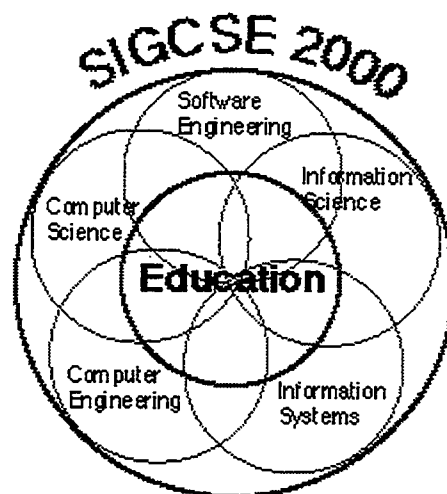


The Proceedings of the Thirty-first SIGCSE Technical Symposium on Computer Science Education



Austin, Texas
March 8-12, 2000

Symposium Co-Chairs:

Boots Cassel, Villanova University

Nell Dale, The University of Texas at Austin

Program Chair:

Henry Walker, Grinnell College

Proceedings:

Susan Haller, University of Wisconsin - Parkside

Sponsored by:

The Association for Computing Machinery

Special Interest Group on Computer Science Education



Table of Contents

Message from the SIGCSE Chair	iii
Message from the Symposium Chairs	iii
Letter from the Program Chair	iv
Message from the Proceedings Chair	v
SIGCSE 2000 Symposium Committee	v
SIGCSE 2001 Contact	v
The SIGCSE 2000 Doctoral Consortium	vi
Recipients of SIGCSE Awards	viii
Roster of Referees	ix
Table of Contents	xv

Papers

Graphics/Visualization

Thur., 10:30-12:00, Ballroom B

<i>A Visualization System for Multithreaded Programming</i>	1
Ching-Kuang Shene, Xianglong Huang, Steve Carr, and Michael Bedy, Michigan Technological University	
<i>Increasing Visualization and Interaction in the Automata Theory Course</i>	6
Susan Rodger and Ted Hung, Duke University	
<i>CUPV - A Visualization Tool for Generated Parsers</i>	11
Alan Kaplan and Denise Shoup, Clemson University	

Algorithms

Thur., 10:30-12:00, San Antonio

<i>Design and Analysis of Algorithms Reconsidered</i>	16
Anany Levitin, Villanova University	
<i>Design Patterns for Lazy Evaluation</i>	21
Stephen Wong, Oberlin College and Dung "Zung" Nguyen, Rice University	
<i>Cogito, Ergo Sum! Cognitive Processes of Students Dealing with Data Structures</i>	26
Dan Aharoni, Technion, Israel Institute of Technology	

Database

Thur., 10:30-12:00, Sabine

<i>Teaching Database Management Systems with Java</i>	31
Matthew Merzbacher, Mills College	
<i>Enhancing the Value of a Project in the Database Course</i>	36
Mary Ann Robbert, Bentley College	
<i>The Next Decade of the Database Course: Three Decades Speak to the Next</i>	41
Frederick Springsteel, University of Missouri at Columbia; Mary Ann Robbert, Bentley College; and Catherine Ricardo, Iona College	

Curricular Issues

Thur., 1:30-3:00, Sabine

<i>Powers of 10: The Case for Changing the First Course in Computer Graphics</i>	46
Steve Cunningham, California State University Stanislaus	
<i>Curriculum Development in Internet and Multimedia Technology</i>	50
Sylvia Sorkin, Kathleen Harmeyer, Donna Tupper, and Barbara Mento, Community College of Baltimore County-Essex Campus	
<i>An Innovative University Program on Management and ICT</i>	55
Karel Lemmen, Open University of the Netherlands; Wim Smit, Hague College of Higher Professional Education; and Fred Mulder, TSM Business School	

Distributed/Parallel Computing

Thur., 1:30-3:00, San Antonio

<i>"Alfonse, You Have a Message!"</i>	60
Stephen Hartley, Richard Stockton College	

<i>Parallel Computing to Start the Millennium</i>	65
Christopher Nevison, Colgate University; Nan Schaller, Rochester Institute of Technology; and Joel Adams, Calvin College	
<i>Teaching Parallel Algorithms With Process Topologies</i>	70
Chris McDonald and Kamran Kazemi, The University of Western Australia	
CS 1	Thur., 1:30-3:00, Pecos
<i>Design Early Considered Harmful: Graduated Exposure to Complexity and Structure Based on Levels of Cognitive Development</i>	75
Duane Buck and David J. Stucki, Otterbein College	
<i>Programming Patterns and Design Patterns in the Introductory Computer Science Course</i>	80
Viera K. Proulx, Northeastern University	
<i>Conservatively Radical Java in CS1</i>	85
Stuart Reges, University of Arizona	
Artificial Intelligence	Thur., 3:30-5:00, Pecos
<i>Application of Security to the Computing Science Classroom - Lessons Learned</i>	90
Rayford Vaughn, Mississippi State University	
<i>Iambic IBM AI: The Palindrome Discovery AI Project</i>	95
Eric Siegel, Columbia University	
<i>Machine Learning in the Liberal Arts Curriculum</i>	100
Clare Congdon, Colby College	
Web-based Technology	Thur., 3:30-5:00, Ballroom B
<i>Web-Based Lab Manual for CS 1: An Experiment</i>	105
Thomas Cheatham, Middle Tennessee State University	
<i>JHAVE - An Environment to Actively Engage Students In Web-based Algorithm Visualizations</i>	109
Thomas Naps, Laura Norton, and James Eagan, Lawrence University	
<i>Remote Logging in Java using Jeli: A Facility to Enhance Development of Accessible Educational Software</i>	114
Steven Robbins, University of Texas at San Antonio	
Communication Skills	Thur., 3:30-5:00, Sabine
<i>Fostering and Assessing Communication Skills in the Computer Science Context</i>	119
Mark Michael, King's College	
<i>Experiments with the Use of Popular Press in the Computer Science Curriculum</i>	124
Paula Gabbert and Kevin Treu, Furman University	
<i>Getting Participation Through Discussion</i>	129
Martyn Clark, University of Leeds	
Course Pedagogy	Fri., 8:00-10:00, Sabine
<i>Experiences in Using Animations in Introductory Computer Science Lectures</i>	134
Guido Rößling, and Bernd Freisleben, University of Siegen	
<i>PILOT: An Interactive Tool for Learning and Grading</i>	139
Stina Bridgeman and Roberto Tamassia, Brown University and Stephen Kobourov and Michael Goodrich, Johns Hopkins University	
<i>Supporting Reflection in Introductory Computer Science</i>	144
Alan Fekete, Kapila Wimalaratne, Jeffrey Kingston, and Judy Kay, University of Sydney	
<i>Pedagogically Sound Responses to Economic Rationalism</i>	149
Tony Greening, University of Sydney	
Software Engineering	Fri., 8:00-10:00, Wedgwood
<i>Empirical Results of a Software Engineering Curriculum Incorporating Formal Methods</i>	157
Ann Sobel, Miami University	
<i>Algorithma 99: An Experiment in Reusability & Component-Based Software Engineering</i>	162
Arturo Concepcion, Allan Knight, and Nathan Leach, California State University at San Bernardino	

<i>Software Quality: A Curriculum Postscript?</i>	167
Thomas Hilburn and Massood Towhidnejad, Embry-Riddle Aeronautical University	
<i>Evaluating Individuals in Team Projects</i>	172
Dawn Wilkins and Pamela Lawhead, The University of Mississippi	
CS 1/2	Fri., 8:00-10:00, Trinity
<i>Encouraging Students in Large Classes</i>	176
Roy Andersson, Lund Institute of Technology/Lund University and Torgny Roxå, Lund University	
<i>Introducing Computer Science Using a Breadth-First Approach and Functional Programming</i>	180
Scott Vandenberg, Siena College, and Michael Wollowski, Rose-Hulman Institute of Technology	
<i>Toolkits in First Year Computer Science: A Pedagogical Imperative</i>	185
Richard Rasala, Northeastern University	
<i>Real-World Program Design in CS2 – The Roles of a Large-Scale, Multi-Group Class Project</i>	192
Samuel Rebelsky and Clif Flynt, Grinnell College	
CS Education Research	Fri., 8:00-10:00, San Antonio
<i>Programming in Java: Student-Constructed Rules</i>	197
Ann Fleury, Aurora University	
<i>Empirical Investigation throughout the CS Curriculum</i>	202
David Reed and Grant Braught, Dickinson College and Craig Miller, DePaul University	
<i>The Effect of a Constructivist Learning Environment for Field-Dependent/Independent Students on Achievement in Introductory Computer Programming</i>	207
David Gibbs, University of Wisconsin–Stevens Point	
<i>Evaluating Programming Ability in an Introductory Computer Science Course</i>	212
Albert Chamillard and Kimberly Braun, U.S. Air Force Academy	
Social and Ethical Issues	Fri., 11:00-12:00, Pecos
<i>Video Resources For Use In Teaching Ethics and Computing</i>	217
Kevin Bowyer, University of South Florida	
<i>A Primer for Applying Service Learning to Computer Science</i>	222
Pete Sanderson and Ken Vollmar, Southwest Missouri State University	
First Year Courses	Fri., 11:00-12:00, Ballroom B
<i>A fundamentals-based Curriculum for First Year Computer Science</i>	227
Ian Sanders and Conrad Mueller, University of the Witwatersrand	
<i>Instrumentation: A Multi-Science Integrated Sequence</i>	232
Herbert Tesser, Gary Anderson, and Hisham Al-Haddad, Marshall University	
Object-Oriented Issues	Fri., 11:00-12:00, Trinity
<i>Aristotle and Object-Oriented Programming: Why Modern Students Need Traditional Logic</i>	237
Derek Rayside, University of Waterloo and Gerard Campbell, University of St. Jerome's College	
<i>Myths about Object-Orientation and its Pedagogy</i>	245
John Lewis, Villanova University	
Networks and E-commerce	Fri., 11:00-12:00, Sabine
<i>E-commerce and Computer Science Education</i>	250
Yuzhen Ge, University of the Pacific and Jiangeng Sun, IBM Global Services	
<i>Bugs as Features: Teaching Network Protocols Through Debugging</i>	256
Brad Richards, Vassar College	
Undergraduate Research/Capstones	Fri., 2:00-3:30, Wedgwood
<i>Design and Implementation of Computer Games: A Capstone Course for Undergraduate Computer Science Education</i>	260
Randolph Jones, Colby College	
<i>A Capstone Course for a Computer Information Systems Major</i>	265
Sharon Tuttle, Humboldt State University	

<i>Interdisciplinary Capstone Group Project: Designing Autonomous Race Vehicles</i>	270
Steve Mosiman, and Christoph Hiemcke, Loras College	

Information Technology

Fri., 2:00-3:30, Trinity

<i>Implementing an IT Concentration in a CS Department: Content, Rationale, and Initial Impact</i>	275
Phil Pfeiffer and Terry Countermine, East Tennessee State University	

<i>Fluency in Information Technology: A Second Course for non-CIS Majors</i>	280
Ned Kock, Munir Mandviwalla, and Robert Aiken, Temple University	

<i>A Bachelor of Science in Information Technology: An Interdisciplinary Approach</i>	285
David Spooner, Rensselaer Polytechnic Institute	

CS 1/2

Fri., 2:00-3:30, Ballroom B

<i>Yet More Web Exercises for Learning C++</i>	290
Bruce Elenbogen, University of Michigan-Dearborn; Chris McDonald, University of Western Australia; and Bruce Maxim, University of Michigan-Dearborn	

<i>Strategies for Encouraging Individual Achievement in Introductory Computer Science Courses</i>	295
Eric Roberts, Stanford University	

<i>SAIL: A System for Generating, Archiving, and Retrieving Specialized Assignments Using LaTeX</i>	300
Stephen Kobourov, Johns Hopkins University; Roberto Tamassia and Stina Bridgeman, Brown University; and Michael Goodrich, Johns Hopkins University	

Recursion

Fri., 4:00-5:30, Ballroom B

<i>EROSI - Visualising Recursion and Discovering New Errors</i>	305
Carlisle George, Middlesex University, London	

<i>Recursion in Gradual Steps (Is Recursion Really that Difficult?)</i>	310
J. Ángel Velázquez-Iturbide, Universidad Rey Juan Carlos	

<i>Recursively Speaking: Analyzing Students' Discourse of Recursive Phenomena</i>	315
Dalit Levy and Tami Lapidot, Technion - Israel Institute of Technology	

Operating Systems

Fri., 4:00-5:30, Wedgwood

<i>Operating Systems Projects Built on a Simple Hardware Simulator</i>	320
John Dickinson, University of Idaho	

<i>A Study of Common Pitfalls in Simple Multi-Threaded Programs</i>	325
E Christopher Lewis, University of Washington and Sung-Eun Choi, Los Alamos National Laboratory	

<i>Experimentation with Bounded Buffer Synchronization</i>	330
Steven Robbins, University of Texas at San Antonio	

Distance Education

Sat., 10:30-12:00 Trinity

<i>Distance Education: Law Attempts To Catch Up With Technology (Battle Between Copyright Owners and Academics)</i>	335
Jill Gerhardt-Powals, Richard Stockton College of New Jersey and Matthew Powals, Member of the Bars of New Jersey and New York	

<i>Multimedia: From Topic to Course</i>	343
John Minor Ross, Indiana University Kokomo	

<i>Distance Teaching Workloads</i>	347
Wendy Doube, Monash University	

Object-Orientation in the First Year

Sat., 10:30-12:00 Wedgwood

<i>Checkmate: Cornering C++ Dynamic Memory Errors With Checked Pointers</i>	352
Scott Pike and Bruce Weide, The Ohio State University and Joseph Hollingsworth, Indiana University Southeast	

<i>Object-orientation in Java for Scientific Programmers</i>	357
Judith Bishop, University of Pretoria and Nigel Bishop, University of South Africa	

<i>An Object-Oriented View of Backtracking</i>	362
Bob Noonan, College of William and Mary	

CS Education Research

Sat., 10:30-12:00 Pecos

- Enhancing Student Learning Through On-line Quizzes* 367
Denise Woit and Dave Mason, Ryerson Polytechnic University
- Combining Cooperative Learning and Peer Instruction in Introductory Computer Science* 372
Joe Chase and Edward Okie, Radford University
- Assessing the Benefits of Interactive Prediction Using Web-based Algorithm Animation Courseware* 377
Duane Jarc, Rachelle Heller, and Michael Feldman, The George Washington University

Human-Computer Interaction

Sat., 10:30-12:00 Sabine

- Do Visualizations Improve Program Comprehensibility? Experiments with Control Structure Diagrams for Java* 382
Dean Hendrix, Matthew McKinney, Saeed Maghsoodloo, and James Cross, Auburn University
- Experiences with the Webware, Interfaces and Networking Experimental Laboratory* 387
David Brown, Isabel Cruz, David Finkel, Robert E. Kinicki, and Craig E. Wills, Worcester Polytechnic Institute
- Teaching Software Psychology: Expanding the Perspective* 392
Kam VAT, University of Macau, Macau

Panels

- CS1 and CS2: Foundations of Computer Science and Discrete Mathematics* 397
Moderator: D.S. Tomer, University of Central Arkansas
Thur., 10:30-12:00, Pecos
- Panelists: Doug Baldwin, State University of New York at Geneseo; Carl H. Smith, University of Maryland; Peter B. Henderson, State University of New at Stony Brook; and Venu Vadisigi Southern Polytechnic State University
- Using Large Projects in a Computer Science Curriculum* 399
Moderator: K. Todd Stevens The University of Mississippi
Thur., 10:30-12:00, Wedgwood
- Panelists: John Lewis, Villanova University; Joel Henry, East Tennessee State University; Pamela B. Lawhead, The University of Mississippi; Constance Bland, Mississippi Valley State University; and Mary Jane Peters, Jacksonville State University
- Experiences in Starting Computer Engineering Programs* 401
Moderator: Daniel D. McCracken City University of New York
Thur., 1:30-3:00, Wedgwood
- Panelists: Manuel A. Pérez-Quñones, University of Puerto Rico, Mayaguez; Robert Bryant, Gonzaga University; Fred Springsteel, University of Missouri at Columbia; and Anne-Louise Radimsky, California State University, Sacramento
- The Database Course: What Should/Must be Taught* 403
Moderator: Mary Ann Robbert, Bentley College
Thur., 1:30-3:00, Trinity
- Panelists: Ming Wang, Embry-Riddle Aeronautical University; Mario Guimaraes, Texas A&M University; and Martha E. Myers, Kennesaw State University
- Student Outcomes Assessment: What Works and What Doesn't* 405
Moderator: Sue Fitzgerald, Metropolitan State University
Thur., 1:30-3:00, Ballroom B
- Panelists: Mark Fienup, University of Northern Iowa; Norbert J. Kubilus, The College of New Jersey; Dian Lopez, University of Minnesota, Morris; and Merry McDonald, Northwest Missouri State University
- Concurrency the First Year: Experience Reports* 407
Moderator: Max Hailperin, Gustavus Adolphus College
Thur., 3:30-5:00, Wedgwood
- Panelists: David Arnow, Brooklyn College; Judith Bishop, University of Pretoria; Chester Lund, George Washington University; and Lynn Andrea Stein, Massachusetts Institute of Technology

<i>Proposed Information Systems Accreditation Criteria</i>	409
Moderator: Doris K. Lidtke, Towson University	Thur., 3:30-5:00, San Antonio
Panelists: John Gorgone, Bentley College; Willis King, University of Houston; and Gayle Yaverbaum, Penn State at Harrisburg	
<i>Curriculum 2001: Evaluating the Strawman Representatives of the ACM/IEEE-CS Task Force</i>	411
Moderator: Eric Roberts, Stanford University	Fri., 8:00-10:00, Ballroom B
Panelists: Carl Chang, University of Illinois-Chicago; C. Fay Cover, Pikes Peak Community College; James Cross, Auburn University; Gerald Engel, University of CT-Stamford; and Russ Shackelford, Georgia Institute of Technology	
<i>What's New with ABET/CSAB Integration</i>	413
Moderator: Doris K. Lidtke, Towson University	Fri., 11:00-12:00, Wedgwood
Panelists: Lee Saperstein, President of ABET; Kenneth Martin, University of North Florida; and Della Bonnette, University of Southern Louisiana at Lafayette	
<i>Teaching Advice and Support for New and Adjunct Faculty: Experiences, Policies, and Strategies</i>	414
Moderator: David G. Kay, University of California, Irvine	Fri., 2:00-3:30, Pecos
Panelists: Clare Bates Congdon, Colby College; Sue Fitzgerald, Metropolitan State University; Merle S. King, Kennesaw State University; and Pat Semmes, Trinity University	
<i>Recommendations for Changes in Advanced Placement Computer Science</i>	416
Moderator: Owen Astrachan, Duke University	Fri., 4:00-5:30, Sabine
Panelists: Robert Cartwright, Rice University (representing ACM Ed. Board); Gail Chapman, College Board; David Gries, University of Georgia; Rich Kick, Hinsdale Central (Illinois) High School; Cay Horstmann, San Jose State; Fran Trees, Westfield (New Jersey) High School; Henry Walker, Grinnell College; and Ursula Wolz, College of New Jersey	
<i>IT Programs and CS Departments</i>	417
Moderator: Elliot Koffman, Temple University	Fri., 4:00-5:30, San Antonio
Panelists: Joe Turner, Clemson University; Loren K. Rhodes, Juniata College; Dorothy Deremer, Montclair State University; Curt White, DePaul University; Rebecca Thomas, Marist College; and Chris McDonald, The University of Western Australia	
<i>Undergraduate Computational Science Education</i>	419
Moderator: Angela B. Shiflet, Wofford College	Fri., 4:00-5:30, Pecos
Panelists: Philip Holmes, Princeton University; Chuck Niederriter, Gustavus Adolphus College; Robert M. Panoff, The Shodor Education Foundation, Inc.; and Ernest Sibert, Syracuse University	
<i>Teams Work</i>	421
Moderator: Louise Moses, Mount Union College	Sat.. 10:30-12:00, San Antonio
Panelists: Sally Fincher, University of Kent at Canterbury and James Caristi, Valparaiso University	
<i>The Assimilation of Software Engineering Into the Undergraduate CS Curriculum</i>	423
Moderator: Renee McCauley, University of Southwestern Louisiana	Sat.. 10:30-12:00, Ballroom B
Panelists: Nell Dale, The University of Texas at Austin; Thomas Hilburn, Embry-Riddle Aeronautical University; Susan Mengel, Texas Tech University; and Branson W. Murrill, Virginia Commonwealth University	

Teaching *Software Psychology* : Expanding the Perspective

Kam Hou VAT
Faculty of Science & Technology
University of Macau, Macau
fstkhv@umac.mo

Abstract

This paper describes the curriculum development and teaching experience of a junior core course entitled *Software Psychology*, offered in the undergraduate Software Engineering program at the author's affiliated university. In particular, the pedagogy of problem-based learning is introduced, together with the evolution of the course content. It will also address issues such as resources and facilities needed for the course, and the students' perceived learning as well as the author's lessons learned therein.

1 Introduction

Software Psychology, according to our Faculty (founded in 1989) records, derived its name from the domain of study of human behavior in software engineering. The first suggested textbook in the course back in 1989, was found to be Ben Shneiderman's 1980 edition of *Software Psychology: Human Factors in Computer and Information Systems*. Since 1993, an ongoing tailoring process is in place to deliver an appropriate curriculum, which is composed of a suitable mix of such elements as human factors, user expectations, human-computer interfaces construction, cognitive psychology, and some latest development on user-centered and/or performance-centered design in interactive system development. Yet, the original course title has been retained because any change in course name involves some lengthy bureaucratic approval process in the local Authorities. It is understood that if we intend our students to have better exposure to this field of human-computer interaction (HCI), as it is presently known today, we need more than a continually updated curriculum. We also need an appropriate pedagogy, whose mission is to facilitate students' active learning.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SIGCSE 2000 3/00 Austin, TX, USA
© 2000 ACM 1-58113-213-1/00/0003...\$5.00

And the author has chosen the Problem-Based Learning (PBL) approach [1, 2, 3]. It has been reported that PBL helps develop in students the following characteristics of quality performance:

- high level communications, technological literacy, and informational abilities that enable individuals to gain and apply new knowledge and skills as needed.
- the ability to arrive at informed judgements; namely, to effectively define problems, gather and evaluate information related to those problems, and develop solutions.
- a range of attitudes and dispositions including: flexibility and adaptability; ease with diversity; motivation and persistence; creativity and resourcefulness; ability to work with others in team settings.

2 The PBL Pedagogy

The modern history of PBL began in the early 1970s at the medical school at McMaster University in Canada, and ever since, PBL has been adopted in various fields such as Teaching, Engineering and Management. Pedagogically, the PBL approach uses real-world problems to drive the learning rather than mere lectures with the instillation of subject matter. It acknowledges the possibility of prior knowledge held by the learner. Further knowledge is acquired on a 'need to know' basis, enabling the learner to diagnose one's own learning needs. Knowledge gained is fed back into the problem in an iterative loop, allowing the synthesis of topics and know-how [3]. When applied to the course setting, PBL should encourage students' active participation and develop in them self-directed learning and problem-solving skills while they interact, discuss and share relevant knowledge and experience. More importantly, PBL revolves around a focal problem, group work, feedback, class discussion, skill development and final reporting. The instructor's role is to organize and pilot this cycle of activity, guiding, probing and supporting students' initiatives along the way so as to empower them to be responsible in their own learning.

3 Course Goals

The course entitled *Software Psychology*, is offered annually in the fall semester as a compulsory subject for

Software Engineering majors. The goals of the class include the following:

- to help students become HCI-literate by developing fundamental understanding of HCI in relation to human factors, usability engineering, cognitive psychology, and computer science.
- to encourage students to formulate and express their views on user interface design of interactive systems, through project development, written work, oral presentations and classroom discussions.
- to raise students' awareness of the HCI impact on computer industry, and the wide-spread focus of HCI from human factors, to usability engineering, to user-centered design, in constructing systems that support human activity.

4 Course Content

The fall semester of about 15 weeks' duration, is structured into three sessions of loosely 5 weeks each. The course content is then arranged into three sessions as follows:

4.1 First 5-Week Session

The first session covers the basic foundations of HCI, introducing the idea that HCI is concerned with understanding, designing, evaluating and implementing interactive computing systems for human use. And it draws from the knowledge and methods of many different disciplines, chiefly computer science, cognitive psychology, social sciences and human factors. Next we try to interpret HCI as a model where people, activities, technology and the environment are closely interrelated. We then emphasize the need to identify the design problem, and to differentiate the situation of concern, which brings about designing the system. Often we need to involve some key concepts, such as usability factors and representations of human activities.

4.2 Second 5-Week Session

The second session focuses on the design process, developing the idea that HCI is concerned with designing computer systems to match the needs of people. It explains how to gather data about human activity in preparation for design, and discusses how to approach the design process from a user-centered perspective. It will also cover modeling techniques used to ensure usability principles, such as the task analysis methods. Furthermore, we will introduce the concept of requirements definition from the software engineering perspective, explaining the need to define, in a set of requirements, a usable functional form for the system.

4.3 Third 5-Week Session

The third session extends the second with more coverage in design support, deepening the idea that HCI design employs

a rapid, iterative design process with comprehensive evaluations. It will look at analytic and empirical methods for evaluating interactive systems prototypes to meet user requirements. Students are reminded that design support is needed to deal with the complexity of a design process. It will also cover the use of guidelines, explaining how to make use of sources of design knowledge and how to apply guidelines to answering questions of detailed design.

5 Course Activities

In each of the three sessions throughout the semester, students, on being presented with a problem or scenario, embark on the PBL cycle of learning through organized groups of 3-5 students. Initially, they will not possess enough prior information to solve the problem or to clarify the scenario. Problems are often ill-structured, but devised according to concrete, open-ended situations. To construct a solution, students must identify, locate, and use appropriate resources. They ask questions referred to as "learning issues" on the various aspects of the problem, which help them realize what knowledge they require, thus focusing their learning efforts and establishing a means for integrating the information they acquire. There are generally three relevant stages of activities: analysis, research and reporting, with discussion and feedback from peers and teacher at each stage [1, 2, 3].

5.1 Analysis. Throughout this stage, students organize their ideas and prior knowledge related to the problem, and start defining its requirements. This helps students devise a specific statement of the problem. Meanwhile, they are encouraged to pose learning issues, defining what they know and what they do not know. This helps them assign responsibilities for research, eliciting their existing knowledge as a crucial step in learning new information.

5.2 Research. Throughout this stage, students collect necessary information on specific learning issues raised by the group. They may conduct library searches, seek sources on the Internet, and/or interview knowledgeable authorities. Students teach themselves as they research their learning issues. It is intended that when they come to realize the complexity and texture of the problem, they may see that information is meant to manage problems effectively.

5.3 Reporting. At this stage, students report their findings to the group. Individual students become "experts" and teach one another. Subsequently, their discussion may generate a possible solution, or new learning issues for the group to explore. Final solutions are reported to the class as a whole, and to the teacher. The teacher's feedback addresses if the original learning issues were resolved and if the students' grasp of the basic principles, information, and relationships is sufficiently deep and accurate.

For inexperienced learners, or for retaining clear control, instructors could intersperse mini-lectures or teacher-led discussion into the group-oriented PBL work. Likewise, the problems should contain some breaks at which teachers can shift attention from the group to the instructor. At such breaks, the teacher may deliver critical information, lead a class discussion to clarify concepts or misconceptions, introduce the next stage of the problem, shift students' attention to a different skill, or compare groups' progress.

6 Course Resources

In each of the fifteen weeks throughout the semester, we have four hours (extendible to six upon students' initiative and instructor's schedule) of contact with the students, devoted to PBL activities, and when required, to tutorials on foundational understanding, technical and team support. In the PBL classroom, also one of our Software Engineering Laboratories, equipped with UNIX-based workstations and Windows-based PCs, we have collected some tools and references for students' exploration, with the help of a laboratory technician, a teaching assistant and the course instructor.

6.1 Course References

- 1) Baecker, R.M., Grudin J., et al. (Eds.). Readings in Human-Computer Interaction: Toward the Year 2000, 2nd Edition. Morgan Kaufmann (MK), 1995.
- 2) Baecker, R.M. (Ed.). Readings in Groupware and Computer-Supported Cooperative Work: Assisting Human-Human Collaboration. MK, 1993.
- 3) CHI Conference Proceedings, "Human Factors in Computing Systems," 1995, 1996, 1997, 1998, 1999.
- 4) Dix, A.J., Finlay, J.E., Abowd, G.D., and Beale, R. Human-Computer Interaction, 2nd Edition. Prentice Hall (UK), 1998.
- 5) *interactions*: a quarterly publication of ACM.
- 6) Johnson, P. Human-Computer Interaction: Psychology, Task Analysis and Software Engineering. McGraw Hill, 1992.
- 7) Newman, W.M., and Lamming, M.G. Interactive System Design. Addison Wesley, 1995.
- 8) Nielsen, J. Usability Engineering. AP Professional, 1993.
- 9) Preece, J, et al. Human-Computer Interaction. Addison Wesley, 1994.
- 10) Preece, J., and Keller, L. (Eds). Human-Computer Interaction. Prentice Hall (UK), 1990.
- 11) *SIGCHI bulletin* : a quarterly publication of the ACM
- 12) Wiklund, M.E. (Ed.). Usability in Practice: How Companies Develop User-Friendly Products. AP Professional, 1994.

6.2 Course Development Tools

We have acquired the following software at educationally discounted prices or free from the Internet for students' project work in interface design for interactive systems:

- Java Virtual Machines with suitable JDK (ver 1.1/1.2)
- C++ Builder 4 Enterprise edition
- Delphi 5 Enterprise edition
- JBuilder 3 Enterprise edition

6.3 Course Support Web-sites

Starting from the fall 1999 semester, we are experimenting with the faculty-wide WWW-enabled course support environment and the PBL support environment. These two web-sites are respectively to support the concept of learner-centered education. The former being oriented towards the constructivist model, is called **REAL** [4], implying a Rich Environment for Active Learning and the latter, oriented toward the PBL model, is called **SUPER** [5], denoting **SU**itable and **P**actical **E**ducational **R**esources.

7 Course Assessment

There are three 5-week sessions of activities throughout the semester. Students' teams each with a designated team leader, are supposed to complete a session-long assignment in each period. Each PBL group will be given a different topic within the same context area to explore. At the end of each session, a project document is due for grading, and the resulting group grade is the same for all students in the team. Each student is also graded individually by the instructor and by each of the team members. The instructor's evaluation is based on what each team member adds to the team discussion and presentation, and what the instructor perceives each member's contribution to the team to be. The peer's evaluation is based on a confidential rating sheet, to be completed by each team member at the end of each major assignment. It includes each team member's contribution for that phase with explanatory comments. The overall assignment assessment is composed of the group grade and the individual grade. The course evaluation is made up of the total grades of the three project assignments in the three sessions of the semester, plus the final examination grade, which comprises reading some research articles and writing responses within a 3-hour period.

8 Course Delivery

In each fall semester, the author has to plan from three to five sets of study materials, based on the number of groups whose size is of 3-5 students, for each of the semester's three sessions, for a total of about twenty-five students enrolled in the course. The following presents the author's preliminary selections for the first two sessions, and the arrangement of group-based project work in the last session.

8.1 First Session

In each session, individual PBL groups are assigned to read a number of articles and some tutorial chapters from reference texts. They are to conduct the analysis-research-reporting cycle of PBL studies, and culminate their results in a study report and a presentation in front of all other groups, at the end of the session. In the process, they need to have at least two in-class team meetings (plus some self-directed group meetings outside the class schedule, without the instructor's presence, in which they organize themselves to manage the assignment) to which they invite the instructor to observe and give feedback to their discussion. The topics selected for the five groups in the first session are respectively as follows:

8.1.1 Compulsory Articles

- Norman, D.A., "The Psychopathology of Everyday Things," Section 6.1, Book #1, 5-22.
- Baecker, R.M., and Buxton, W.A.S., "A Historical and Intellectual Perspective," Section 6.1, Book #1, 35-48.

8.1.2 Selective Articles

Individual PBL groups are to read a different article, besides the compulsory ones, to extend their understanding.

- Barnard, P., "The Contributions of Applied Cognitive Psychology to the Study of Human-Computer Interaction," Section 6.1 Book #1, 640-658.
- Card, S.K., and Moran, T.P., "User Technology: From Pointing to Pondering," Section 6.1, Book #1, 587-602.
- Landauer, T.K., "Let's Get Real: A Position Paper on the Role of Cognitive Psychology in the Design of Humanly Useful and Usable Systems," Section 6.1 Book #1, 659-666.
- Landauer, T.K., "Relations Between Cognitive Psychology and Computer System Design," Section 6.1, Book #10, 141-160.
- Olson, J.R., and Olson, G.M., "The Growth of Cognitive Modeling in Human-Computer Interaction Since GOMS," Section 6.1, Book #1, 603-625.

8.2 Second Session

In the second session, each PBL group is mainly responsible for investigating a specific case study of HCI design, and for proposing a topic of interactive system design for experimentation in the third session.

8.2.1 Compulsory Articles

- Gould J.D., "How to Design Usable Systems (Excerpt)," Section 6.1, Book #1, 93-121.

- Lewis, C, and Rieman, J., "Getting to Know Users and Their Tasks," Section 6.1, Book #1, 122-127.
- Mountford, S.J., "Tools and Techniques for Creative Design," Section 6.1, Book #1, 128-141.
- Boehm, B.W., "A Spiral Model of Software Development and Enhancement," Section 6.1, Book #1, 281-292.
- Grudin, J., "Interactive Systems: Bridging the Gaps Between Developers and Users," Section 6.1, Book #1, 293-303.

8.2.2 Selective Case Studies

- Jackson, S.L., Krajcik, J., and Soloway, E., "The Design of Guided Learner-Adaptable Scaffolding in Interactive Learning Environments," In Proc. of CHI 98, Apr. 18-23, Los Angeles, 187-194.
- Wallace, R., Soloway, E, Krajcik, J, et al., "ARTEMIS: Learner-Centered Design of an Information Seeking Environment for K-12 Education," In Proc. of CHI 98, Apr. 18-23, Los Angeles, 195-203.
- Rose, A., Ding, W., Marchionini, G., et al., "Building an Electronic Learning Community: From Design to Implementation," In Proc. of CHI 98, Apr. 18-23, Los Angeles, 203-210
- Tamaki, M., Kuwabara, T., et al., "Network-based Education System Designed to Allow Individual Student Progress and Improve Teacher Efficiency," In Proc. of the third Asian Pacific Computer & Human Interaction, Jul. 15-17, 1998, Kangawa, Japan.
- Yoshino, T., Munemori, J., Ito, S., and Nagasawa, Y., "Development of a Multi-User Electronic Conference System DEMPO III for Supporting New Ideas," In Proc. of the third Asian Pacific Computer & Human Interaction, Jul. 15-17, 1998, Kangawa, Japan.

8.3 Third Session

In this last session, each PBL group is assigned the project of designing an interactive application and constructing a prototype for evaluation. Prototyping tools such as those mentioned in *Section 6.2*, are introduced in the weekly tutorials. The prototyping process requires each team to designate a client, apart from the already selected team leader and other team members. Each team is then assigned a client from another team. Namely, each team, acting as developers, is to complete an interactive system design and prototype for another team. Yet, the same team is the client of another group. The instructor will act as the project supervisor for all teams. To help students work in team, it is stipulated that the client's role is to clarify the project, and to resolve ambiguities as they arise, whereas the supervisor's is to guide, motivate and provide feedback to the team. The team leader has to coordinate the team

activities, and ensure effective team communications, whereas the team members have to help set the project goals, accomplish tasks assigned, meet deadlines, attend team meetings and participate in editing the document to be turned in at the end of the project period. To support students' activities in this session, instructors, depending on the projects, will suggest suitable chapters from reference texts, and timely and informative articles from different research magazines and journals. At the end of the project, each PBL team is required to present their project, and lead a class discussion to elicit students' responses, besides submitting the project document and demonstrating the software product. In the process, the supervisor must be ready to resolve any team conflicts, which arise from lack of coordination or communications. It should be noted that team coordination could be enhanced by exercising care in teaming up the PBL groups from the very beginning.

9 Course In-Class Organization

The course is organized to encourage students to come to class on time and to read the assigned articles and start the PBL-cycle of activities. We have a flexible time-slot for progress briefing with questions, if available by each team at the beginning of each meeting. This is then followed by the instructor's update. Next comes the student-led team-based PBL discussion. Those teams, which fail to conduct regular progress briefings because of members' absence or late arrivals or under-preparation, will be tracked and have negative score recorded for lack of professional attitude. This is considered as a sign of under-performance, and is strongly discouraged. However, this arrangement requires cautious and flexible time management. It is the author's experience that the PBL discussion often overruns the scheduled class-time slots, and we have to re-arrange our classes so that we have a weekly seminar/workshop-like meeting whose duration could go from 3 to 4 hours, with some refreshment(s) breaks.

10 Lessons Learned

Admittedly, the students' responses to the PBL approach are not deprived of barriers. Some students reported that since they came from the didactic model of education, they did not have the skills to work as an independent group, which involves setting their own learning objectives and having to research them. And there are students who just prefer to be given the information (answers). They encounter the discomfort to "learn to learn". They often find the group process and the notion of independent learning particularly hard to cope with. As PBL instructors, it is found that we could become most helpful as facilitators if we could balance directions to students with assistance to them. We must be prepared to ask leading and open-ended questions, monitor progress, help students to reflect and create a positive learning atmosphere. This process requires time on the part of both students and the instructor.

Nevertheless, knowing how to work with groups as well as how to train group-members to work with one another, is not something most faculty presume expertise in. It is also learned that alignment among the course content, PBL instructional objectives, team evaluation, and student practice, is of critical concern. Such expectations should be communicated continuously so that students are not confused and can efficiently learn without guessing "what the professor wants." Positively, it has been observed that students in PBL groups use their training to enrich their performance; they articulate their newly constructed ideas and through argument and persuasion, build shared meaning. This is something not often seen in the traditional format of teacher-centered, subject-based learning, with the dominant mode being courses of lectures. Yet, it is convinced that the PBL-approach requires trust and cooperation among students and the instructor, so as to promote meaning-making and shared ideas among all participants.

11 Conclusion

In this article, the author has presented one approach to teaching human-computer interaction, incarnated as *Software Psychology*. This approach may not work in every situation or for every instructor. Yet, it is believed that the PBL method enables students to be active learners. As they determine what will be learned and how, they question, explore, explain, evaluate, and collaborate. They tailor their learning by pursuing individually the learning issues that interest them and by focusing in groups on the areas they understand least. The resulting learning should be integrated, accumulative and connected.

References

- [1] Albanese, M., and Mitchell, S., "Problem-Based Learning: A Review of Literature on Its Outcomes and Implementation Issues," *Academic Medicine*, Vol. 68, No. 1, 1993, 52-81.
- [2] Engel, J., "Not Just a Method But a Way of Learning," In D. Bould and G. Felletti (Eds.), *The Challenge of Problem-Based Learning*. New York, N.Y.: St. Martin's press, 1991, 21-31.
- [3] Ryan, G., "Student Perceptions about Self-directed Learning in a Professional Course Implementing Problem-Based Learning," *Studies in Higher Education*, Vol. 18, 1993, 53-63.
- [4] Vat, K.H., "REAL: Towards a WWW-Enabled Course Support Environment for Active Learning," Faculty Technical Report, FST/SE-1999-01, University of Macau, Macau.
- [5] Vat, K.H., "SUPER: Towards a WWW-Enabled PBL Support Environment for Software Engineering Education," Faculty Technical Report, FST/SE-1999-02, University of Macau, Macau.

