

# Global Perceptions on the Use of WeBWorK as an Online Tutor for Computer Science

Olly Gotel, Christelle Scharff, Andrew Wildenberg, Mamadou Bousso, Chim Bunthoern, Phal Des, Vidya Kulkarni, Srisupa Palakvangsa Na Ayudhya, Cheikh Sarr and Thanwadee Sunetnanta  
 ogotel@pace.edu, cscharff@pace.edu, awildenberg@cornellcollege.edu, mamadoubousso@gmail.com, cbunthoern@rupp.edu.kh, phaldes@rupp.edu.kh, vkulkarni@cs.du.ac.in, ccsp1@mahidol.ac.th, csarr@thies.univ.sn, cctth@mahidol.ac.th

**Abstract** - Numerous (mostly commercial) web-based systems for the assessment of programming assignments have emerged in the past few years to support the teaching and learning of programming fundamentals. WeBWorK, an initiative led by the University of Rochester to support Mathematics education, is unusual in that it is an open-source and extensible system. Since 2005, collaborators at Pace University and Cornell College have been working to adapt WeBWorK to extend its reach to Computer Science. This paper reports on a global experiment undertaken with Computer Science students and faculty from three continents based on the use of WeBWorK. Students in the US, Cambodia, India, Senegal and Thailand were presented with a set of programming exercises in a controlled environment. The intention was to explore the impact of diverse cultures, distinct first languages and differences in prior everyday exposure to the Internet and use of pedagogical tools on the usability and perceived value of such tools in Computer Science education. The study poses an important question with regard to the global uptake of everyday and typically US-centric educational technology. It provides findings likely to be of value to academic institutions interested in its adoption and companies interested in its commercialization.

*Index Terms* - Automated Assessment Systems, C, Java, JUnit, Open Source, Programming, WeBWorK.

## INTRODUCTION

Technology is often seen as a silver bullet to solve many of the problems of the developing world. Airlift computers to the third-world and this will miraculously solve social and economic problems. This perspective is partially evident in the aspirations behind both the One Laptop Per Child (OLPC) (<http://laptop.org>) and the Intel's Classmate PC (<http://www.classmatepc.com>) initiatives that propose to send laptop computers with Internet access to villages with no electricity or water for use by children. Such initiatives depend heavily upon the existence of educational applications and their integration in educational programs.

Many students across the globe have minimal contact with their teachers and instructors outside of the classroom. Feedback on homework assignments can be either limited or

untimely, often because the student / instructor ratio can be very large or because of extremely high teaching workloads. Web-based assessment systems are designed to provide personalized feedback on individual student work in real-time. Such systems obviously have the potential to offer much needed educational support around the globe to both students and instructors, and to provide a ready application set for the emerging educational technology initiatives. However, most of these web-based systems have been developed primarily by Americans and Europeans for the US and European market. The major research question we propose to investigate in this study is: do students and instructors around the globe use these systems in similar ways and perceive the same values resulting from their use?

## WEB-BASED ASSESSMENT SYSTEMS FOR PROGRAMMING

Numerous (mostly commercial) web-based systems for the assessment of programming assignments have emerged in the past few years to support the teaching and learning of programming fundamentals (e.g., loops, conditionals, methods, arrays, recursion and sorting). They typically support questions ranging from true / false, multiple-choice and fill-in-the-blank type questions to writing program fragments in C, C++ and Java, with a predominance of focus on C / C++ in Europe and on Java in the US, reflecting the languages typically taught in the first programming courses.

Assessing a computer program involves measuring its correctness, quality (e.g., code typography) and authenticity (to prevent plagiarism). Correctness is generally evaluated based on running shell scripts and unit testing mechanisms. A selection of some of the early systems can be found in a 2004 journal issue dedicated to the topic [1]. These include: Quiz PACK (<http://www.sis.pitt.edu/~taler/QuizPACK.html>), CourseMarker (<http://www.cs.nott.ac.uk/coursemarker>) and BOSS (<http://www.dcs.warwick.ac.uk/boss>). New systems, such as CodeLab (<http://www.turingscraft.com>), JavaBat (<http://javabat.com>), Gradiance (<http://www.gradiance.com>), OWL (<http://owl.course.com>), ViopE (<http://www.viopE.com>) and WebCAT (<http://webcat.cs.vt.edu>) focus mainly on delivering quiz and programming questions with feedback. They offer graphical-based progress and performance visualization, with error highlighting of code and hints on how to correct it. They also begin to use sophisticated data mining techniques on the

repository of existing submissions to provide students' learning profiles and to generate dynamic questions at the appropriate level of difficulty.

Being developed primarily in the US and Europe, these web-based programming assessment systems target largely US and European Computer Science education. In this paper, we explore the perception and use of such tools with Computer Science students and faculty from the US, Cambodia, India, Senegal and Thailand. Students were presented with a set of WeBWorK Java or C programming exercises in a controlled environment.

WeBWorK [2], the targeted system, is an open source web-based system used to deliver and grade homework and distribute the solutions. Over sixty institutions use it to teach Mathematics. It offers traditional multiple-choice and fill-in-the-blank style questions. Since 2005, Pace University and Cornell College have been collaborating to adapt and extend it for use in the Computer Science curriculum. A library of questions has been developed for Java, Python, C and SML. Moreover, a major part of the initiative has been to develop a WeBWorK extension based on JUnit called WeBWorK-JAG (Java Auto Grader) to automatically collect and grade Java program fragments, thereby augmenting the standard range of WeBWorK questions to make it a better fit for practicing programming. Our efforts with WeBWorK have been described in previous papers [3,4,5].

## STUDY PARTICIPANTS

We selected institutions in the US, Cambodia, India, Senegal and Thailand to participate in this study due to prior relationships. A synopsis of each institution and its Computer Science education system are provided below and summarized in Tables I and II.

### I. Universities and Students

**Cornell College** (<http://www.cornellcollege.edu>) is a national private liberal arts undergraduate college located in Iowa, US. Its unusual One-Course-At-A-Time (OCAAT) schedule has students take one course in each of nine short terms of three and a half weeks. This intense curriculum puts students in contact with faculty for up to twenty hours per week. Students choose the major-related and liberal arts courses they take and generally graduate in four years. Many students choose to complete a second major or a minor.

**The University of Delhi** (<http://www.du.ac.in>) is one of the prestigious public institutions in India granting Bachelors, Masters and Ph.Ds. Admission to the Master of Science and Master of Computer Applications is through entrance examination and limited to thirty students in each. Students enrolled at the University of Delhi attend classes for about thirty hours a week and have lots of independent work.

**Mahidol University** (<http://www.mahidol.ac.th>), one of the oldest institutions of higher education in Thailand, was ranked number one in terms of qualities of both teaching and research by the Thai university league in 2006. The newly opened program in Information and Communication Technology (ICT) that grants Bachelors targets: Database

and Intelligent Systems, Multimedia Systems, Electronic Business Systems and Computer Science. Students take their specialty course to pursue their major in their third year.

**The Royal University of Phnom Penh** (RUPP) (<http://www.rupp.edu.kh>) is Cambodia's oldest and largest university. It is a semi-public institution offering undergraduate and graduate degrees and accepting students with scholarships only after the successful completion of a competitive entrance exam. Undergraduate students attend more than fifteen hours of class per week in their major only. Courses are organized by semesters. Students complete their degree within four years.

**The University of Thiès** (<http://www.thies.univ.sn>) in Senegal was created in February 2007 by merging the prestigious Polytechnic School of Thiès and the Thiès Engineering Schools, and adding the Economics and Social Sciences, and Agronomy education and research units. This public university offers general education at the undergraduate and graduate levels, and organizes the curriculum by semesters. Full-time students have around twenty-four hours of class per week and go through the courses by yearly cohorts.

Additional information concerning the five institutions is given in Table I.

TABLE I  
UNIVERSITIES

University	Type	Total Number of Students	CS Enrollment	Internet Access	Labs	Cohort
Cornell	Private	1,200	Even	Yes	Yes	No
Delhi	Public	220,000	Up	Yes	Yes	Yes
Mahidol	Public	24,000	Up	Yes	Yes	Yes
RUPP	Semi-Public	10,000	Up	No	Yes	Yes
Thies	Public	1,000	Up	Yes	Yes	Yes

### II. Computer Science Education

The targeted students were undergraduate and graduate Computer Science students. They assess themselves to be intermediate Java and C developers. Most of the students have access to computers – desktops or laptops – but are not so universally connected to the Internet. They access the Internet outside of school from home (US, India and Thailand) and from cyber-cafes (Cambodia and Senegal). Except for students from the US and Cambodia, most students had no prior experience with WeBWorK. The Cambodian students were exposed to WeBWorK during a visit of the US authors to RUPP. In addition, a few US students had prior experience with the Moodle course management system (<http://moodle.org>) and some Cambodian and Thai students used <http://www.w3schools.com> or Java certification online quizzes for practice. Overall 73.9% of the student subjects in the study did not have prior experience with web-based programming assessment systems. Additional information concerning the Computer Science education at the targeted institutions is given in Table II.

TABLE II  
COMPUTER SCIENCE EDUCATION

University	Total Number of Students	Class Size	Students' Load (hours per week)	Hours in Programming (per week)	Student/Prof Contact	Programming Languages by Order	Graded Homework Assignments
Cornell	30	15	20	20		Java, some C	Daily
Delhi	150	30	30	6		Java	Biweekly
Mahidol	778	70	18	3		C, Java, ASP.NET, C#	Weekly
RUPP	1870	60	15	3		Java, C, C++, PHP, C#	Monthly
Thies	120	60	24	6		C, C++, PHP, Java	Monthly

EXPERIMENTAL SET-UP

The experiment asked the students from the five institutions to answer ten C or Java questions within WeBWorK, followed by a survey to establish a baseline of the study population and solicit feedback on perceptions of using WeBWorK. Completeness of the surveys from each country (i.e., participation rates) varied from 60% to 100%. The instructors were familiarized with WeBWorK and completed a survey that examined teaching approaches, requirements for support and local barriers to the adoption of web-based systems, in addition to their perception of WeBWorK. Summary data about the set-up are provided in Table III.

TABLE III  
EXPERIMENT SET-UP

University	Number of Students	Level of Participating Students	Professors	Targeted Programming Language	Experiment Profctored on Campus	Participation Rate
Cornell	10	UG	1	Java	Yes	100%
Delhi	9 / 8	Grad	1	Java / C	No	60%
Mahidol	8	UG	1	C	No	90%
RUPP	10	UG	1	Java	Yes	100%
Thies	8	UG	1	C	Yes	80%

I. WeBWorK Problems

To address the programming language familiarities of the groups of students, sets of ten questions for C and Java were each created. Each problem set allowed for an unlimited number of attempts and time to completion was limited to ninety minutes. Each student was presented with authentication information to access WeBWorK. Questions between the two sets were designed to be as similar as possible, although since there is no possibility of automatically grading C programs within WeBWorK as yet, the WeBWorK-JAG questions were substituted with `printf` and `scanf` questions. A screenshot of one of the WeBWorK-JAG problems is provided in Figure I to illustrate the interface and format.



FIGURE I  
A REPRESENTATIVE WEBWORK-JAG PROBLEM

The common C and Java questions consisted of multiple choice questions on valid variable names, array declaration, loop semantics, and fill-in-the-blank questions on counting for and while loop iterations, and on drawing an 'X' composed of asterisks. The loop semantics question proposed eleven answers concerning the behavior of a piece of code and students had to select the four right answers. The C-specific questions addressed type expressions, type casting and conversion, and `printf` and `scanf`. Additionally, there were three WeBWorK-JAG questions. The first required writing a method that returns the negation of a single `boolean` parameter. The second asked students to write a method to fill an array of `ints` with the Fibonacci numbers. The last directed students to return the sum of the even numbers from 0 to a given limit (included), throwing an `IllegalArgumentException` for negative arguments.

II. Student Survey

The student surveys were administered within QuestionPro (<http://www.questionpro.com>) and comprised four sections. Section A collected background information, including the student's programming experience to date and typical weekly efforts spent on programming. Section B surveyed the student's access to technology to support learning: computers, Internet and applications, as well as prior exposure to web-based assessment systems. Section C queried the student's experience using WeBWorK in the study, using five-level Likert scales and nineteen short statements of opinion. It also gathered data on preference for the different WeBWorK question types, in terms of ease and perceived value for learning. Section D posed open-ended questions on what the student liked the most and least about WeBWorK, and solicited suggestions for improvement. A single question asked whether the student would consider using WeBWorK again. The student survey was completed by forty-four students in total from across the five institutions participating in the survey.

### III. Professor Survey

The professor surveys also had four sections administered by QuestionPro. Section A collected background information about the instructor, institution and programming classes, including languages, teaching style, homework levels, preparation and marking time, contact hours, etc. Section B surveyed the instructor's access to technology and its wider availability for student use in the classroom, labs and institution, as well as their prior exposure to web-based assessment systems. Section C asked the instructor's opinion on the potential use of WeBWork to support their teaching, using five-level Likert scales and a number of question clusters. These clusters gathered data on the perceived value of the different WeBWork question types, impressions as to how their students felt about WeBWork during the study, the most useful features of WeBWork for their teaching style and potential barriers to use. Section D posed open-ended questions on what the instructor liked the most and least about WeBWork, and solicited suggestions for improvement. A single question asked whether the instructor would consider using a web-based assessment system like WeBWork. The survey was completed by one professor from each institution participating in the experiment.

## FINDINGS

### I. WeBWork Results

Consistently for C and Java multiple-choice questions, the average number of attempts per students per question was the highest for the questions on the variable names, a question that was too specific on the syntax of the language, and the loop semantics question that required four answers chosen amongst a set of eleven correct answers. Moreover, this latter question required mastery of the English language, as the variants of the answers were very specific.

Not surprisingly, and based on the difficulty of the concepts and the free-form of the questions, the two Java programming questions linked with Fibonacci and exception handling, and the C questions on `printf` and `scanf`, also had a high number of attempts. A very interesting point was that the average number of attempts per question ranged from one to five, except for the US students whose average number of attempts per question was between one and twelve (and some questions were attempted more than twenty times). One explanation is that the US students are used to these web-based assessment systems, previous experience has taught them that they can experiment to arrive at the correct answer, and so they are driven by the goal of reaching 100% on the questions. Also, they had developed the distinct attitude that if the system scored them with a zero, the system was obviously wrong, something not immediately obvious in the other countries! As the experiment was arranged to last ninety minutes, very few of the students got all the questions completely correct.

The only real logistical problem affecting the experiment was the Internet connection in Cambodia. It was simply too slow and the RUPP students did not get to

complete the questions in the allocated time period. Note that we do not list the respective scores of the students in the various countries. The research question was not to compare and contrast their levels and abilities, but to examine their comments and reflections on the experience of using a web-based assessment system (see the section below).

### II. Students' Perception

All student participants stated they were seeking a Computer Science degree. The majority claimed to spend between two to ten hours in programming classes per week, with the US spending twenty hours (based on OCAAT), and with three to four hours being average. These students claimed to spend a comparable time on personal programming practice. Student responses regarding their access to technology in the classroom and their institution in general seemed higher than their professor's responses (see the next section). The largest issue for students was access to the Internet from home, this being a concern largely of the Cambodian and Senegalese students.

When students were asked: "*What did you like most about using WeBWork?*" the free-form responses overwhelmingly said the instant feedback. A US student replied: "*Fast and easy way to see what I understand and what I don't.*" In addition, the option for taking multiple attempts at a question was stressed as important. Ease of use was valuable to nearly all students but, basically, usability is crucial for such systems to be universally accepted.

When asked: "*What did you like least about using WeBWork?*" students mostly said the lack of clarity in the instructions and in the questions themselves, particularly with respect to the number of answers being sought in the multiple-choice type questions. This focus on the need for better clarity in the written descriptions (more than in the WeBWork interface itself) was the major topic raised when asked: "*Do you have any suggestions you would like to share with us for improving WeBWork?*" The Indian students suggested the most interesting improvements to WeBWork as it seems that they really saw such a system as a way for them to practice and improve. These suggestions included 'live' links to people available online to help, links to forums, email feedback and visible count-down timers.

Even though the WeBWork-JAG problems were found to be the most difficult, students said they were more challenged by and interested in these free-form coding questions. A US student commented: "*[I] liked how it makes you think about everything that you are typing, which many times is done for us by the compiler.*" Even the students taking the C version of the experiment and not exposed to the WeBWork-JAG questions seemed to value this style of question most for learning. The only real exception was the Senegalese students. A number of these students mentioned that they thought the multiple-choice questions were great, something that was not mentioned by any of the other students. It raises the question as to the level of prior exposure to such online multiple-choice style questions in various countries for educational purposes. The perception

as to the ease of the question type and the value of the question type do not correlate in general; while most of the students agreed that the multiple-choice type questions were the easiest for them to answer, the majority of the students said these were not as valuable to them as the other types.

Regarding the various reasons we presented to the students as rationale for possibly using WeBWorK, the most resonant with the students was simply – to practice. However, six students (five in the US and one in Thailand) did not think any similar such system would help them with their practice. Also, only six students (three in Cambodia, two in Senegal and one in Thailand) did not find WeBWorK fun to use. Overall, 72.5% of the students were interested in using WeBWorK again, while 27.5% were a maybe.

Representative quotes from the various student constituencies responding to the survey are given below to illustrate some of the country-specific dominant concerns:

- *“WebWorK is good but needs the user to have the Internet. So if we don't have internet we can't do it.” [Cambodia.]* For the Cambodian student, the concerns really lie with the logistics – how can we get to use systems like this in practice when our Internet connectivity is so problematic?
- *“Though it was my first experience at WeBWorK, I found it quite simple to use and really encouraging to remind me about Java, as I haven't did any java practice for the last 3-4 months. Its interface was nice.” [India.]* For the Indian student, systems like this provide the opportunity to extend their personal practice.
- *“I liked the concept of WeBWorK. It is a useful tool for testing students' level at programming.” [Senegal.]* For the Senegalese student, use of such a system is an instructive way to gauge how they are doing.
- *“WeBWorK make it convenient to do the test by automatically checking the answer and responding back to the user.” [Thai.]* For the Thai student, the automatic checking and feedback is a primary value of such a system, perhaps reflective of their largest class size amongst those participating in the experiment?
- *“It was a fun experience to go through the questions and find the answers. I enjoyed reviewing, and verifying that I know how to write code.” [US.]* For the US student, seeing that they get answers correct using such systems is a confidence builder.

### III. Professors' Perception

The professors reported a great disparity on the availability of the Internet and computers across the institutions. A number of the questions drew disparate responses of one and five on the Likert scale, reflecting the variety of situations and environments of the five countries. The availability of computers and the Internet at the university was better for the professors than for the students in Cambodia, India and Senegal. The Cambodian and Senegalese professors often use a computer to demonstrate concepts in the classroom, but do not require a connection to the Internet to do so.

In responding to possible barriers to adopting systems like WeBWorK, professors surprisingly were more

concerned about access to computers than to the Internet. Computers were used for programming classes often in Cambodia and Senegal, sometimes in Thailand and the US and never in India, while the Internet was never used in India and Cambodia, and rarely in the other institutions. Examining responses to this question further, it is obvious that all the professors seek systems to alleviate some of the time they spend compiling homework and test questions (one to five hours on average per week) and then spend on marking responses (one to ten hours on average per week). They all consider the investment in learning about and starting to use new educational system worth it, as the satisfaction with the current way they do things and the time necessary to learn new systems were ranked as the lowest barriers to entry (by far).

All the professors surveyed conduct their programming classes face-to-face with their students. The Indian professor conducts these classes in a traditional classroom, the Cambodian professor in a lab, and the remaining three use classrooms and labs evenly. Across the institutions, lectures accounted for 48% of the programming classes, with 34% of the remaining time spent on practical work using computers and 9% spent on other forms of practical work. All the professors spend between three and six contact hours per week with the students, except for the US professor who spends 20 hours, due to the OCAAT schedule at Cornell.

The question types that all the professors found the most valuable for student learning were open-ended and fill-in-the-blanks (very valuable); multiple-choice questions were favored only a little lower (some value), whereas they were more noticeably valued less by the students. Multiple-choice questions are often administered for expediency, so systems such as WeBWorK can offer similar levels of expediency whilst using much more valuable question types, one of the key drivers towards adoption.

Three of the five professors answered the question *“What did you like most about using WeBWorK?”* with a response similar to the following: *“I liked the immediate response. Students can get the feedback immediately and learn from mistakes.”* The nine potential features of WeBWorK that we sought opinions on regarding perceived value were all found either very valuable or just a little less so, the library of questions being the highest scoring and the ability to set time constraints the least so. Instant grading and the option to give students different problems to work on scored high, though the Senegalese professor did not value the feedback mechanism of right / wrong answers so highly preferring to get the overall score at the end.

The professors' perceptions of their students' experiences of using WeBWorK in the experiment included the observation that they seemed to enjoy participating. Professors stated that their students found the feedback helpful and wanted to keep trying questions when scored incorrectly (encouraging persistence). While some of the students ran out of time doing the questions, two of the professors said that their students found the questions too easy (India and Cambodia).

All the professors stated that they wanted to use a web-based assessment system and all expressed interest in using WeBWorK again. All were open to using new systems, especially if they included a library of prefabricated questions to use. Only the US professor had prior experience using web-based assessment systems, although some of the others had used a course management system before and expressed a desire to integrate WeBWorK within such a system. Professors seemed to value WeBWorK more for homework and classroom exercises than for examinations.

Representative quotes from the professors responding to the survey are included below to illustrate some of their dominant country-specific concerns:

- “*I think the problem should be clear - should [be] basic Java and OOP concept in Java. Internet was too slow to have all students complete the exercises.*” [Cambodia.] Reflecting their students’ concerns, Cambodian professors want to find a way to use educational applications to support their classes, but they cannot rely on web-based applications at present. An Intranet version will be more appropriate for a number of developing countries.
- “*I will be happy to see how it will test a large programming assignment with multiple classes.*” [India.] Again echoing their students’ motivation for such systems, the Indian professor was looking ahead to scaling up and taking WeBWorK further, increasing its value for students.
- “*Internet use which expands collaborations.*” [Senegal.] The major motivation for the Senegalese professor appeared to be the potential for systems like WeBWorK to help them integrate into a wider web of Computer Science educators, to exchange and share materials beneficial to students. Such systems naturally bring together professors with the same pressures and concerns from across the globe to leverage each other’s hard work and experiences.
- “*...limited programming library; expand problem library...*” [US.] This comment reflects the fact that the US professor (one of the WeBWorK collaborators) is aware of pre-existing programming questions in the WeBWorK library, their potential value for use by others, and the need to get to work in building up the repository.

### CONCLUSIONS AND RECOMMENDATIONS

Systems such as WeBWorK offer the potential to transfer knowledge and teaching practices from one country to another. Developing countries like Senegal demand this. In such countries, open source tools are undoubtedly the only real option, so the ability to develop customized questions within such open frameworks is essential. Given the current problematic state of Internet connectivity in both Cambodia and Senegal, deploying a system such as WeBWorK over an Intranet is an obvious strategy to explore.

The main challenges surrounding use of systems such as WeBWorK lie with the system and question designers (also professors). The formulation of questions has to be simple and instructions universally clear in terms of language. Problem writing is not a trivial undertaking and requires time-consuming quality assurance prior to deployment.

Future development work, given the preference for and value assigned to open-ended programming questions, means integration with additional Integrated Development Environments would be desirable.

### ACKNOWLEDGMENT

We thank all the students who were involved in this study. The WeBWorK in Computer Science project is supported by NSF CCLI AI grants #0511385 and #0511391. We are grateful to Sophal Chiv, Jacqueline Baldwin and Nathan Baur for their help with WeBWorK. We also would like to thank the Cambodian One Click Communications cyber-cafe (<http://www.cocc.biz>) for facilitating access to the Internet.

### REFERENCES

- [1] Special Issue on Automated Assessment of Programming Assignments, Journal on Educational Resources in Computing (JERIC), 5, 3, 2005.
- [2] Gage, M.E. and Pizer, A.K., “WeBWorK – Math Homework on the Web”, Proceedings of the Annual International Conference on Technology in Collegiate Mathematics, 1999.
- [3] Gotel, O. and Scharff, C., “Adapting an open-source web-based assessment system for the automated assessment of programming problems”, IASTED Web-based Education Conference, Chamonix, France, March 2007.
- [4] Gotel, O. and Scharff, C. and Wildenberg, A. “Extending and contributing to an open-source web-based assessment system for the automated assessment of programming problems.” Proc. ACM Conference on Principles and Practices of Programming in Java, Lisboa, Portugal, September 2007.
- [5] Gotel, O. and Scharff, C. and Wildenberg, A. “Teaching Software Quality Assurance by Encouraging Student Contributions to an Open Source Web-based System for the Assessment of Programming Assignments”, The 13th Annual Conference on Innovation and Technology in Computer Science Education, Madrid, Spain, 2008.

### AUTHOR INFORMATION

- Oly Gotel**, Assistant Professor, Pace University, NY, USA, [ogotel@pace.edu](mailto:ogotel@pace.edu).  
**Christelle Scharff**, Associate Professor, Pace University, NY, USA, [cscharff@pace.edu](mailto:cscharff@pace.edu).  
**Andrew Wildenberg**, Assistant Professor, Cornell College, IA, USA, [awildenberg@cornellcollege.edu](mailto:awildenberg@cornellcollege.edu).  
**Mamadou Bousso**, Assistant Professor, University of Thies, Senegal, [mamadoubousso@gmail.com](mailto:mamadoubousso@gmail.com).  
**Chim Bunthoern**, Assistant Professor, Royal University of Phnom Penh, Cambodia, [cbunthoern@rupp.edu.kh](mailto:cbunthoern@rupp.edu.kh).  
**Phal Des**, Director of IT Center, Royal University of Phnom Penh, Cambodia, [phaldes@rupp.edu.kh](mailto:phaldes@rupp.edu.kh).  
**Vidya Kulkarni**, Reader, University of Delhi, India, [vkulkarni@cs.du.ac.in](mailto:vkulkarni@cs.du.ac.in).  
**Srisupa Palakvangsa Na Ayudhya**, Lecturer, Mahidol University, Bangkok, Thailand, [ccspl@mahidol.ac.th](mailto:ccspl@mahidol.ac.th).  
**Cheikh Sarr**, Assistant Professor, University of Thies, Senegal, [csarr@thies.univ.sn](mailto:csarr@thies.univ.sn).  
**Thanwadee Sunetnanta**, Assistant Professor, Mahidol University, Bangkok, Thailand, [cctth@mahidol.ac.th](mailto:cctth@mahidol.ac.th).