



Title	Practical Problem-Based Learning in Computing Education
Authors(s)	O'Grady, Michael J.
Publication date	2012-07
Publication information	O'Grady, Michael J. "Practical Problem-Based Learning in Computing Education." ACM, July 2012. https://doi.org/10.1145/2275597.2275599 .
Publisher	ACM
Item record/more information	http://hdl.handle.net/10197/4342
Publisher's statement	This is the author's version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in the BACM Transactions on Computing Education, Volume 12 Issue 3, July 2012, Article No. 10 http://dx.doi.org/10.1145/2275597.2275599
Publisher's version (DOI)	10.1145/2275597.2275599

Downloaded 2025-07-03 19:07:20

The UCD community has made this article openly available. Please share how this access benefits you. Your story matters! (@ucd_oa)



© Some rights reserved. For more information

Practical Problem Based Learning in Computing Education

Michael J. O'Grady, University College Dublin

Computer Science (CS) is a relatively new discipline and how best to introduce it to new students remains an open question. Likewise, the identification of appropriate instructional strategies for the diverse topics that constitute the average curriculum remains open to debate. One approach considered by a number of practitioners in CS education involves Problem Based Learning (PBL), a radical departure from the conventional lecturing format. PBL has been adopted in other domains with success, but whether these positive experiences will be replicated in CS remains to be seen. In this paper, a systematic review of PBL initiatives in undergraduate and postgraduate CS is presented from a Computing Education Research (CER) perspective. This includes analyses of a range of practical didactic issues, including the degree to which PBL has been systematically evaluated, practical problem description in the literature, as well as a survey of topics for which a PBL approach has been adopted.

Categories and Subject Descriptors: K.3.2 [**Computer and Education**]: Computer and Information Science Education—*Computer Science Education*

General Terms: Human Factors

Additional Key Words and Phrases: Problem-Based Learning (PBL), Computing Education Research (CER)

1. INTRODUCTION

Practitioners in Computer Science (CS) education face many of the same issues as their peers in other disciplines. These include how best to enable deep learning in the subject matter, as well as standard issues such as retention rate and so on. CS is a very young discipline relatively speaking, only coming into university establishments in the last fifty years. Thus the subject matter remains quite dynamic in many respects. More importantly, there is an ongoing debate concerning the core nature of CS and computing [Müller 2008; Denning 2005]. This issue of definition is aggravated by the interdisciplinary nature of CS, and its need to respond to ongoing technical developments, which have been remarkable in the last few decades, and continue unabated. All of this combines to raise particular difficulties for CS education practitioners who must accommodate the notions of CS as a discipline in its own right, as well as manage the needs of other constituencies who may have a different perspective, for example, one that may regard computing as tool for aiding in the fulfilment of a multiplicity of purposes.

As with many disciplines, those charged with computing education have acted primarily as knowledge transmitters. However, this method of instruction, though essential in some instances, is perceived as limited in that it does not promote higher order thinking and advanced reasoning skills amongst others. Both skills are essential for students seeking to make careers in the computing industry. However, how to do this in CS courses remains an open question; thus many CS education practitioners have explored alternative approaches to course delivery. For example, Project-Based Learning [Krajcik and Blumenfeld 2006] and Inquiry Based Learning [Edelson et al. 1999] are just two approaches that have been considered. For the purposes of this discussion, Problem Based Learning (PBL) is considered in the context of CS undergraduate and postgraduate education. PBL has been adopted successfully in medical education and its potential in other disciplines is being actively explored. Indeed, Ellis et al. [1998] consider that the CS discipline lends itself to the PBL approach as it is for the most part problem driven. Furthermore, the project group is the dominant modus operandi of the computer industry - an industry that is highly dynamic, making lifelong learning a necessity. Since the inception of PBL, a number of CS educators have explored its potential by designing and delivering individual modules around this format, and published their experiences.

In this paper, an analysis of the existing literature in PBL CS education is provided. This complements and extends an earlier review described by Beaumont et al. [2004]. The paper is structured as follows: Section 2 provides a brief introduction to PBL. The methodology followed for this review is described in Section 3. Section 4 presents the results of the analyses in a number of didactic dimensions, after which the paper is concluded.

2. REVIEW OF PROBLEM BASED LEARNING

Problem-Based Learning (PBL) [Dolmans et al. 2005; Boud and Feletti 1997] is an instructional approach that makes the student the focus of the learning process, seeking to empower them such that they take responsibility themselves for their own learning. This differs from the traditional approach where the lecturer or instructor, acting as a transmitter of knowledge, drives the learning process. PBL is grounded in the philosophy of John Dewey who believed that learning is based on discovery and that it is guided by mentoring rather than the transmission of knowledge. In this way the curiosity of the learner is aroused, resulting in them questioning, critically thinking about problems, and, hopefully, creatively solving problems.

PBL has its origins in medical education and was first implemented at McMaster University in the 1960s. It was recognized by Barrows [1996] that a combination of hypothetical-deductive reasoning augmented with specific knowledge in a range of domains was essential to the diagnostic process. The traditional lecture approach isolated most of the relevant content from its context or its clinical application. PBL was seen as a remedy to this deficiency. In the intervening years, PBL has been widely implemented by many medical faculties, and is gaining momentum in other domains such as business and engineering. It must be stated unequivocally that there are a number of interpretations of PBL amongst its practitioners and that as yet, a consensus has not emerged. Indeed, a rigorous definition may never emerge as PBL methodologies may need to be modified according to the characteristics of the learning domain in question.

In practice, following the methodology outlined by Barrett [2005], students are presented with a problem. In small groups, this problem is discussed and refined, resulting in students identifying what it is that they need to know to solve this problem. Outside of the tutorial, students engage in independent study on the learning issues. They then report back on what they have learnt, share information, debate and argue if need be, and come to a consensus about a solution to the problem. Depending on the scale of the problem, and whether the lecturer is adopting an approach of full or partial problem disclosure, this cycle may be repetitive. Ultimately, students review what it is that they have learnt, engaging in a process of self, peer and tutor review.

Finally, it is instructive to reflect on the effectiveness of PBL. The wide-spread implementation of PBL in medical schools suggest faculty view it as a positive development. For example, Tiwari et al. [2006] in a study of undergraduate nursing students identified significant differences in students' disposition to critical thinking with those exposed to PBL showing a higher score than those who attended a traditional lecture format. In a meta-analysis of studies conducted between 1976 and 2007, Walker and Leary [2009] claimed that on standardized tests of basic concepts, PBL can hold its own in comparison with lecture-based approaches. Furthermore their study found that when assessments measured application of knowledge and principles, the results clearly favoured PBL, a conclusion also reached by Strobel and van Barneveld [2009]. However some PBL researchers have observed that empirical evidence supporting the superiority of PBL over traditional approaches is almost non-existent and that due caution should be exercised [Sanson-Fisher and Lynagh 2005; Carrero et al. 2007]. Kirschner et al. [2006] claim that minimal guidance during instruction does not work, concluding that PBL and other constructivist approaches do not take the human cognitive architecture into consideration. Schmidt et al. [2007] take issue with this, agreeing that novices are a particular case in point, but that the use of PBL is not

equivalent with minimally guided instruction. This debate will undoubtedly continue. For the purposes of this discussion, the observation of Boud and Feletti [1997] that "it is rarely possible to translate a given approach from one context to another without considerable modification" is noted; whether the success of PBL in other disciplines can be replicated in CS remains to be seen.

3. REVIEW METHODOLOGY

The overall objective of this study is to investigate how PBL is being harnessed in curricula oriented towards the teaching of CS. Keeping in mind that in the ACM Computing Curricula [ACM 2005], CS is perceived as one sub-discipline, along with Computer Engineering, Information Systems, Information Technology and Software Engineering, there are a wide variety of courses that could be perceived as common to each sub-discipline. While acknowledging the distinction, in practice, many course modules are being developed that could be seamlessly integrated into a course in each of these sub-disciplines. Thus any PBL case study that contributes to any of these sub-disciplines is of interest. The domain is undergraduate and postgraduate curricula. Professional and adult education courses, as well as K-12, are outside the scope of this study.

3.1. Methodology Specification

A number of categorisation methodologies for Computing Education Research (CER) have been documented in the literature, for example, see Joy et al. [2009]. These seek to identify a range of characterises of the published research that would serve as suitable criteria for categorisation or taxonomic purposes. This is a subjective process, and additional dimensions could be required, according to the requirements of the research agenda in question. Thus, a generic categorisation methodology that meets the needs of all research constituencies does not exist, as of now. In the meantime, a variety of schemes have been proposed, amended and adopted.

Fincher and Petre [2004] identified 10 research area categories in computer science education research. Vessey et al. [2005] proposed a unified classification scheme for computing research that incorporates 5 dimensions - topic, research approach, research method, unit/level of analysis and reference discipline. Simon [2007] proposes four dimensions: topic, context, nature and scope. Sheard et al. [2009] used five criteria for classifying research - type of research, data gathering techniques, analysis techniques, aims of the research & research outcomes.

Recently, the didactic triangle [Kansanen and Meri 1999] has been considered as a basis for CER. Berglund and Lister [2010] have harnessed this construct as a basis of analysing the teaching and learning of programming. Kinnunen et al. [2010] have defined a methodology based on the didactic triangle but augmented with two additional levels. The basic triangle - a triad of teacher-content-student, represents the course perspective. The second level represents the perspective of an organisation while the third level represents that of the society. The net result is a three-layered didactic structure where the nodes co-exist and interact, giving rise to eight classification categories. This classification scheme supports meta analyses of published research, and, according to the authors, seeks to identify missing types of research.

Any of the schemes described here provides a basis for an analysis of the PBL literature in the CS domain. However, how any approach in of itself explicitly support the extraction of details pertaining to practice is open to question. It is not the intention here to develop a new methodology that specifically addresses the needs of this survey. Thus it is proposed to adopt a two stage approach.

- (1) In the first instance, all papers will be categorised using the approach proposed by Kinnunen et al. [2010]. This offers an intuitive classification scheme that incorporates

the key actors (students, teachers and so on) as well as the content and objectives of educational programmes and their interrelationships.

- (2) Secondly, a number of additional questions will be formulated that seek to identify what motivated the adoption of PBL and how the success or otherwise of the initiative was quantified. Given the importance of problems, are any of the problems used in the case studies documented and discussed. Finally, where in the CS curriculum is PBL being harnessed.

Formally, the four additional questions being addressed in this analysis are as follows:

- (1) What has motivated the adoption of PBL?
What factors have influenced lecturers to adopt PBL? For example, was there a institutional or departmental policy? or was it an initiative on behalf of the lecturer who was dissatisfied with traditional teaching methods and believed students would respond better to a different approach?
- (2) Has PBL been systematically evaluated in CS courses?
Has there been any efforts made to evaluate the effectiveness of PBL in comparison to other approaches? Rather than a generic questionnaire canvassing students' opinions on select aspects of an arbitrary course, has there been some effort made to ascertain what aspects of the PBL process were successful or otherwise?
- (3) Are the problems used in PBL courses documented?
Suitably scoped problems are essential to PBL, and will not be found in typical course textbooks. What the key features of such problems might be has been explored, for example Duch [2001] and Weiss [2003] suggest a number of desirable characteristics. For those new to PBL, examples of problems that have been applied in the classroom would inform their own problem design efforts. Likewise, experienced practitioners may find a reflection on practitioners' experiences with documented problems instructive.
- (4) Where in the CS curriculum has PBL been harnessed?
This addresses the practical question of what levels has PBL been harnessed in a typical computing curriculum and what topics have been presented through PBL.

3.2. Research Paper Selection & Refinement

In the first instance, a number of relevant journals were identified and their online repositories searched. Secondly, a number of digital libraries were searched including those of the Association for Computing Machinery (ACM) and the Institute of Electrical and Electronic Engineers (IEEE). Finally, Academic Search Premier was queried. Generic searches using appropriate combinations (including spelling variations) of the terms "Problem Based Learning" and "computing" or "software" resulting in a significant number of potentially relevant publications being returned. To make this process more manageable, only those publications that had some combination of these search terms in the title, abstract or keywords were considered. The acronym PBL was not used as this returned a significant quantity of papers on Project Based Learning. The abstract was read initially to determine the degree to which the paper was concerned with the practice of PBL. Only papers whose content was influenced by practice were retained. The text and references of all these papers were examined, leading to the identification of additional studies which were subsequently found in Google Scholar. Finally, a number of colleagues provided references to papers that they considered relevant. For this study, the cut-off date was November 2011.

Ultimately, 107 publications that merited a more detailed analysis were identified. Of these, 44 were rejected for a variety of reasons. Some proposed modified PBL practices; others were courses that were in preparation but had not been delivered in practice. A number of authors described blended learning approaches with PBL being just one contributory approach. Papers that did not make reference to the PBL literature were also excluded - using problems to aid learning is not the same as adopting a PBL approach.

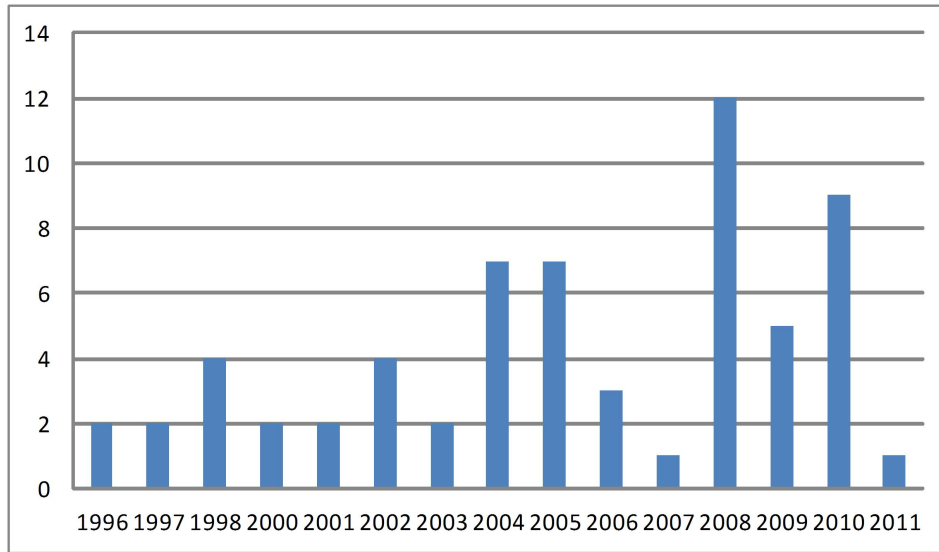


Fig. 1. Publications concerning PBL in CS by year.

Where a number of case studies referred to the same course over a number of years, the most recent treatment was analysed as this tended to be the most relevant. Studies that were not part of a broad computing curriculum were excluded. In some instances, the classic 7 step model for PBL [Schmidt 1983] was not fully adopted due to a variety of constraints. However, this did not serve as justification for excluding a study. Finally, 63 papers were selected for inclusion in this study. Figure 1 illustrates the spread of these papers by date of publication. Papers appeared in print from the mid-1990s onwards. While the number increased in the last decade, it is too early to say if there is trend that might indicate whether PBL will flourish in CS education, or otherwise.

4. ANALYSIS

All papers were initially categorized using the scheme proposed by Kinnunen et al. [2010]. Each was then further scrutinized to ascertain how PBL was adopted in practice, thus providing answers to the additional research questions outlined in Section 3.1.

4.1. Categorization of PBL Publications

The results of the categorization may be seen in Table I. While all papers have been assigned a primary category, it sometimes occurs that a paper may have a strong claim for inclusion in another category. Four such papers were identified and these are indicated in parenthesis. Overall, it can be seen that all but four of the publications were focused at the course level.

To elucidate further on the classification process: many studies focused on the issue of pedagogy. In many cases, where authors reported on course design, augmented with some brief observations or anecdotal comments concerning either their own experience or that of their students, for example Fabiani [2009] and Matzko and Davis [2008], such studies were classified under Category 7.3, the pedagogical activities of teachers. Other studies considered the issue of pedagogy but proceeded to provide a more in-depth analysis of the results, for example, those studies by Qiu and Chen [2010], Pereira et al. [2010], and Wang et al. [2010]. Such studies were classified as Category 8 which focuses on how students relate to pedagogical actions. Indeed, two studies in this category, those of Hämäläinen [2004] and Clarke et al. [2005], compared the outcomes of both the traditional and PBL approach

across a number of dimensions. Richardson and Delaney [2010] describe a PBL case study and proceed to provide a detailed reflection on their experience; this study is classified as Category 7.1, as it primarily reports on their conceptions of students' experiences. Mitchell and Delaney [2004] focuses on the issue of assessment to determine learning outcomes; this study is classified as Category 6, the relation of the teacher to the goals of the instructional process. All of these studies are placed at the course level.

Table I. Studies classified according to their primary didactic focus, and secondary focus

	Category	Course Level	Organization Level	Society Level
1.	Goals & content	-	-	-
2.	Student(s)/community of students /citizens	-	-	-
3.	Teacher(s)/organization/society	-	-	-
4.	Student(s)/community of students/citizen - teachers/organization/society	-	-	-
5.1	Students: understanding of and attitude about goals and content	- (1)	-	-
5.2.	Students: actions of students	-	-	-
5.3.	Students: results of students' actions	2	-	-
6.	Teacher(s)/organization/society - goals and content	5	2	1
7.1.	Teacher's conceptions of student's understanding of goals/ content	6	-	-
7.2.	Teacher's conceptions of students' actions towards achieving goals	2	-	-
7.3.	Teacher's pedagogical activities	20(2)	-	-
8.	Student(s)/community of students/citizens - teacher(s)/organization(s)/society's pedagogical means to enhance learning	24(1)	1	-

Three studies were placed at the organizational level. Fee and Holland-Minkley [2010] consider PBL across the entire curriculum of their school whereas Gibson [2009] considers how the teaching of formal methods can be integrated across a curriculum. Bunch [2009] presents the design of a curriculum for a four year degree in Web development. The study described by Hogan and Thomas [2005] is of particular interest in that it is primarily motivated to equip students with those soft skills that they will need to progress in their working lives, in this case, to work as professional software engineers. Thus the study is placed at the society level.

Most of the studies cluster around two aspects - pedagogical interventions and students' reactions to these innovations. Teachers' perceptions and experiences of the PBL process and students' experiences were also documented. However, no systematic studies focusing exclusively on teachers' or organisations' views were found. Likewise, no study explored students' perception of their lecturers or facilitators in the PBL process. Studies of students actual engagement with the PBL process remains an open area for research in CS contexts. Likewise, issues pertaining to the overall objectives or goals of a CS PBL course, and the degree to which these were met, both from a CS learning outcomes perspective, but particularly in terms of those student attributes, for example, initiative, that PBL promises

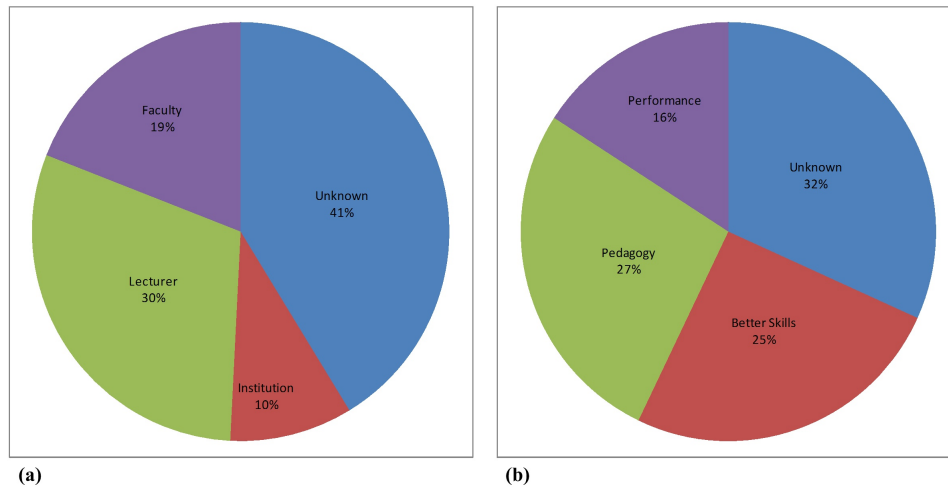


Fig. 2. (a) Who championed the introduction of PBL?; (b) Why was PBL adopted?

to engender, remain to be addressed. Attitudes and experiences of teaching faculties as a group, and the perceptions of management, remains to be seen.

Given the relative newness of PBL and the dynamic nature of the computing technology with its inherent effects on computing curricula, the emphasis on the course level is understandable. From an organization level perspective, it would be useful to see some studies exploring the pertinent aspects of students' and teachers' experience in curricular where PBL is the norm. At the society level, it would be useful to ascertain how students who have completed their computing studies through PBL are perceived by key stakeholders including employers and research supervisors.

4.2. What has motivated the adoption of PBL?

Motivation for introducing PBL can be considered under two headings: who inspired the adoption of PBL in an arbitrary institution? and what motivated this? Disappointingly, a significant number of authors did not explicitly address these issues (Figure 2). Of those that did, the lecturer was found to be the dominant driving force for introducing PBL, followed by faculty (19%) and the institution (10%). The reasons for the introduction of PBL were mainly pedagogical (27%) and included enabling student-centered learning, encouraging students to learn by doing, and to foster independence, amongst others. The other dominant reason was the perception that PBL would aid students acquire skills that would serve them better in the workplace (25%). Finally, certain courses had been experiencing difficulties over a period of time, for example, poor grades and high dropout rates. In these cases (16%), it was hoped that the introduction of PBL would help address these difficulties. All authors reported that this was indeed the case. Some used exam results and student evidence as evidence; however, in many cases, it was anecdotal.

4.3. Has PBL been systematically evaluated in CS courses?

It is good practice to acquire feedback on all aspects of a course, and practically all institutions demand this. Thus all research papers reported feedback to varying degrees. As far as could be ascertained, many of the instruments used were those approved by the institution in question. This would have had the advantage of continuity, enabling comparison along a consistent range of dimensions, including final grades. However, this does nothing to isolate factors that might have influenced student perceptions and experiences of PBL, either

positively or negatively. Indeed, only 37% of the studies reported on attempts to systematically evaluate the PBL approach. While feedback tends to be positive, it is invariably mixed as expected. Some students respond and flourish with PBL; others prefer a familiar didactic approach. In most cases, however, this represented a student's first exposure to a PBL environment. In many cases, it was also the instructor's first attempt at delivering a PBL course. In combination, this would increase the probability of a student's experience being a negative one.

It is concluded that a more systematic approach to the evaluation of the PBL approach in computing courses is required. This would fulfil two functions. In the first instance, it would address the major question as to whether PBL is a particularly apt paradigm for the teaching of computing in general. Secondly, it would enable greater insights into its applicability in a sub-discipline or individual topics, for example, software engineering.

4.4. Were the problems used in PBL courses documented?

Only 19% of the papers included some sample problems and discussed some of the issues in their design and implementation. While disappointing, it must be acknowledged that space limitations in printed proceedings may be a factor here; nevertheless, alternative methods of presenting individual problems must be identified, possibly using supplements or appendices accessible via appropriate digital repositories.

Good problems are a prerequisite to successful PBL. Constructing such problems is time-consuming and a non-trivial task, demanding significant reflection on the part of the PBL practitioner, and can only be validated in the class room. Experience can make this process easier; however, for those new to PBL, defining suitable scoped problems can be a daunting task. Thus the availability of sample problems, illuminated with some of the factors influencing their design, would be an invaluable resource.

4.5. At what levels has PBL been harnessed?

PBL has been adopted at all stages throughout computing undergraduate and postgraduate curricula (Table II). Over 37% of the described case studies involve first year students. In other levels, the adoption of PBL is far lower. It might be expected that at postgraduate level, the use of PBL would be greater, given the relative maturity of the students; however, this does not seem to be the case.

Why PBL dominates at Level 1 may be guessed at. Keeping in mind the motivations for adopting PBL discussed in Section 4.2, in the case of Level 1, studies specified performance issues, pedagogy and acquiring a better skill sets. It is likely that at the other levels, similar factors contributed. A further explanation may be that as students are making the significant transition from high school education to university, this may be perceived as a favourable opportunity to change their mindset about what university education is about, and expose them to alternative approaches to learning.

4.6. What topics are being covered by a PBL approach?

PBL has been harnessed to teach a variety of topics pertinent to the average computing curriculum. Software engineering and programming courses have been the most popular; however, a range of other topics including some that may be viewed as abstract in nature have been covered via PBL. If the ACM Computer Science Curriculum [ACM 2008] is perceived as archetypical, 10 of the 14 specified knowledge areas have been addressed to some degree. Whether these are covered to the required depth and breadth is open to question. The four outstanding knowledge areas where the harnessing of PBL is still outstanding includes Computational Science, Social and Professional Issues, Information Management and Graphics & Visual Computing.

Table II. How PBL has been adopted from a Topic and Level perspective, for the 59 course-level studies identified in the Category Analysis

Topic	Level 1	Level 2	Level 3	Level 4	Level 5
Artificial Intelligence	1				1
Computer Architecture	1	1		1	1
Computer Networks			1		1
Computer Science	2				
Data Mining				1	
Data Structures		1			
Embedded Systems	1			1	
HCI			2	1	1
Information Technology		2			1
Intelligent Systems					1
MIS	1				
Operating Systems			2		
Parallel Computing				2	
Programming	11	1		1	
Real-time Systems		1	1		
Software Design				1	1
Software Engineering	2		4	3	2
Systems Design	1				
Theoretical CS		2		1	
Total	20	8	10	12	9

5. CONCLUSION

The penetration of PBL into the average computing curricula is shallow. In many cases, faculty members are working in isolation, and face the formidable challenge of introducing PBL into curricula that remain essentially didactic. In some cases, it is a measure of last resort in an effort to improve retention or improve performance. Nevertheless, lecturers' and students' experiences are broadly positive. It is probable that the adoption of PBL in computing will continue in an ad-hoc and random fashion. This situation is unlikely to change unless the key actors - students, teachers, and key stakeholders in society, perceive that PBL offers quantifiable benefits that other approaches do not. Experience of employers with graduates who have experience of PBL would be of particular interest.

If PBL is to succeed in the broad computing education spectrum, practitioners must adopt a more systematic approach towards its adoption and validation. Motivations, objectives, learning outcomes and graduate attributes must be clearly defined. An agreed approach to measuring and validating the effectiveness of PBL must be identified. Innovative and challenging problems must be conceived, and validated. Finally, practitioners must be prepared to share their experiences in appropriate fora.

Overtime, there continues to be an increasing incidence of institutions where entire curricula are delivered through PBL. This offers a significant opportunity for CS PBL practitioners to evaluate PBL in contexts where PBL is the norm.

APPENDIX

The corpus assembled for this analysis is listed here; papers are subdivided by level, and then listed alphabetically.

A. LEVEL 1 TOPICS

- AMBRÓSIO, A.P. AND COSTA, F.M. (2010) Evaluating the impact of PBL and tablet PCs in an algorithms and computer programming course. In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education (SIGCSE '10)*. 495-499.
- BAKAR, M. S., AND SHAIKH AB RAHMAN, S. N. 2005. A Kick Start in implementation of PBL in computer programming. *Proceedings of the 2005 Regional Conference on Engineering Education (RCEE 2005)*.
- BARG, M., FEKETE, A., GREENING, T., HOLLANDS, O., KAY, J., KINGSTON, J. H., AND CRAWFORD, K. 2000. Problem-based learning for foundations computer science courses. *Comp. Sci. Educ.* 10, 2, 1-20.
- BEAMONT C AND FOX C. 2003. Learning Programming: Enhancing Quality through problem based learning. *Proceedings of the 4th Annual LTSN-ICS Conference*. 241-242.
- BENTLEY, J. 2002. Introductory systems analysis and design: A problem-based learning approach. In *Proceedings of the 13th Australasian Conference on Information Systems (ACIS2002)*. Paper 47.
- CLARKE, S., THOMAS, R., AND ADAMS, M. 2005. Developing case studies to enhance student learning. In *Australasian Computing Education: Proceedings of the Seventh Australasian Computing Education Conference (ACE2005)*. 101-108.
- DUKE, R., SALZMAN, E., BURMEISTER, J., POON, J., AND MURRAY, L. 2000. Teaching programming to beginners - choosing the language is just the first step. In *Proceedings of the Australasian Conference on Computing Education (ACE2000)*. 79-86.
- FEKETE, A., GREENING, T., AND KINGSTON, J. 1998. Conveying technical content in a curriculum using problem based learning. In *Proceedings of the 3rd Australasian Conference on Computer Science Education (ACSE '98)*. 198-202.
- GARCÍA-FAMOSO, M. 2005. Problem-based learning: a case study in computer science. In *Proceedings of the International Conference on Multimedia and Information and Communication Technologies in Education (m-ICTE 2005)*. 817-821.
- GREENING, T., KAY, J., KINGSTON, J. H., AND CRAWFORD, K. 1996. Results of a PBL trial in first-year computer science. In *Proceedings of the 2nd Australasian Conference on Computer Science Education (ACSE '97)*. 201-206.
- HELIOTIS, J. AND ZANIBBI, R. 2011. Moving away from programming and towards computer science in the CS first year. *J. Comput. Small Coll.* 26, 3, 115-125.
- MASKELL, D.L. AND GRABAU, P.J. 1998. A multidisciplinary cooperative problem-based learning approach to embedded systems design. *IEEE Transactions on Education*, 41, 2, 101-103.
- MYKYTYN, K., PEARSON, A., SOUREN, P., AND MYKYTYN, P. 2008. The use of problem-based learning to enhance MIS education. *Decision Sciences - Journal of Innovative Education*, 6, 1, 89113.
- NUUTILA, E., TÖRMÄ, S., KINNUNEN, P, AND MALMI, L. 2008. Learning programming with the PBL method – Experiences on PBL Cases and Tutoring. In *Reflections on the Teaching of Programming, Lecture Notes In Computer Science*, 4821. 47-67.
- O'KELLY, J. 2005. Designing a hybrid PBL course: A case study of first year computer science in NUI Maynooth. *Handbook of Enquiry and Problem Based Learning*. 43-53.
- PENG, W. 2010. Practice and experience in the application of problem-based learning in computer programming course. In *Proceedings of the International Conference on Educational and Information Technology (ICEIT)*. 170-172.
- PEREIRA, H.B. DE B., ZEBENDE, G.F. AND MORET, M. A. 2010. Learning computer programming: Implementing a fractal in a turing machine. *Comput. Educ.* 55, 2, 767-776.
- QIU, M. AND CHEN, L. 2010. A problem based learning approach to teaching an advanced software engineering course. In *Proceedings of the Second International Workshop on Education Technology and Computer Science (ETCS)*. 252-255.
- RICHARDSON, I. AND DELANEY, Y. 2009. Problem based learning in the software engineering

classroom. In *Proceedings of the 22nd Conference on Software Engineering Education and Training (CSEET '09)*. 174-181.

SHEN, V.R.L. AND JUANG, T-Y. 2006. A web and problem-based learning system in artificial intelligence. In *Proceedings of the 5th IEEE/ACIS International Conference on Computer and Information Science and 1st IEEE/ACIS International Workshop on Component-Based Software Engineering, Software Architecture and Reuse (ICIS-COMSAR'06)*. 299-304

B. LEVEL 2

BEDNARIK, R. 2004. Problem-based learning in teaching theoretical computer science. In *Proceedings of the International Conference on Engineering Education and Research Conference (iCEER 2004)*. 801-807.

GÜLSEYEN, S. 2004. Problem-based learning application to teach IT skills to prospective teachers in a Faculty of Education. In *Proceedings of the Fifth International Conference on Information Technology Based Higher Education and Training (ITHET 2004)*. 187-191.

HAMILTON, M., HARLAND, J., AND PADGHAM, L. 2003. Experiences in teaching computing theory via aspects of problem-based learning. In *Proceedings of the Fifth Australasian Conference on Computing Education*, 20. 207-211.

LAMBRIX, P. AND KAMKAR, M. 1998. Computer science as an integrated part of engineering education. In *Proceedings of the 6th Annual Conference on the Teaching of Computing and the 3rd Annual Conference on integrating Technology into Computer Science Education: Changing the Delivery of Computer Science Education (ITiCSE '98)*. 153-156.

LENNARTSSON, B. AND SUNDIN, E. 2002. Experience from a course aiming at understanding system development with focus on system design and integration. In *Proceedings of the 32nd Annual Conference on Frontiers in Education (FIE'02)*. 1-6.

MARTINS, C.A.P.S., CORRÊA, J.B.T., GÓES, L.F.W., RAMOS, L.E.S. AND MEDEIROS, T.H. 2002. A new learning method of microprocessor architecture. In *Proceedings of the 32nd Annual Frontiers in Education Conference (FIE 2002)*, 3. 16-21.

MATZKO, S. AND DAVIS, T.A. 2008. A graphics-based approach to data structures. In *Proceedings of the 13th Annual Conference on innovation and Technology in Computer Science Education (ITiCSE '08)*. 109-113.

RYOO, J., FONSECA, F., AND JANZEN, D.S. 2008. Teaching object-oriented software engineering through problem-based learning in the context of game design. In *Proceedings of the 21st Conference on Software Engineering Education and Training (CSEET '08)*. 137-144.

C. LEVEL 3

BRODIE, L., ZHOU, H., AND GIBBONS, A. 2008. Steps in developing an advanced software engineering course using problem based learning. *Engineering Education: Journal of the Higher Education Academy Engineering Subject Centre*, 3, 1, 2-12.

HUANG, H. AND YANG, D. 2008. Teaching design patterns: A modified PBL approach. In *Proceedings of the 9th international Conference For Young Computer Scientists (ICYCS)*. 2422-2426.

MISHRA, V.K., MISHRA, M., AND SHARMA, H.R. 2010. Active learning environment: Gain the most from PBL for teaching virtual machine module on Operating System. In *Proceedings of the First International Conference on Parallel Distributed and Grid Computing (PDGC)*. 378-382.

MITCHELL G.G. AND DELANEY J.D. 2004. An assessment strategy to determine learning outcomes in a software engineering problem-based learning course. *Int. J. Eng. Educ.* 20, 3, 494502.

NADJM-TEHRANI, S. 1997. Towards real-time systems education with PBL. In *Proceedings of the 2nd International Workshop on Real-Time Systems Education (RTEW'97)*. 39- 48.

NOOR, R. M. AND HUSSIN, N. 2004. First experience in implementing PBL for network design and management course. *Journal of Problem-Based Learning*, 2, 1, 11-18.

VAT, K. H. 2001. Teaching HCI with scenario-based design: the constructivist's synthesis. In

Proceedings of the 6th Annual Conference on innovation and Technology in Computer Science Education (ITiCSE '01). 9-12.

VAT, K.H. 2004. Toward a learning organization model for student empowerment: A Teacher-Designer's Experience as a Coach by the Side. In *Proceedings of the IADIS International Conference on Cognition and Exploratory Learning in Digital Age (CELDA2004)*. 131-140.

WANG, Y. DU, H. AND HAO, Y. 2010. A case study of problem based learning instruction design under web-based environment. In *Proceedings of the 2nd International Conference on Education Technology and Computer (ICETC2010)*. 586-591.

YI-RAN, H., CHENG, Z., FENG, Y. AND MENG-XIAO, Y. 2010. Research on teaching operating systems course using problem-based learning. In *Proceedings of the 5th International Conference on Computer Science and Education (ICCSE)*. 691-694.

D. LEVEL 4

ARMAREGO, J. 2002. Advanced software design: a case in problem-based learning. In *Proceedings of the 15th Conference on Software Engineering Education and Training (CSEET '02)*. 44-54.

DUNLAP, J.C. 2005. Problem-based learning and self-efficacy: How a capstone course prepares students for a profession. *Educational Technology Research and Development*. 53, 1, 66-85.

FITZPATRICK, C. 2008. A Problem-Based Learning (PBL) module on electronics and the environment. *IEEE International Symposium on Electronics and the Environment (ISEE 2008)*. 1-6.

GARCIA-ROBLES, R., DIAZ-DEL-RIO, F., VICENTE-DIAZ, S., AND LINARES-BARRANCO, A. 2009. An eLearning standard approach for supporting PBL in computer engineering. *IEEE Transactions on Education*, 52, 3, 328-339.

HÄMÄLÄINEN, W. 2004. Problem-based learning of theoretical computer science. In *Proceedings of the 34th Annual Conference on Frontiers in Education (FIE'04)*. 20-23.

KAY, J. AND KUMMERFELD, B. 1998. A problem-based interface design and programming course. In *Proceedings of the Twenty-Ninth SIGCSE Technical Symposium on Computer Science Education (SIGCSE '98)*. 194-197.

KIRSCH, R. P. 1996. Teaching OLE automation: a problem-based learning approach. In *Proceedings of the Twenty-Seventh SIGCSE Technical Symposium on Computer Science Education (SIGCSE '96)*. 68-72.

MOEN, P. (2006) Problem-based learning in data mining. In *Proceedings of the Fourth International Workshop on Teaching, Learning and Assessment of Databases (TLAD'06)*. 1-5.

Ó CINNÉIDE, M. AND TYNAN, R. 2004. A problem-based approach to teaching design patterns. *SIGCSE Bull.* 36, 4, 80-82.

ORIOGUN, P., FERGUSON, A., LUDMANN, M., MAISURIA, N. AND SWHE, YU. M. 2001. Using the problem-based learning grid in the management of a software engineering module : towards an effective learning intervention for final year undergraduates. In *The power of problem based learning: Proceedings of the 3rd Asia Pacific Conference on PBL*. 99-110.

ROVER, D.T., MERCADO, R.A., ZHANG, Z., SHELLEY, M.C. AND HELVICK, D.S. 2008. Reflections on teaching and learning in an advanced undergraduate course in embedded systems. *IEEE Transactions on Education*, 51, 3, 400-412.

YANG, J., XIE, Y., GENG, Q., WANG, J., AND BAO, N. 2008. Using cP²BL in teaching multi-core related contents. In *Proceedings of the 9th international Conference For Young Computer Scientists (ICYCS)*. 2449-2453.

E. LEVEL 5

CAVEDON, L., HARLAND, J., AND PADGHAM, L. 1996. Problem-based learning with technological support in an AI subject: description and evaluation. In *Proceedings of the 2nd Australasian Conference on Computer Science Education (ACSE '97)*, 2. 191-200.

CHEONG, F. 2008. Using a problem-based learning approach to teach an intelligent systems course.

Journal of Information Technology Education, 7, 47-60.

DOS SANTOS, S.C., DA CONCEIÇÃO MORAES BATISTA, M., CAVALCANTI, A.P.C., ALBUQUERQUE, J.O. AND MEIRA, S.R.L. 2009. Applying PBL in software engineering education. In *Proceedings of the 22nd Conference on Software Engineering Education and Training (CSEET '09)*. 182-189.

FABIANI, E. 2009. Experiencing a problem-based learning approach for teaching reconfigurable architecture design. *International Journal of Reconfigurable Computing*, Article ID 923415.

LAI, Y. 2005. Teaching computer applications to pre-school teachers through problem-based learning approach. *SIGCSE Bull.* 37, 4, 89-92.

LINGE, N. AND PARSONS, D. 2006. Problem-based learning as an effective tool for teaching computer network design. *IEEE Transactions on Education*, 49, 1, 5-10.

PURAO, S., VAISHNAVI, V., BAGBY, J., BORTHICK, F., CAMERON, B., LENZE, L., SAWYER, S., SUEN, H. AND WELKE, R. 2008. Using problems to learn service-oriented computing. In *Proceedings of the 2008 IEEE Congress on Services - Part I (SERVICES '08)*. 33-36.

RICHARDSON, I. AND DELANEY, Y. 2010. Software quality: From theory to practice. In *Proceedings of the Seventh International Conference on the Quality of Information and Communications Technology (QUATIC '10)*. 150-155.

SCHULTZ, N. AND CHRISTENSEN, H. 2004. Seven-step problem-based learning in an interaction design course. *European Journal of Engineering Education*, 29, 4, 533-541.

F. THE CURRICULUM AND SOCIETY PERSPECTIVE

BUNCH, J. 2009. A Constructivist approach to teaching web development in post-secondary vocational settings. *Journal of Information Technology Education*, 8, 257-271.

FEE, S. B. AND HOLLAND-MINKLEY, A. M. 2010. Teaching computer science through problems, not solutions. *Computer Science Education*, 20, 2, 129-144.

GIBSON, J.P. 2009. Weaving a formal methods education with problem based learning. In *Leveraging Applications of Formal Methods Verification and Validation, Communications in Computer and Information Science*, 17. 460-472.

HOGAN, J. M. AND THOMAS, R. 2005. Developing the software engineering team. In *Proceedings of the 7th Australasian Conference on Computing Education*, 42. 203-210.

ACKNOWLEDGMENTS

The support of Dr Terry Barrett, Centre for Teaching and Learning, University College Dublin, and CLARITY: Centre for Sensor Web Technologies, is greatly appreciated. The insightful comments of the anonymous reviewers are gratefully acknowledged. This work is partially supported by Science Foundation Ireland under grant 07/CE/I1147.

REFERENCES

ACM 2005. Computing Curricula 2005: The Overview Report. <http://www.acm.org/education/curricula-recommendations>.

ACM 2008. Computer Science Curriculum 2008: An Interim Revision of CS 2001. <http://www.acm.org/education/curricula-recommendations>.

BARRETT, T. 2005. Understanding problem-based learning. In *Handbook of Enquiry & Problem Based Learning*, T. Barrett, I. MacLabhrainn, and H. Fallon, Eds. AISHE, Dublin, 14-25.

BARROWS, H. S. 1996. Problem-based learning in medicine and beyond: A brief overview. *New Directions for Teaching and Learning* 1996, 68, 3-12.

BEAUMONT, C., SACKVILLE, A., AND CHENG, C. S. 2004. Identifying Good Practice in the use of PBL to teach computing. *ITALICS e-journal* 3, 1, 1-19.

BERGLUND, A. AND LISTER, R. 2010. Introductory programming and the didactic triangle. In *Proceedings of the Twelfth Australasian Conference on Computing Education - Volume 103*. ACE '10. Australian Computer Society, Inc., Darlinghurst, Australia, 35-44.

- BOUD, D. AND FELETTI, G. 1997. *The challenge of problem-based learning*. Kogan press, London.
- BUNCH, J. M. 2009. A constructivist approach to teaching web development in post-secondary vocational settings. *Journal of Information Technology Education* 8, 257–271.
- CARRERO, E., GOMAR, C., PENZO, W., AND RULL, M. 2007. Comparison between lecture-based approach and case/problem-based learning discussion for teaching pre-anaesthetic assessment. *European Journal of Anaesthesiology* 24, 12, 1008–1015.
- CLARKE, S., THOMAS, R., AND ADAMS, M. 2005. Developing case studies to enhance student learning. In *Proceedings of the 7th Australasian conference on Computing education - Volume 42*. ACE '05. Australian Computer Society, Inc., Darlinghurst, Australia, 101–108.
- DENNING, P. J. 2005. Is computer science science? *Commun. ACM* 48, 27–31.
- DOLMANS, D. H. J. M., DE GRAVE, W., WOLFHAGEN, I. H. A. P., AND VAN DER VLEUTEN, C. P. M. 2005. Problem-based learning: future challenges for educational practice and research. *Medical Education* 39, 7, 732–741.
- DUCH, B. 2001. Writing problems for deeper understanding. In *The power of problem-based learning: A practical how to for teaching undergraduate courses in any discipline*, B. Duch, S. Groh, and D. Allen, Eds. Stylus Publishing, Sterling, VA, 47–53.
- EDELSON, D. C., GORDIN, D. N., AND PEA, R. D. 1999. Addressing the challenges of inquiry-based learning through technology and curriculum design. *The Journal of the Learning Sciences* 8, 391–450.
- ELLIS, A., CARSWELL, L., BERNAT, A., DEVEAUX, D., FRISON, P., MEISALO, V., MEYER, J., NULDEN, U., RUGELJ, J., AND TARHIO, J. 1998. Resources, tools, and techniques for problem based learning in computing. *SIGCUE Outlook* 26, 41–56.
- FABIANI, E. 2009. Experiencing a problem-based learning approach for teaching reconfigurable architecture design. *Int. J. Reconfig. Comput.* 2009, 9:1–9:1.
- FEE, S. B. AND HOLLAND-MINKLEY, A. M. 2010. Teaching computer science through problems, not solutions. *Computer Science Education* 20, 2, 129–144.
- FINCHER, S. AND PETRE, M. 2004. *Computer Science Education Research*. Taylor & Francis, London, UK.
- GIBSON, J. P. 2009. Weaving a formal methods education with problem-based learning. In *Leveraging Applications of Formal Methods, Verification and Validation*, T. Margaria and B. Steffen, Eds. Communications in Computer and Information Science Series, vol. 17. Springer, Berlin Heidelberg, 460–472.
- HÄMÄLÄINEN, W. 2004. Problem-based learning of theoretical computer science. In *Proceedings of the 34th Annual Conference on Frontiers in Education*. Vol. 3. IEEE Press, Piscataway, NJ, USA, 20–303.
- HOGAN, J. M. AND THOMAS, R. 2005. Developing the software engineering team. In *Proceedings of the 7th Australasian conference on Computing education - Volume 42 (ACE '05)*. Australian Computer Society, Inc., Darlinghurst, Australia, 203–210.
- JOY, M., SINCLAIR, J., SUN, S., SITTHIWORACHART, J., AND LÓPEZ-GONZÁLEZ, J. 2009. Categorising computer science education research. *Education and Information Technologies* 14, 105–126.
- KANSANEN, P. AND MERI, M. 1999. The didactic relation in the teaching-studying-learning process. *TNTEE Publications* 2, 1, 107–116.
- KINNUNEN, P., MEISALO, V., AND MALMI, L. 2010. Have we missed something?: identifying missing types of research in computing education. In *Proceedings of the Sixth international workshop on Computing education research (ICER '10)*. ACM, New York, NY, USA, 13–22.
- KIRSCHNER, P. A., SWELLER, J., AND CLARK, R. E. 2006. Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching. *Educational Psychologist* 41, 2, 75–86.
- KRAJCIK, J. S. AND BLUMENFELD, P. C. 2006. Project-based learning. In *The Cambridge handbook of the learning sciences*, R. K. Sawyer, Ed. Cambridge University Press, New York, 317–334.
- MATZKO, S. AND DAVIS, T. A. 2008. A graphics-based approach to data structures. *SIGCSE Bull.* 40, 109–113.
- MITCHELL, G. AND DELANEY, J. 2004. An assessment strategy to determine learning outcomes in a software engineering problem-based learning course. *International Journal of Engineering Education* 20, 3, 494–502.
- MÜLLER, V. C. 2008. What a course on philosophy of computing is not. *APA Newsletter on Philosophy and Computers* 8, 1, 36–38.
- PEREIRA, H. B. D. B., ZEBENDE, G. F., AND MORET, M. A. 2010. Learning computer programming: Implementing a fractal in a turing machine. *Computers & Education* 55, 2, 767–776.
- QIU, M. AND CHEN, L. 2010. A problem-based learning approach to teaching an advanced software engineering course. In *Second International Workshop on Education Technology and Computer Science (ETCS)*. Vol. 3. IEEE Press, Piscataway, NJ, USA, 252–255.

- RICHARDSON, I. AND DELANEY, Y. 2010. Software quality: From theory to practice. In *Proceedings of the 2010 Seventh International Conference on the Quality of Information and Communications Technology (QUATIC '10)*. IEEE Computer Society, Washington, DC, USA, 150–155.
- SANSON-FISHER, R. AND LYNAGH, M. 2005. Problem-based learning: A dissemination success story. *Medical Journal of Australia* 183, 5, 258–260.
- SCHMIDT, H., LOYENS, S. M., VAN GOG, T., AND PAAS, F. 2007. Problem-based learning is compatible with human cognitive architecture: Commentary on kirschner, sweller, and clark (2006). *Educational Psychologist* 42, 2, 91–97.
- SCHMIDT, H. G. 1983. Problem-based learning: Rationale and description. *Medical education* 17, 1, 11–16.
- SHEARD, J., SIMON, S., HAMILTON, M., AND LÖNNBERG, J. 2009. Analysis of research into the teaching and learning of programming. In *Proceedings of the fifth international workshop on Computing education research workshop (ICER '09)*. ACM, New York, NY, USA, 93–104.
- SIMON. 2007. A classification of recent australasian computing education publications. *Computer Science Education* 17, 3, 155–169.
- STROBEL, J. AND VAN BARNEVELD, A. 2009. When is pbl more effective? a meta-synthesis of meta-analyses comparing pbl to conventional classrooms. *Interdisciplinary Journal of Problem-based Learning* 3, 1, 44–58.
- TIWARI, A., LAI, P., SO, M., AND YUEN, K. 2006. A comparison of the effects of problem-based learning and lecturing on the development of students' critical thinking. *Medical Education* 40, 6, 547–554.
- VESSEY, I., RAMESH, V., AND GLASS, R. L. 2005. A unified classification system for research in the computing disciplines. *Inf. Softw. Technol.* 47, 245–255.
- WALKER, A. AND LEARY, H. 2009. A problem based learning meta analysis: Differences across problem types, implementation types, disciplines, and assessment levels. *Interdisciplinary Journal of Problem-based Learning* 3, 1, 12–43.
- WANG, Y., DU, H., AND HAO, Y. 2010. A case study of problem-based learning instruction design under web-based environment. In *Second International Conference on Education Technology and Computer (ICETC)*. IEEE Press, Piscataway, NJ, USA, 586–591.
- WEISS, R. E. 2003. Designing problems to promote higher-order thinking. *New Directions for Teaching and Learning* 2003, 95, 25–31.