

Problem Based Learning

and Asymetrix ToolBook II Instructor



Total Multimedia Solutions - <http://www.tms.nl>

Hens Zimmerman - hens@gmx.net

Co-edited by Carmen Barth

OLE 1998, Cheltenham



Problem Based Learning (PBL)

“How can I get my students to think?” is a question asked by many faculty, regardless of their disciplines. Problem-based learning (PBL) is an instructional method that challenges students to ‘learn to learn,’ working cooperatively in groups to seek solutions to real world problems. These problems are used to engage students’ curiosity and initiate learning the subject matter. PBL prepares students to think critically and analytically, and to find and use appropriate learning resources.”

Barbara Duch <bduch@udel.edu>

Before we’ll arrive at a definition of PBL, it’s interesting to look at what clearly *isn’t* computerized PBL. Through the years, we’ve all seen programs that most certainly aren’t engaging students to learn. They usually go a little something like this:

- ◆ Introduction
- ◆ Prelude to a question
- ◆ Question
- ◆ Evaluation of student’s answer
- ◆ *Yet another prelude to yet another question*
- ◆ *Yet another question*
- ◆ *Yet another evaluation*
- ◆ ... {repeat several times}
- ◆ *ZZZZZZZ...*
- ◆ **Wake up!!**
- ◆ Some conclusive thoughts/scoring

Sometimes a student is allowed to move back and forth to re-visit pages, but in some worst case scenarios even that isn’t allowed. Instead of creating an engaging computerized multimedia-version of a book, a student has to sit through an entire program with even less navigational ‘tools’ than he knows from reading a book. Sometimes multimedia-elements are added for no other reason than to be able to call the application an ‘Interactive Video Program’, while it is nothing more than a “page-turner” [BEETSMA92] offering interrupted video...

At the exact opposite we find the ‘Too Much Freedom’ concept. Examples of this kind of approach are lots of Windows helpfiles and giving a student the entire World Wide Web to answer a question. Jacob Nielsen has addressed this problem:

“In general, a problem with hypertext is that it destroys the authority of the author to determine which sections readers need to read first [...]. Some people may think that the need to guide the reader [...] is an indication that the whole notion of non-sequential text is flawed. If one needs to make certain buttons very large and graphically attractive in order to induce readers to select them first, why bother giving any options at all? The reason is that there are several different classes of readers. Some readers may be experts in the domain of the information base and will know how to navigate it to find the information of specific interest to them. [...] Other readers, however, may be novices who are more in need of guidance.” [NIELSEN95]

“When users move around a large information space as much as they do in hypertext, there is a real risk that they may become disoriented or have trouble finding the information they need.” [NIELSEN95]

“Even in [a] small document, which could be read in one hour, users experienced the Lost In Hyperspace phenomenon as exemplified by the following user comment: ‘I soon realized that if I did not read something when I stumbled across it, then I would not be able to find it later.’ Of the respondents, 56% agreed fully or partly with the statement, ‘When reading the report, I was often confused about where I was’.” [NIELSEN95]

While trying to give our students more freedom, they tend to get overwhelmed and we are not helping them one bit with all this fancy technology.

“The most important function of the 8 buttons at the bottom of the screen is to give the user the feeling that he is completely in control of the program. It is possible to go from one screen to the next on a quiet guided tour through the material. But it is also possible to start on your own journey and do research in the database or move to another lesson or listen to opinions. You should never get lost, you should feel safe and you should not realize that you are working with an interface.” [NOLTHUIS91]

Filling in the blank...

What would help both students and teachers is a marriage between traditional learning (student gets assignment and disappears into university library for several weeks to find material for his paper) and modern CBT-technology. The part of the teacher that, given enough time, would like to assist every student individually can be simulated by a computer program to a certain level. Students are working on their assignments, aided by multimedia-technology and a program (‘Agent’) that can be both a passive observer and an active personal guide. Course results (and some highly interesting statistics) are relayed to the teacher afterwards. If nothing else, it seems at least appropriate to find out what we can gain by combining all this new technology with a traditional problem solving ‘tool’ called Problem Based Learning:

“Student-centered; faculty-facilitated

Problem-based learning is a pedagogical strategy for posing significant, contextualized, real world situations, and providing resources, guidance, and instruction to learners as they develop content knowledge and problem-solving skills (Mayo, Donnelly, Nash, & Schwartz, 1993). In problem based learning, students collaborate to study the issues of a problem as they strive to create viable solutions. [...] Because the amount of direct instruction is reduced in problem based learning, students assume greater responsibility for their own learning (Bridges & Hallinger, 1991). [...] The instructor's role is to encourage student participation, provide appropriate information to keep students on track, avoid negative feedback, and assume the role of fellow learner (Aspy et al., 1993).

Evolution of Problem Based Learning

Although the roots of problem based learning can be traced back through inquiry training, John Dewey, and apprenticeships, recent evolution of the pedagogy was pioneered at Case Western Reserve University in the early 1950s. The structure developed by this university now serves as the basis of the curriculum at many secondary, post-secondary, and graduate schools including Harvard Medical School (Savery, 1994). In fact, over 80% of medical schools use the problem based learning methodology to teach students about clinical cases, either real or hypothetical (Vernon & Blake, 1993, Bridges & Hallinger, 1991).

Going Beyond Content

The ability to solve problems is more than just accumulating knowledge and rules; it is the development of flexible, cognitive strategies that help analyze unanticipated, ill-structured situations to produce meaningful solutions. Even though many of today's complex issues are within the realm of student understanding, the skills needed to tackle these problems are often missing from instruction. Typical problem solving taught in schools often tends to be situation specific with well-defined problem parameters that lead to predetermined outcomes with one correct answer. In these situations, it is often the procedures required to solve the problem that are the focus of instruction. Unfortunately, students skilled in this method are not adequately prepared when they encounter problems in which they need to transfer their learning to new domains, a skill required to function effectively in society (Reich, 1993).

Real-life problems seldom parallel well-structured problems; hence, the ability to solve traditional school-based problems does little to increase relevant, critical thinking skills students need to interact with life beyond classroom walls. Well-structured problems with their sterile environments in which there is only one right answer simply teach students about problem solving, not how to problem solve. In real life, we seldom repeat exactly the same steps to solve problems; therefore, the lockstep solution sequence taught in well-structured classroom problems is seldom transferable. Instead, real-life problems present an ever-changing variety of goals, contexts, contents, obstacles, and unknowns which influence how each problem should be approached. To be successful in their chosen career, students need practice solving ill-structured problems that reflect life beyond the classroom. This skill is the goal of problem based learning.” [SANDIEGO96]



Problem-Based Learning put to use at the University of Groningen (RuG), Department of Animal Physiology.

In traditional PBL, a problem is allowed to be "ill-structured" to resemble the nature of problems as they occur in the real world. In order to make some means of scoring or feedback possible (which is at the heart of most CBT programs), we need at least *some* interface-element which enables student-interaction with the program. A multiple choice widget seems to be unsuitable for PBL, which after all is supposed to engage and motivate a student and challenge him to design his own learning strategy. A multiple choice widget, which is by definition a perfectly structured problem, just won't cut it. A fill-in-the-blank widget is much more in line with PBL and can be a perfect starting point for a student's journey into information. At the University of Groningen (RuG), Dept. of Animal Physiology, we are also experimenting with drag-and-drop exercises. The staggering amount of possible answers with even the simplest of drag-and-drop exercises makes this an interesting PBL assignment, though it isn't ill-defined.

Problems provide clues, context, and motivation; they are the maps which guide learners to useful facts and concepts. This challenges CBT-designers to add a certain amount of *forgiveness* to an assignment. Where the answer to a binary multiple choice (yes/no) is either wrong or right, a fill-in-the-blank interface is able to give stimulating feedback to the student as an intermediate step and suggest a certain action if it turns out a student has not visited the right pages yet. This is different from direct scoring and giving away the right answer.

An example of fill-in-the-blank scoring requires a student to input a *summary*. To label a student's answer as adequate, he must have entered certain terms in his summary. ToolBook's string-functions can be used to count hits, reduce words, remove junk-words, junk-characters, doubles, etc. By phonetic discriminating (*SoundEx*-like), removing double-printed characters and ignoring the sequence in which characters make up a word, ToolBook can even be forgiving to dyslectic students (which, whether you like it or not, are 5 to 10 percent of the users of your program) [See: DUMONT91] At the University of Groningen, Dept. of Animal Physiology, we are experimenting with several different versions of TB 'to get' handler structures that reduce student-words *and* good answers to simplified (more forgiving) versions before matching. This is just one small step ahead in a society that is dyslexia-aware.

A 'good' problem cannot be understood or resolved as first encountered; it is to challenge thinking and organize learning. This is much in line with what's required from a biology-student anyway. PBL is traditionally used in medical sciences, so it's not remarkable that this particular problem solving method fits biology equally well. It should be clear that PBL is not the one and only ultimate method of teaching for any learning situation; other disciplines will benefit more from the use of traditional linear learning methods, multiple choices, strict trial and error scoring or role-playing. For instance, someone who must learn to operate a machine is supposed to master certain skills. We don't really need to engage him or get him motivated. CBT still sounds like a cool idea to create a cost-effective teaching method, but a *simulation* with trial-and-

error scoring is more in line with this kind of learning than PBL could ever be. This does not imply you shouldn't be using simulations as part of your PBL-program.

The teacher models, coaches, provides a scaffold for the apprentice investigators, and then fades into the background to become an observer and tutor of metacognition. But how much of the human teacher is still needed for this time-consuming task? We'll talk about this in more detail in the paragraph labeled "Artificial Intelligence and Agents".



Layered Design

Whoever gives computerized PBL some thought eventually arrives at a point where different *layers* of information can be distinguished:

- Information space
- Problem space

The *problem space* is where we invite a student to solve a problem. We could also call this the *assignment layer*. This can still be quite a linear part of a program. Most of the times a student will get some introduction and one or more problems to work on. Maybe we want to restrict him from moving to problem 2 when he hasn't finished problem 1 adequately. This depends on the kind of program we are designing, whether it is a (scored) test or a self-learning course. Even in some self-learning (non-scored) programs it may be undesirable if a student skips working on an unfinished problem and starts working on the next one (which assumes he has mastered the knowledge that could be gained from solving the earlier problem).

The information space can consist of one or more databases (TBK-files). A student should feel free to move from the assignment layer to the information layer(s) and back. We may give him the benefit of the doubt and initially open the entire information space with all its navigational tools. If our program detects a student is spending too much time without getting near a solution to the current assignment, it can decide to close off parts of the information space. It is as if there's a teacher inside the program who says: "Oh dear, this is getting you nowhere. Let's try something different here..."

One way to achieve dynamic manipulation of the information space is the use of a table which lists pages and actions to be taken. In pseudo-code:

Name of page	Action to be taken
menu 1	Close link to submenu
page x	Disable navigation except for the back-button
...	...

If a student arrives at a page called 'menu 1' in this example, the link to a submenu is closed. When he visits 'page x' (another very bad name for a page...), the program temporarily disables navigation, sans the back-button. Of course we could also *enable* or *add* objects that are initially disabled or absent, but there is a certain logic to a default-state of "Everything is possible" and dynamically *closing* doors when needed. This also makes the creation of an 'Encyclopedia'-version of our information space very simple, because it is nothing more than the *absence* of a modification-table. Such an 'encyclopedia' can be useful for teachers who need to design an assignment.

Manipulation-tables can be swapped during the course of a program to increase or decrease the idea of freedom. If a student takes too much time solving a problem, we can swap the “*Everything is allowed, as if you were using MS Encarta*”-table for the “*Let’s restrict your freedom a bit*” table.

There are several ways to create manipulation-tables. One particular powerful option is a table that contains actual OpenScript, compiled and executed during runtime. This can turn out to be a bit tricky to debug at times, but who said humans should write the code that goes inside the table?



Artificial Intelligence and Agents

“Knowledge about the user can be both obtained explicitly (by questioning) and inferred (by noticing)” [LAUREL93]

Our layered environment is very well capable of noticing what students are doing. In other words: it doesn't need to ask them.

“Responsiveness [...] requires that the agent has access to a dynamic model of the user, or at the very least, a log of his experience in a particular application or environment with rules for interpreting that experience when formulating actions.” [LAUREL93]

In ToolBook, basically every place a student visits is a page with a name (if you didn't give it a name yourself, it still has a unique name that distinguishes it from other pages). We can log where a student has been and how many times he's been there. Because we know the entire information-space (as opposed to letting our student surf the ever changing WWW), we can also see where a student *didn't* go yet. This information is simply computed by subtracting the visited pages (logfile) from a table that lists all available pages, i.e. the complete information space. For a given assignment, a teacher can mark certain pages as 'recommended visits'. This enables our program to compare a subset of the entire information space to the logfile of a student. If a student comes up with an adequate solution to an assignment without visiting recommended pages, he is either brilliant (already well-versed in the current subject) or maybe cheating. If, however, a student fails to answer adequately, the program can decide to close off certain parts of the information space and guide the student to the recommended pages.

Our 'Agent' may be a simple evaluation routine that opens and closes hyperlinks inside the information space, based on variables such as the following:

- How much time has this student spent on this problem already? (implemented by a low resolution Windows timer that is problem specific or a passive mechanism that uses for instance the Windows API function `GetTickCount`)
- How many times did this student give an incorrect or partly correct answer? Is he going in the right direction? (implemented for instance by a *fill in the blank(s)* scoring mechanism that counts the use of unique words expected to be found in an adequately answered question)
- Which pages did this student visit? Which ones did he skip? Were the pages he *did* visit appropriate for the current assignment or is he clearly.... lost in hyperspace? (implemented by a counting routine and a log-file)

Because our program is counting visits to pages anyway, it is in theory capable of combining logfile-data and make its own decisions. Every CBT-developer knows there are moments of doubt, where one road looks just as good as a different one. If statistics can tell which roads appear to get better student-results, our ToolBook-program is completely capable of changing itself:

```
script of book library = newScript;  
  
save changes to book library;
```

“Individual living organisms can generally both interact with and modify their environment, as well as adapt to small changes in that environment. This makes an organism flexible, and stable in the face of change. [...] Artificial organisms should also be capable of interacting with their environment and adapting to it in a limited way. Of course, one should not expect an artificial organism to be capable of adapting to any arbitrary change in its environment. It can adapt to some changes, while others might completely throw it off balance, just as in the case of biological organisms.” [LUDWIG93]

Note how useful logfiles can be in the long run, when we have collected the logs of tens or maybe hundreds of students. Both teachers and CBT-developers can easily see which pages don't get visited and which ones do. If for example 200 students *never* visited a certain page, this is a clear sign that the current interface does not *invite* students to go to that page. If the information on such a page is vital to solving a problem, the page could be promoted to a more prominent place somewhere down the hierarchy. Maybe we'll discover there were too little hyperlinks to this page, or maybe its name should be included in the result-sets of more student queries. The latter could be achieved by adding more/better keywords to this page. Or maybe we are looking at a page which is redundant to solving an assignment. It may still be vital for future assignments, so dropping it from the information space would seem a little hasty. We want to *re-use* our information space as much as possible. It's easy to close a page during one assignment (if it gives away the answer immediately) and still use it for a different assignment. This 'closing' and 'opening' can be performed dynamically, so a page may be hidden during assignment 1, while showing during assignment 2.



Problem based conclusions

Creation of a multimedia-library is time-consuming and needs lots of testing. Logfiles however provide feedback to teachers/courseware-designers so improvements can be made while already using the program. It is not undesirable to let students work with an incomplete program. This can even turn out to be cost-effective, because bad design can be spotted before it snowballs into a lot of wasted time and effort.

Most of this technology is still rather new and it isn't clear what works and what doesn't. It could be desirable to create a program that self-adapts and changes its strategies permanently if some idea turns out to be a false road. This of course turns our students into guinea pigs...

The many different angles to Problem Based Learning require a full team, consisting of domain experts, instructional designers, programmers, artists, testers, etc. A team like this will usually come with a certain price-tag. It wouldn't hurt to join forces and use the same technology in different fields of science. Effort can be divided between institutions and one teacher's assignment can be shared by others via the WWW. Unfortunately (and this is truly a big pain in the rear, if not *the* biggest torpedo for cooperation) not everybody uses Asymetrix ToolBook II Instructor... yet.

Because of the nature of PBL, one tends to think in terms of web-deployment. A hybrid Web/CD-ROM (or DVD in the nearby future) program certainly seems to be the ultimate PBL design-challenge:

- Students can access a teacher's program (assignment-layer) via the web.
- Different (relatively small) assignments can be made available via the web, modified and uploaded by a teacher using an ftp client.
- A fixed library of information and multimedia-files is delivered on CD-ROM, CD-R or CD-RW media, available for students and teachers.
- A web-application (assignment) accesses libraries and multimedia on the local CD-ROM.
- Student-results are sent to a teacher (by ftp or e-mail).
- A CD-ROM library can even be accessed stand-alone (There's that Encyclopedia again...) by a different program that's stored on the CD-ROM as well.



First tests with Asymetrix ToolBook II Neuron are promising to say the least. Preliminary experiments using the "one large library/several different assignments" concept are received with great enthusiasm as well.



Hens Zimmerman
Oosterstraat 19 A 55
9711 NN Groningen
THE NETHERLANDS

home-phone: +31-(0)50-3118456
work-phone: +31-(0)50-3632332
hens@gmx.net ICQ 4435768

Total Multimedia Solutions
Neuhuyskade 94
2596 XM Den Haag
THE NETHERLANDS

Tel.: +31-(0)70-3144630
Fax.: +31-(0)70-3144638
Web: <http://www.tms.nl>

Bibliography

BEETSMA92 – “Computers als Onderwijs Ondersteuner”, Drs. Yta Beetsma (COWOG/ICCE), handout for CBT-workshops at faculty of medical sciences, University of Groningen.

CLIPPER92 – CA-Clipper Reference Guide Vol.2, Manual © 1992. Computer Associates International, Inc.

DUMONT91 – “Dyslexie – Theorie, diagnostiek, behandeling”, Prof.Dr.J.J.Dumont. Lemniscaat Rotterdam, ISBN 90-6069-773-1

HOGAN97 – “Exploring the world of artificial intelligence: Mind Matters”, James P. Hogan. DelRey, The Ballantine Publishing Group New York, ISBN 0-345-41240-0

NIELSEN95 - “Multimedia and Hypertext - the Internet and beyond” , Jakob Nielsen. AP Professional (a division of Academic Press), ISBN 0-12-518408-5

NOLTHUIS91 – “Multimedia and the art of linking reality”, a paper by Joseph Nolthuis, Educa Video/Utrecht School of Arts.

LAUREL93 - “The Art of Human Computer Interface Design”, Brenda Laurel, editor. Addison-Wesley Publishing Company, Inc., ISBN 0-201-51797-3

LUDWIG93 – “Computer Viruses, Artificial Life and Evolution”, Mark A. Ludwig. American Eagle Publications, Inc., ISBN 0-929408-07-1

SANDIEGO96 - San Diego State Universities website
<http://edweb.sdsu.edu/clrit/learningtree/PBL/WhatisPBL.html>

WOODS94 – “Problem-based Learning: how to gain the most from PBL”, Donald R. Woods. D.R. Woods, Waterdown ON Canada, ISBN 0-9698725-0-X



Weblinks

Bibliography - Problem Based Learning

<http://ultibase.rmit.edu.au/Resources/bibliography/problem.html>

Center for Problem-Based Learning Core

<http://www.imsa.edu/team/cpbl/cpbl.html>

Characteristics of Problem-Based Learning

<http://www.realtime.net/~peregrin/pblcharacteristics.html>

Email privacy problem; a real PBL-example on the www

<http://www.uchsc.edu/chanc1lr/offedu/pbl/pbl.html>

Links to Problem Solving

<http://www.oise.utoronto.ca/~ggay/pblmlink.htm>

McMaster University - Problem Based Learning

<http://socserv2.socsci.mcmaster.ca/soc/beehive/pbl.htm>

Model Page for HiED Web/PBL

<http://www.abo.fi/instut/hied/pbl.htm>

PBL - a webquest

<http://edweb.sdsu.edu/clrit/PBLReport12.html>

PBL Case Writing Workbook (gopher text)

<gopher://gopher.medlib.iupui.edu:70/00c%3A/iuinfo/pblast/archive/pblcasew.asc>

PBL description

<http://www.district94.dupage.k12.il.us/science/carper/pbl/description.html>

PBL in the context of large classes (includes book!)

<http://chemeng.mcmaster.ca/pbl/pbl.htm>

PBLIST Home Page; PBL in health care

<http://ddsdx.uthscsa.edu/pblast/pblasthome.html>

Problem Based Learning at Fulmore Middle School

<http://www.realtime.net/~peregrin/pblhome.html>

Problem-Based Learning Directory

http://www.biology.iupui.edu/Biology_HTML_Docs/biocourses/K345/PBL_Web_Pages/PBL_Directory.html

Problem Based Learning sites

<http://www.uchsc.edu/chanc1lr/offedu/pbl.html>

Problem Solving - The Library-in-the-Sky

http://www.nwrel.org/sky/Classroom/Science/Problem_Solving/Problem_Solving.html

Problem Solving Skills

http://www.biology.iupui.edu/Biology_HTML_Docs/biocourses/K345/PBL_Web_Pages/ProblemSolvingProcess.html

Teachers & Technology - Problem Based Learning

<http://www.csuchico.edu/~andersb/webtrain/pbl.html>

The OSU Problem Based Learning Home Page

<http://www.med.ohio-state.edu/PBL/>

Total Multimedia Solutions

<http://www.tms.nl>

UD PBL: PBL or related links

<http://www.udel.edu/pbl/others.html>

University of Delaware - Problem Based Learning

<http://www.physics.udel.edu/wwwusers/pbl/>

What is PBL?

<http://edweb.sdsu.edu/clrit/learningtree/PBL/WhatisPBL.html>



Sorry... what was that again?

Agent

A computer process that acts as guide, as coach and as amanuensis.
[LAUREL93]

AI

Artificial Intelligence

API

The Application Programming Interface is a set of functions needed to write applications for the Windows Operating System.

CBT

Computer Based Training

MISPWOSO

Maximegalon Institute of Slowly and Painfully Working Out the Surprisingly Obvious (Douglas Adams... who else?)

Neuron

An Asymetrix webbrowser plug-in that enables students and teachers to access ToolBook II applications via the internet/intranet.

OpenScript

ToolBook's programming language

PBL

Problem Based Learning

RuG

Rijks Universiteit Groningen (University of Groningen)

SoundEx

A character function that indexes and searches for sound-alike or phonetic matches. It is used in applications where the precise spelling of words is not known or where there is a high probability of misspelled names. [CLIPPER92]

TB

ToolBook, the number 1 authoring environment.

TBK

A ToolBook-file

TMS

Total Multimedia Solutions

WWW

World Wide Web



Appendix - Steps in Problem-Based Learning

from Problem-Based Learning at Fulmore Middle School
<http://www.realtime.net/~peregrin/pblhome.html>

- What do you know about this situation?
- Have you ever seen anything like this before
- What's going on?
- What do we need to know?
- Are you sure of the "facts" you know?
- What else do we need to know?
- Where can you find the information you need?
- When can you have the information for us?
- How will you bring the information to us?
- How are we doing?
- What's working? What's not working?
- Describe the resolution of the problem.
- Test the resolution.
- Debrief and synthesize.

