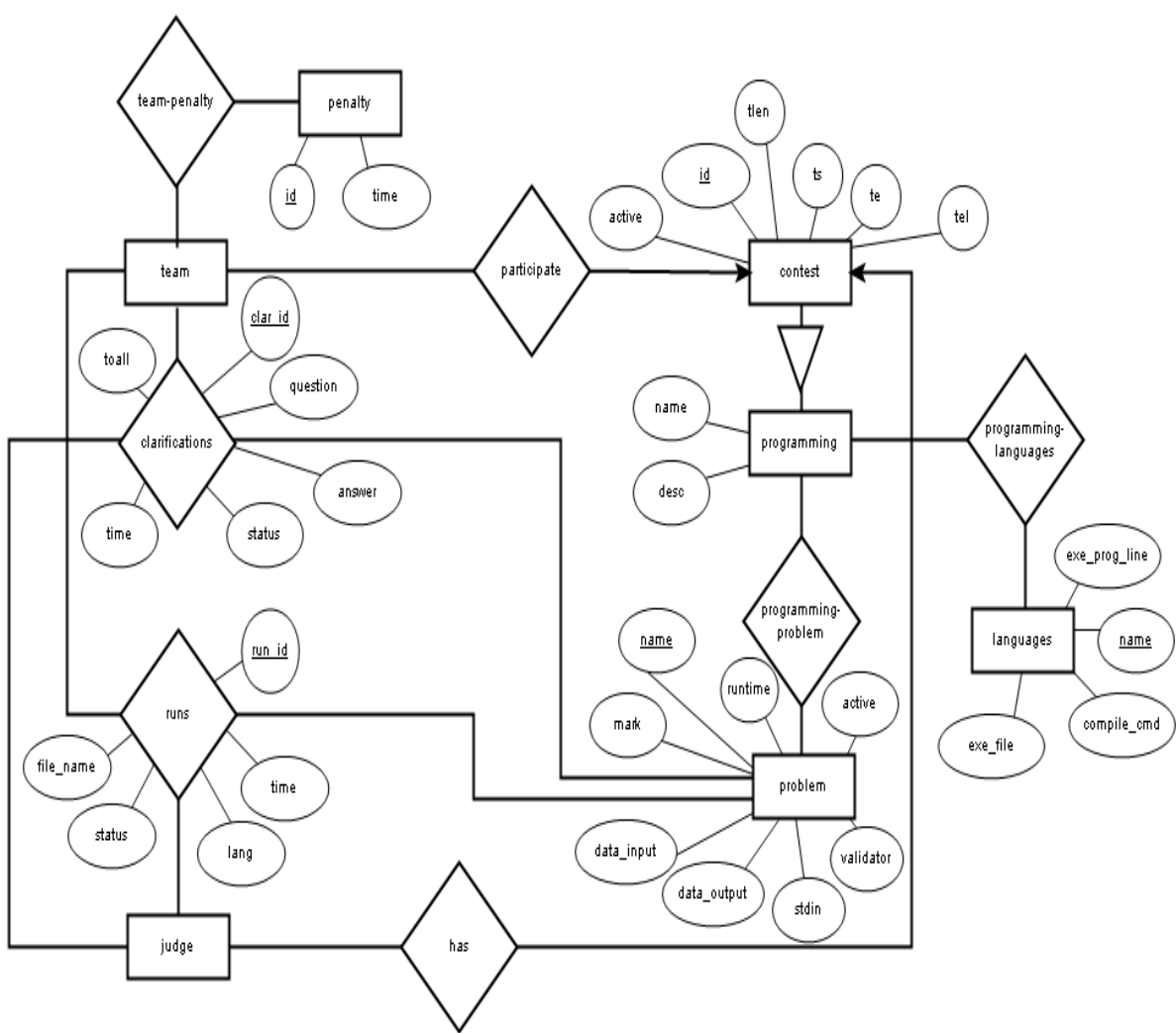


Fig. 7. Entity Relationship Diagram

Figure 8 shows the state-transition of the WACS system. The figure is self-explanatory.

3. Programming contest

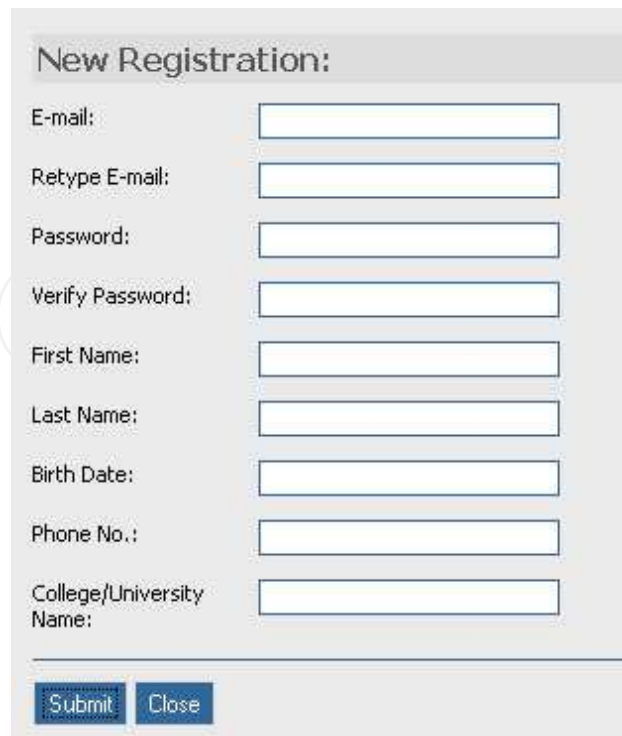
The WACS system is implemented to allow only coaches to register teams. Unregistered coaches, who have teams to participate in the contest, can register and then create accounts for their teams. Coaches have to fill standard forms as shown in Figure 9 to collect information such as names, passwords and email addresses and so on.



Fig. 8. State-transition diagram

The WACS stores the information in the database after checking whether it is there or not and sends an email to the coaches to activate their accounts.

The teams can submit their source code, once they finished solving any problem, via Submit window as shown in Figure 10. It provides the team with two options test the program before uploading it and submit it to the judges.



New Registration:

E-mail:

Retype E-mail:

Password:

Verify Password:

First Name:

Last Name:

Birth Date:

Phone No.:

College/University Name:

Fig. 9. Registration Form



Submit

Problem:

Language:

Main File:

Results: Your Answer Submitted Successfully.

Fig. 10. Submit and Test Page

The WACS administrator can add or remove problems. For each problem, he/she has to set the mark for it and to insert the input and output files into the database. WACS has a script that allows judges to compare the teams' results (outputs) with the results stored in the database for each problem. Depending on the results of these comparisons, the judges send appropriate messages to any team that submitted any source code for evaluation. The administrator manages the time of the contest. He/She can start, stop or reset the contest time. Finally, WACS has a scoreboard as shown in Figure 11 that allow users to view the teams' ranks, the number of tries and the number of problems solved by each team and the time taken to solve these problems. It fetches the information from the database and it is refreshed automatically using Ajax techniques (Asleson & Schutta, 2006).



Fig. 11. Contestants Score Board

3. Conclusion

This chapter describes the WACS system as an alternative to the famous *Programming Contest Control System*, PC². The implementation of the WACS system is mainly focused on programming languages contests. However, to add any other contest type is an easy task because of the flexibility of web programming languages and technologies. We used extensively, during the development of the WACS, open-source applications, web languages and technologies such as XHTML, CSS, PHP, Apache, MySQL, and Ajax. Since WACS should be installed on a server (running Linux or Windows) and without any component on client machines, it will reduce the burden for educational institutions or any participant without IT experts. All the configurations files, applications and servers (web, database) reside on the main server, which should be in the institution who hosts the competition. The only requirement needed in order to connect to WACS is a web-browser. The WACS system can be used inside computer labs, where students are given one or more problems to solve in a given period of time let say one or two hours. This activity will encourage students to work in a competitive environment. For the lab supervisors the code marking will be very easy and quick. Also students can use the WACS system in their free time such as breaks or weekends to improve problem solving skills and to practice in real competition environment.

4. Future work

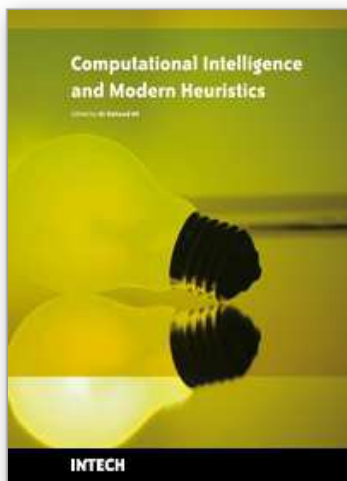
The WACS system needs to be tested in large group of teams and between different cities or countries to study the system limitations and performance. Regarding the performance of the WACS system, one should consider the compilation and the execution of programs to be put on a different physical server than the WACS web server. We should have more than one server in the system: One web server to host the WACS system, and at least one more server dedicated to compile the code files, execute the programs on other machines, and get the results from them. Some features can be added to the system, such as:

- 1. Security for login: right now, the passwords are sent to the system in clear text. It should be encrypted

2. The compilation of users' programs and running them are done with root/administrator privileges, which is not secure at all, since users, can send malicious code to damage the whole system. The solution to this is to use the system as unprivileged users such as nobody in Linux to limit the system damage.
3. Mirroring the system
4. The surveillance system: Cameras should be added in each center and should be controlled and monitored by the judges. One camera to monitor each team and several cameras to cover the whole center. If the camera system fails the center is withdrawn from the competition. As an alternative to cameras, each center or educational institution has to send one judge or staff to another center to supervise the competition. The educational institutions will cover the travel and hotel expenses of only this staff instead of all participants and their coaches.
5. The WACS system is implemented and tested for programming languages and can be extended easily for other subjects such as math, physics, Arabic or any other type to written tests.
6. The WACS interface is in English language, but we hope to implement also translations to other languages such Arabic, French and others.
7. As it is implemented now, only the administrator can start contests, add problems and users. It would be nice to allow some privilege users such lecturers or coaches to start their contests, to add their problem sets and to create their users.

5. References

- Apache Group (2009). Available [Online] <http://httpd.apache.org/>
- Asleson, R., Schutta, N. T. (2006). Foundations of Ajax, Apress.
- Brendan, E. (2005). Available [Online] <http://en.wikipedia.org/wiki/JavaScript>
- Converse, T. CSUS (2009). PC2 Home Page. Available [Online] <http://www.ecs.csus.edu/pc2/>
- Garrett, Jesse James (2005). Available [Online] [http://en.wikipedia.org/wiki/Ajax_\(programming\)](http://en.wikipedia.org/wiki/Ajax_(programming))
- Goodman D., Morrison M. (2004). JavaScript Bible, Fifth Edition Wiley.
- KUSTAR (2006). National Programming Contest 2006. Available [Online] <http://www.euc.ac.ae/eceportal/ece/npc2006.htm>
- KUSTAR (2007). National Programming Contest 2007. Available [Online] <http://www.euc.ac.ae/eceportal/ece/npc07.htm>
- Lerdorf, Rasmus (1995). Available [Online] <http://en.wikipedia.org/wiki/Php>, <http://www.php.net>
- Lie, Håkon Wium 1994, Available [Online] <http://en.wikipedia.org/wiki/CSS>
- McFarland, D. (2006). CSS: The Missing Manual, Pogue Press 1st edition.
- Park, J. (2002). PHP Bible, Wiley, 2nd edition.
- Schwartz, B., Zaitsev, P., Tkachenko, V., Zawodny, J. (2008). High Performance MySQL: Optimization, Backups, Replication, and More, O'Reilly Media 2nd edition.
- Sklar, D., Trachtenberg, A. (2002). PHP Cookbook O'Reilly Media.
- Zakas, N. C. , McPeak, J., Fawcett, J. (2007). Professional Ajax, Wrox, 2 edition.
- Widenius, M., Axmark, D. (1994). Available Online : <http://en.wikipedia.org/wiki/Mysql>



Computational Intelligence and Modern Heuristics

Edited by Al-Dahoud Ali

ISBN 978-953-7619-28-2

Hard cover, 348 pages

Publisher InTech

Published online 01, February, 2010

Published in print edition February, 2010

The chapters of this book are collected mainly from the best selected papers that have been published in the 4th International conference on Information Technology ICIT 2009, that has been held in Al-Zaytoonah University, Jordan in the period 3-5/6/2009. The other chapters have been collected as related works to the topics of the book.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Ahmed Bentiba, Mohamed J. Zemerly and Mohamed Al Mansoori (2010). Smart Web Based Programming Contests Management Tool, Computational Intelligence and Modern Heuristics, Al-Dahoud Ali (Ed.), ISBN: 978-953-7619-28-2, InTech, Available from: <http://www.intechopen.com/books/computational-intelligence-and-modern-heuristics/smart-web-based-programming-contests-management-tool>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen