Agile Problem Driven Teaching in Engineering, Science and Technology

Pradip Peter Dey[1], Thomas M. Gatton[1], Mohammad N. Amin[1],
Mudasser F. Wyne[1], Gordon W. Romney[1], Alireza Farahani[1], Arun Datta[2],
Hassan Badkoobehi[1], Ralph Belcher[1], Ogun Tigli[1] and Albert P. Cruz[1]
[1]National University, 3678 Aero Court, San Diego, CA 92123, U.S.A.
[2] National University Community Research Institute, 11255 North Torrey Pines, La Jolla, CA 92037

ABSTRACT:

In problem driven teaching, all major teaching activities are driven by a problem or a set of problems. Some typical problem solutions are demonstrated by the instructor. Problems could be based on realistic or abstract situations. Recent research suggests that abstract problems may have some advantages over others. This paper demonstrates how course learning outcomes are adequately handled in agile problem driven teaching in Engineering, Science and Technology courses for effective interactions.

Problem driven teaching is not the same as Problem Based Learning (PBL). In PBL, learners are usually organized into groups, and one or more problems are given to each group for solving the problems under the supervision of the instructors. Although PBL is highly successful in certain environments, it is not necessarily appropriate for all learners and all topics since the teaching methods may not be dynamically modified. Adjusting teaching methods based on learner feedback may be appropriate in multi-model, multi-strategy learning environments. Agility in teaching helps to overcome the different challenges faced by different learners for different topics. Our discussion of agile problem driven teaching considers all instructional strategies including lectures, transformations, experimentations, problem solving, analogical, case-based and mathematical reasoning for affective learning utilizing tools and technologies in an innovative way. An effective teacher takes complete responsibility for student learning and grade specification. Agile problem driven teaching may be the right approach for solving some of the most crucial problems in engineering education.

1. INTRODUCTION

According to the 2005 report of the National Academies, "Rising Above The Gathering Storm: Energizing and Employing America for a Brighter Economic Future," the USA is losing its long-standing global lead in science, math and engineering education (The National Academies, 2005). "Recent test results show that U.S. 10th-graders ranked just 17th in science among peers from 30 nations, while in math they placed in the bottom five" (Wallis, 2008). There is enough indication that this educational trend is temporally coupled with a closely following severe economic downturn. Clearly, prolongation of this trend is a danger to the U.S. economy and the U.S. standard of living. We live in a rapidly changing world, with a global job market, global educational competition, a globally integrated economy, escalating energy problems, mounting trade imbalances and an unprecedented financial crisis. The new U.S. generation needs to have modern educational advantage in order to compete in the global job market and solve exigent

problems. Well educated engineers, technologists and scientists are in demand. In order to build a robust economy with sustainable growth, educated problem solvers are needed. However, schools and colleges are not succeeding in producing innovative problem solvers. Many different teaching strategies have been tried and show important improvements in student learning in various settings (Borman, 2005). However, significant nationwide improvements have not been achieved despite these isolated demonstrations of successful cases.

Problem Driven Teaching (PDT) is closely related to Problem-Based Learning (PBL) which is the educational process by which problem solving activities and instructor's guidance facilitate learning. PBL is a pathway by which students "learn how to learn". It challenges students to think critically, be pro-active, analyze problems, and find and use appropriate learning resources (Queens 2009). In PDT, all major teaching activities are driven by a problem or a set of problems. Some typical problem solutions are demonstrated by the instructor and students practice problem solving among other activities in order to master the learning outcomes. Problems could be realistic or abstract. Recent research suggests that abstract problems have some advantages over others (Kaminski, Sloutsky, Heckler, 2008). However, unlike PDL, an important aspect of PDT is that instructor plays an active role in teaching activities.

In a PBL environment, students are usually divided into teams to work on problems. In this model the problems are expected to drive the learning activities, a common practice in medical science, particularly in psychiatry (Adamowski, Frydecka, & Kiejna, 2007). In this strategy the instruction of the topic is organized around problem solving tasks. Some of these tasks are problem analysis followed by relevant information gathering; students may then try to identifying some possible solutions and provide pros and cons of each proposed solution. This strategy gives students abundant opportunities to think critically, communicate with their team members, present arguments as well as counter arguments, transfer knowledge to new situations, and develop creative problem solving skills. A key element of this approach is that all the problems are designed in such a way that students must gain new knowledge before they can solve the problem. Often, problems are complex and may not even be well defined. The new knowledge is acquired as students try to work through the problem. The role of the instructors in this environment is mainly as the facilitator and the guiding mentor. Students are given more responsibility for their own learning and are engaged in active or discovery learning in the sense that students discover and work with content that they determined to be necessary to solve the problem. It is believed that by working through the problems, students are better able to internalize the problem and understand the underlying concepts and fundamental relationships needed to solve the problem. In a traditional setting, instructors tend to simply provide students with facts, this is known as teacher-oriented instruction and many students may not be receptive to or grasp the new concepts. As a result they either tune out or never develop a deep understating of the materials and simply imitate mechanical operations and the essential analysis and arguments necessary to solve the problem. One common criticism of the PBL method is that students may not recognize what might be important for them to learn, so the facilitator must be extra careful to assess each student's prior knowledge. Another criticism is that instructors cannot cover as much material as the traditional method. Furthermore, the method is hard to implement and there are different definition and interpretation of the strategy and the manner of its implementation, within instruction, appears to be ad hoc. PBL requires a lot of planning and extensive work by the instructor.

In this paper we propose a new variation to PBL where the role of the instructor is more than being just a facilitator. We introduce the model of Agile Problem Driven Teaching (APDT) in which the instructor's role more closely approaches the traditional instructor approach. In APDT the instructor's presentation is more dynamic and can easily digress to cover a variety of "relevant" topics according to inquiries received from students. Similar to its PBL counterpart, the class is given a complex problem to solve. However, unlike PBL the role of the instructor in this approach is elevated to periodic coaching in order to accelerate the PBL strategy. In particular the instructor plays a key role contextualizing the problem and actively participating in the research, analysis, and information gathering component of PBL. The instructor becomes a facilitator once the initial body of knowledge to solve the problem is gathered. This gives students an opportunity to put together the final solution. In the APDT method the open problem along with student inquiries and the collective information gathering process derives the lecture, and this is where the agility in instruction becomes apparent and critical. The instructor must have extensive knowledge of the subject to efficiently process information and resolve student's questions and further suggest new directions for information gathering. In this method the new knowledge is shared among all teams in a form of a short presentation by the instructor. This provides the instructor with an opportunity to further clarify student misunderstandings and misinterpretation associated with the problem and the newly acquired information.

Use of technology can further simplify the APDT implementation and enhance student experience. In particular, in one scenario, students and the instructor each will have Tablet PCs with collaborative, interactive teaching tool such as Dyknow vision in a networked environment. In this, often wireless, networked environment, the Instructor's display is broadcasts to student's machines allowing students to synchronously follow instructor's lectures and add notes to the lecture with ink pen. The instructor can also permit students to lead class from their own machine. This connectivity clearly facilitates a powerful medium for students to collaborate and share as they search for new knowledge to solve the problem. Among useful features of Dyknow is the panel submission. Panel submission is most useful for APDT, because each team can, anonymously, submit their findings and request comments from the instructor. The instructor can quickly scan through all student submissions and select one or more panels to share with the rest of the class. It is through this sharing that the instructor can clarify misunderstandings, make additional comments or presentations, or provide new directions for search. The instructor can also opt to privately give comments to student submissions. Many educators have reported increase in student participation when Tablet PCs are used in the class. Panel submission and the follow up discussions often clarify many student misunderstands; the instructor can utilize the panel submissions to point out common errors and misinterpretations, or focus on the new knowledge reported by students to point out its relevance to the problem.

Adjusting teaching methods based on learner feedbacks may be appropriate in multi-cultural learning environments. Agility in teaching/learning and grading helps to overcome the different challenges faced by different learners for different topics. According to Glickman *"Effective teaching is not a set of generic practices, but instead is a set of context-driven decisions about teaching. Effective teachers do not use the same set of practices for every lesson . . . Instead, what effective teachers do is constantly reflect about their work, observe whether students are learning or not, and, then adjust their practice accordingly* (Glickman 1991). Our ability to teach is enhanced with our agility to adjust. In education, the same size does not fit all. Agility is the

basis of APDT and consequently, it may combine a variety of teaching strategies. In addition to PBL, many other teaching and learning methods can be employed including the following: lecture (Cashing 1990); class discussion, brain storming (Instructional Methods, 2009); technology-based teaching learning (Kearsley & Shneiderman, 1999; Trondsen, 1998); game-based learning (Prensky, 2004; Van, 2008); experience based learning (Experience Based Learning Systems, 2008); inquiry-based learning (Eick & Reed, 2002; Educational Broadcasting Corporation, 2008); community-based learning (Owens & Wang, 2008); brain based learning (Brain Based Learning, 2008); work-based learning (Bailey 2003; Cunningham, Dawes & Bennett, 2004); project-based learning (Helic, Maurer, & Scerbakov, 2004; George Lucas Educational Foundation, 2008); team-based learning (Michaelsen, et al., 2008); web-based Learning (Lee & Baylor 2006; O'Neil & Perez 2006); and participatory learning (Barab, Hay, Barnett, & Squire, 2001). There is no conflict between these methods and APDT since it can adopt these methods if needed.

2. PROBLEM DRIVEN TEACHING IN MATH FOUNDATIONS (Main Author: P. Dey)

At National University, the APDT method was applied in a graduate course on mathematical foundations of computer science (National University course number CSC 610). Cohen (1997) was used as the textbook for this course. Lectures, class discussions, brain storming and quizzes were used regularly following the textbook. However, one main problem drove all these activities. In the first meeting for this course, each student was given a variant of the following problem.

Class-Problem: PART_1. Build the most powerful computing machine that you can think of. Your machine should be able to process complex languages such as L_{11} = {$a^n dc^n dy^n$: where n > 0 }. Demonstrate that a string like 'aadccdyy' would be accepted by the machine. You need to build the machine by defining its elements mathematically. You are not required to deliver the machine with hardware components. If you do not use standard notations provided in the textbook or discussed in the class then you need to explain your notations.

It is known that Finite State Machines or Finite Automata can accept regular expressions. So, any string from the regular expression b*aba*b would be accepted by the following Non-deterministic Finite Automaton (Figure 1).

However, Non-deterministic Finite Automata cannot process a language like L_{11}, mentioned above, which requires a more powerful machine. You should be able to build such a machine. Explain how your machine will accept strings from $L_{11.}$

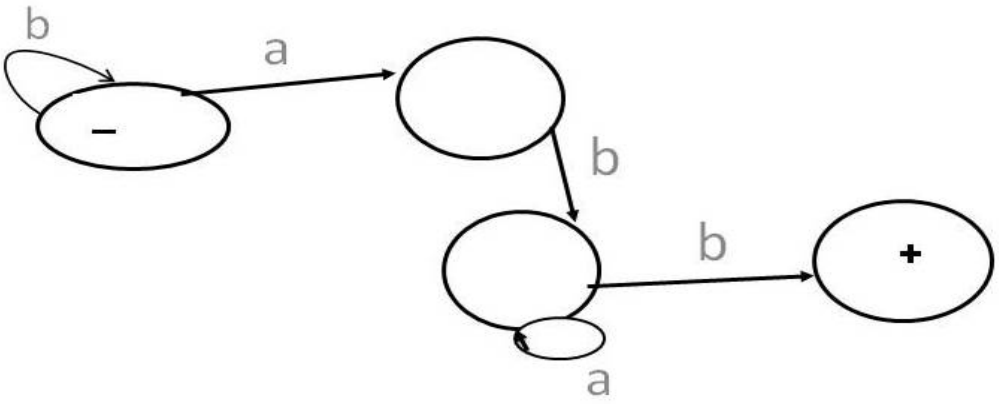

Figure 1. A Non-deterministic Finite Automaton for b*aba*b

PART_2: A Turning Machine has a finite set of states with one START state and some (may be none) HALT states. We always mark the START state with 1 and the HALT state with 2 (when there is only one HALT state).

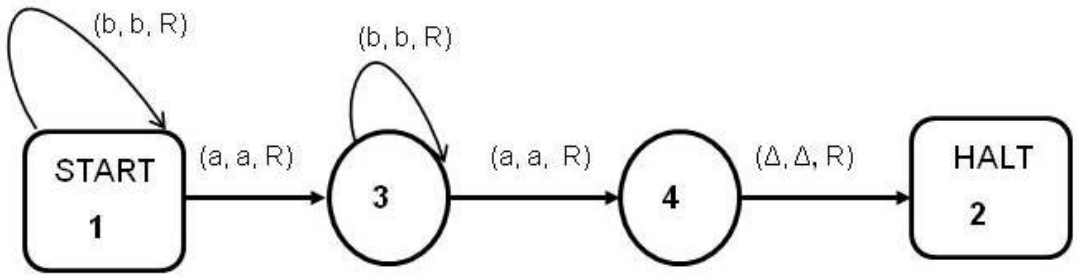

Figure 2. A Turing Machine for { b*ab*a }

The Turing Machine given above accepts the language: { b*ab*a }. When an input is given it is processed by the transitions as the Turing Machine goes from state to state starting from the start state. The input is accepted by reaching the HALT state. If the HALT state and its incoming transition are destroyed then no input is accepted. The Turing Machine of Figure 2 would look like the machine given below after the destruction of the HALT state and its incoming transition.

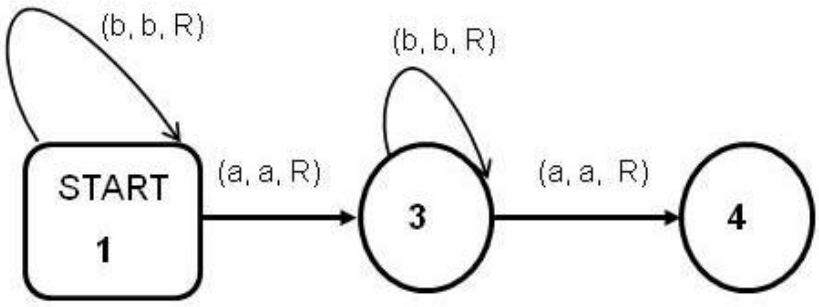

Figure 3. The Turing Machine of Figure 2 after the HALT state and its incoming transition are destroyed.

If the input is: aba, then the machine will start in the START state and reach state 4 after going through the loop of state 3. However, the input is not accepted.

You are asked to complete the following three tasks:

1) In the first step, destroy the HALT state(s) and their incoming transitions of your machine of the assigned problem (of PART_1) and examine the consequences.
2) In the second step, destroy the START state and the associated transitions of your machine (in addition to the destructions mentioned in step 1) and examine the consequences.
3) In the third step, reconstruct the machine so that it is distinct from the original machine of PART_1 (may have one or more additional states and/or transitions) and still process the same language (end of the Class-Problem).

The above problem was given out in the first meeting and the students started working on the problem and all teaching activities were related to this problem. Finite Automata, Pushdown Automata, Linear Bounded Automata, Turing Machines, Post Machines and the related sets accepted by these machines are taught with the goal of solving the above problem. The class started with the Finite Automata which could process regular expressions but would not be adequate for Context-Free languages. Pushdown Automata were then introduced in order to process Context-Free languages as well as regular expressions. The machines were built for dealing with increasingly complex sets. Finally, the problem was solved by building a Turing Machine for $L_{11.}$ The students then experimented with breaking some parts of the machine and examining the consequences and then performing the repair work on the machine following the Part II of the problem. The teaching of the entire course was driven by a set of problems presented including the main problem mentioned above. Effectiveness of the teaching method was not measured scientifically although students evaluated the course according to the usual National University evaluation method which indicated significant improvements over earlier offerings of the same course.

The course learning outcomes for this course are given as follows. Upon successful completion of this course, students will be able to:

* Construct a model of computation for a given specification.
* Construct a Turing Machine for a given computational problem.
* Develop a program implementing the model of computation.
* Prove that regular expressions are equivalent to Finite State Machines.
* Prove that complement of a regular language is regular.
* Prove that a given language is Context-Free.
* Construct a Push Down Automaton for a given computational problem.

The first two learning outcomes are directly achieved through the class problem described above. Other learning outcomes are discussed when attempts are made to build machines of different types in the process of finding increasingly more powerful machines.

3. PROBLEM DRIVEN TEACHING IN DATABASES (Main Author: M. Wyne)

Various approaches to the teaching of computer science courses are currently under use and consideration. I (M. Wyne) have attempted a few teaching strategies that seem to make a difference in student understanding of the subject matter. To start with, I have revised my handouts to include not only the important materials for in-class discussion, but also to include study questions and practice problems in the handouts. The questions in the handout serve to challenge their thinking processes when it is time for them to study. They also help to generate more in-class discussion. Once in a while, I will discuss structure-activity relationships by asking students for their input instead of providing all of the facts for them. If a wrong answer is suggested, I usually follow up with another question to allow the students to see why it is not a good answer. I fully realize that by involving the students in lectures in this way, I may not be able to cover as much material as I would like, but I have never felt that I was not in control of my lectures. I have found that not only do the students learn database design better, but they have also come to enjoy the subject more, and have often come in to discuss their own experiences in database design with me on an informal basis. Furthermore, at the end of each major topic, I usually discuss one or two practice problems from previous exams, again with active participation from the students as a way to show them how to solve the problems. Additionally in my courses, I do encourage my students to come and discuss their solutions with me. In doing so, I am able to see their approach to learning and also which concepts are difficult for them to grasp. This allows me to think of different approaches to convey these concepts.

I have taught required introductory database design course to computer science students. I was interested in incorporating some of the PDT techniques. The PDT approach I use is a new concept in teaching of introductory Database courses. In this approach a problem is introduced and emphasized to students in order to make the study of database design more relevant to what student may experience in their practical life after graduation. For the past few years, what I came up with was a bit of “How to design a database for a company”. The requirements of the company database are also provided as the Company is organized into DEPARTMENTs. Each department has a name, number and an employee who manages the department. We keep track of the start date of the department manager. Each department controls a number of PROJECTs. Each project has a name, number and is located at a single location. For each EMPLOYEE’s we

store, Social security number, address, salary, sex, and birth-date, number of dependents. For each dependent, we keep track of their name, sex, birth-date, and relationship to employee. The number of hours per week that an employee currently works on each project is recorded.. The direct supervisor of each employee is also important. We also know that each employee works for one department but may work on several projects. These initial set of requirements gives students a good start and allow them to guess and work out other details.

The design of the problems that is used in PDT plays an important part in achieving the intended learning objectives set by the faculty teaching a course. To a large extent, the learning objectives set for the course determine their learning activities and form the framework for the direction of teaching methodology. I use the PDT approach to address four challenges: inadequate time to cover important material using only a traditional lecture format; enhance student motivation to study the material on a daily basis rather than just the night before an examination; increase student comprehension of the material, as well as increase student awareness of the level of comprehension required for satisfactory performance on examinations; and the necessity for the students to see the "whole picture" of the design before it is covered in class, as a mechanism for teaching students how to use logical thinking in learning the concepts presented in the course. During the first class with the students I present a problem, design a database for a company. Each lecture starts by introducing and reviewing the main concepts and steps required to develop an efficient database design. Design strategies commonly used are presented and the effects of any of the design features on their overall database design are discussed. The students are asked to recognize the effects of a specific design decision on the overall system operation. This is further emphasized by practical situations that are presented in the lecture. Consequently, they are expected to apply this information to a question on the examination. In this approach, I will ask several probing questions which are designed to stimulate the students' understanding regarding design decisions, and possible side effects.

4. AGILE PROBLEM DRIVEN TEACHING IN INFORMATION TECHNOLOGY (IT) (Main Author: G. Romney)

4.1 APDT in IT Focuses on Real-world Problems

Information Technology (IT) emphasizes synergistic solutions between technology, people and processes to successfully resolve enterprise computer problems. In the IT Management (ITM) program at National University, students learn that people, namely the client, drive the development process. IT professionals, with their knowledge, skills and set of technology tools attempt to meet the requirements specified by the client. In almost every development instance, the client’s perception of the desired product evolves. The initial functional specification frequently defines a problem that may not have a solution as specified. Consequently, the engineering IT development team may have to redefine the problem in order to deliver a workable solution. In order to meet deadlines, frequently, the problem may be subdivided into modular components assigned to separate development teams. This is where APDT better prepares IT professionals to handle such common challenges encountered in the workplace.

Similarly to PBL, APDT as used in ITM focuses on real-world problems. Additionally, "agility" features are introduced to more closely simulate the real-world workplace that students will encounter. Agile components introduced are a) including multi-faceted problems that are subdivided into multiple team interaction and coordination, b) adjusting the defined problem to team-member skills, and c) allowing team-members to discover alternate solutions and "work-arounds" when barriers are encountered while discovering the solution to a problem. Redefinition of the problem in order to achieve a solution is frequently required in ITM. Introducing students to this realistic occurrence by employing the APDT method in their instruction better prepares them for the workplace.

Where PBL is based on a defined problem with usually one solution, APDT is based on the premise that agility, and creativity are required to redefine the problem in order to achieve a successful solution.

4.2 Information Security Technology Course Problem – APDT Examples

A major course deliverable for ITM 475, a senior-level course in Information Security Technology, is an assigned problem that requires multiple team participation. This course teaches Information Assurance (IA) and the domains of the Certified Information Security System Professional (CISSP 2009). IA deals with protecting and defending information and information systems by "ensuring confidentiality, integrity, authentication, availability and non-repudiation" (Information Assurance2009). The *Problem* objective given to the students, charged with assuming the role of IT security professionals, was to architect, design and implement components of a basic secure intranet.

The PDT approach implemented *experiential* hands-on assignments, presentations and projects that progressively contributed to the final solution. Consider the *Problem* to be a picture comprised of components or elements which are *mini-problems.* The *Problem*, then, is the sum and seamless integration of all the *mini-problems*.

The *mini-problems* identified by the students were the following:

1. Specification of the Local Area Network (LAN).
2. Specification of the hardware and software resources available for usage.
3. Demonstration of Wikis for class communication.
4. Demonstration of Blogs for class communication.
5. Evaluation of open source versus industry software for web portals and the selection of Microsoft's Sharepoint for the WebPortal software.
6. Implementation of a prototype Sharepoint webportal that has, among its many features, Wiki and Blog capability.
7. Implementation of Microsoft IIS for Active Directory and Certification Authority functions needed for secure authentication as specified in the *Problem* objective.
8. Implementation of a VMware virtual environment.
9. Implementation of a Microsoft SQL Server.
10. Implementation of a MySQL database.

11. Implementation of a Ruby on Rails web development environment in order to meet the needs of the overall SOET Intranet by integrating the various *mini-problems*.
12. Implementation of a needed Linux test bed.
13. Implementation of a faculty research hard-disk repository.
14. Implementation of portable multi-factor authentication technology.

The *Problem* was divided into three distinct projects with three coordinated development teams using surplus servers and network switches on a gigabit communications pathway to the Internet as follows: 1) Architect the system to function virtualized under VMware, and integrate a virtual Rails environment, 2) Employ Microsoft (MS) SharePoint WSS services as the central node and interconnected via SQL script with both MS SQL Server and MySQL databases, and 3) Provide a certificate authority and implement a portable two-factor authentication process; and provide a Linux and implement a Time-date server for digital signatures. Part of the IA challenge of this defined problem was to architect the authentication process that would provide secure access for specific roles, such as administration, staff, faculty, adjunct faculty and students.

Each team created an initial Memorandum of Understanding (MOU), with the instructor acting as the client, regarding the scope, team organization, deliverables and delivery time-frame for each project. The delivery of the four projects and integrated *Problem* was scheduled four weeks from initiation. As work progressed, the discovery of seemingly insurmountable obstacles that threatened the success of the overall project heightened team frustration levels. The successful delivery of the intended solution of the *Problem* was in jeopardy. The concern by all was not only to deliver Project 1, 2 or 3, but the successful inter-operation of all three comprising the *Problem.*

Specific instances of where agile project management had to be employed, and, hence, where APDT was utilized were the following: 1) Team 3 had to reduce the scope of its project, 2) Team 2 encountered seemingly insurmountable obstacles, 3) Team 1 had to adjust to include the members of Team 3, and 4) all three teams had to adjust to on-going limitations and failure of the hardware.

4.3 Team 3 Reduced the Scope of Its Project

Half-way through Project 3, a key team member was sent out of state by his employer. The Time-date stamp server feature assigned to this student could not be given to another. The Team re-negotiated with the client (the Instructor) and determined that this feature was not on the critical path for successful completion of the *Problem.* Hence, by agreement, and modification of its MOU this feature was deleted, the team combined with Team 1 and successfully completed the remainder of its tasks.

4.4 Team 2 Encountered Seemingly Insurmountable Obstacles

The Team successfully installed .Net Framework 3.0 and SharePoint WSS 3.0 but emphasized that SharePoint could not execute SQL scripts even though it was based upon the MS SQL database. Likewise, there was no way to interface MS SQL or MySQL to SharePoint. The Team resigned itself to not be able to deliver the terms of the MOU and insisted that the client's

objective was not realizable with the specified product, MS WSS SharePoint. One member of the team, however, kept searching and discovered that SharePoint, by default, does not contain any database connections (SharePoint 2009). However, with another MS product, Office SharePoint Designer 2007, it was possible to access both MS SQL and MySQL databases, which the team successfully downloaded and demonstrated.

4.5 Team 1 Adjusted to Include Members of Team 3

Refer to 4.3, above, for the circumstances that required a team management adjustment. This left two teams to successfully complete the *Problem* on schedule.

4.6 All Teams Adjusted to Hardware Limitations and Failure

As frequently is the case in university (and industry) settings, computer hardware availability is limited and IT professionals implement whatever might be supplied. Servers were configured as functional components were identified but this did limit ambitions of the teams. All of this, again, required agile adjustment of plans.

The final result after four weeks, however, was delivery of an operational basic intranet with five servers and seven VMware virtual servers providing the required services. This could only have been achieved through APDT that taught the students how to accomplish what appeared to be "impossible" and to become better prepared for the reality of the real-world workplace.

5. PROBLEM DRIVEN TEACHING IN PROGRAMMING LANGUAGES (Main Author: O. Tigli)

As educators, our role is to facilitate students' learning. Of course, this is easier said than done. Unfortunately, learning doesn't happen overnight. We need to find creative ways for our students to increase their long-term of retention of knowledge. A well-known Chinese proverb states that: "Tell me and I forget; show me and I may remember; involve me and I will understand". The key here is the involvement of the students so that they will get the big picture, the fundamental idea that we are trying to convey as teachers.

In that sense, both "Problem Based Learning" and PDT bring the key involvement factor by introducing a problem to the students to solve.
However, these two approaches differ in their natural progress. While the first one promotes the direct student involvement throughout the process, the second one not only requires students' engagement with the issue in their hand but also promotes the educators to be a guiding figures, and major key players to monitor their students' progress. Definitely, both approaches have a certain place in education, and probably as educators, we utilize both techniques in our way of teaching.

In his inspirational 1988 paper, Stephen Wolfram mentions the fact that the essence of education is interaction, and this does not mean canned programs and canned classes. He emphasizes that the interaction will have to come from people since AI based computer programs are still far

away to communicate with people in a meaningful way. The human touch is still has the extreme importance in education.

In my programming languages classes, I (O. Tigli) try to take advantage of face-to-face communication by guiding students throughout the learning process, by covering fundamental topics and learning outcomes in each and every session. In these sessions, we go through little excursions in computer science, and try to grasp the concepts of lexical analysis, parsing, semantic analysis, and code generation by using initially small but growing examples. I also give them a short-term project to work on. Sometimes an alternative project requiring advanced programming skills is given as an option, as well. Besides all of these, I also try to maintain a simple supplemental web site to provide my students useful online materials including class notes and pointers to the other sites related with our topic. My observation is that this approach is working nicely and increasing the retention level for the students and creating good understanding of the topic on hand.

6. PERSPECTIVES ON AGILE PROBLEM DRIVEN TEACHING

Within the broad APDT framework multiple perspectives are encouraged. However, considerable emphasis is placed on agility in teaching strategies. Adaptation of a PDT approach to suit the learning levels and styles of the students is one of the essential concepts behind APDT. It must consider both the individual student as well as the combined characteristics of each team. Finding a successful teaching path that can benefit the full range of student competency levels is one of the most challenging obstacles in teaching. It is important to bring weak groups and/or individuals up to a suitable level of competency and allow enough flexibility for the advanced groups and/or individuals to continue in the development of more advanced competencies. This can be addressed by paying attention to needs of an individual student in a variety of ways including individual assignments or examination questions that offer “bonus” points. Typically, only advanced students will tackle this type of problem.

A combination of team and individual assignments can vary to promote the development of individual writing and problem solving skills. Simpler team assignments that insure group competencies in basic skills can efficiently develop these individual skills through active problem solving while also allowing the development of basic team skills. The level of problems assigned for team-based problem solving can be perceived by different groups of students as ranging from easy to difficult. This reflects the ranging level of skills possessed by the individual students. Individuals in each group may also reflect differing levels of knowledge. Should all the low level students be put together, or should each team have a distribution of students with basic, intermediate or advanced skills? Usually, students have some knowledge about each other’s skill levels and personalities and pick each other based on their personal preferences. Often, there is a tendency for the exceptional students to isolate themselves from the struggling students, thereby denying struggling students the benefit of their knowledge.

One of the most widely used and important Program Learning Outcomes (PLO’s) is that “The student will demonstrate a capability to work productively in a team environment.” Another

standard PLO is that "The student can demonstrate college level written and oral communication skills." By integrating team solution presentation immediately after a problem-based session, students practice and develop these critical skills. In an extension of PDT, students are required to present their solutions to the rest of the class, after a problem solving session. This serves in the development of their presentation skills as well as their logical thought processes in problem solving. Some teams simply distribute the problems among the members and do little to develop team skills or benefit from problem-based learning. By requiring that the team's solution presentation occur immediately after the problem-solving, and only assigning one problem during a problem solving session, teams are forced to work together on a single problem. Various levels of difficulty can be incorporated by giving each team a set of increasingly difficult problems and letting them pick which problem to solve. While this may allow a lazy, but advanced skill level, group to avoid solving a challenging problem, it is more likely that a team will desire to assert their superior problem solving skills to the class through their presentation.

Within a broad range of interpretations of APDT certain clear guidelines are evolving. The practitioners of APDT realize that certain propositions are more important than others:

1. Problems that match course learning outcomes are better than famous problems that poorly match the learning outcomes.
2. Analyzing feedbacks from students is more important than declaring student centered environments.
3. Dynamically adjusting teaching strategies to learner's goals and preferences is more important than following a teaching plan.
4. Teaching activities are driven by realistic problem solving, feedbacks from students and learning outcomes rather than by a strict schedule or mandate.
5. Demonstrating problem solving strategies on sample problems is more effective for teaching/learning than lecturing on them.
6. Dynamically combining multiple strategies in multiple models is more effective for teaching/learning than relying on a single pre-planned strategy.

One of the authors (A. Datta) will be testing the APDT approach for teaching cyberinfrastructure (CI) to the biology students under CIBRED (CI-TEAM Implementation for Biological Researchers, Educators, and Developers), an NSF funded project. Teaching engineering concept to biologist and other non-engineering students is always challenging (Datta, et al., 2009). Nevertheless, the impact of CI in education is such that the Office of Cyberinfrastructure (OCI) at the National Science Foundation has established a rich source of educational and research materials through TeraGrid (TeraGrid, 2007) to meet the 21st Century's demand for scientific talent. Materials are freely available through CI/TeraGrid (Kay, et. al., 2008). Additionally, OCI put forth the CI-TEAM (Cyberinfrastructure Training, Education, Advancement, and Mentoring for Our 21st Century Workforce) program to aid education initiatives directed toward this new workforce. CIBRED is one of the funded collaborative projects (awarded to O. Crasta of VBI with A. Datta as a collaborator) of this OCI initiative. Courses are now being developed in a collaborative way using multidisciplinary approach integrating scientific and technology information from a variety of disciplines The focus is to teach students from diverse disciplines for learning some essential concepts on computer technology in the context of application of cyberinfrastructure. These courses developed for K13 & K14 levels will be offered in an innovative classroom setting for hands-on experimental learning with a focus on solving a

scientific problem as a team. However, such an approach will be effective if the instructor follows the agile teaching technique to facilitate forming the students group working on a specific scientific problem.

7. CONCLUDING REMARKS: SETTING THE STAGE FOR EXPERIMENTAL STUDIES

Agility in teaching is generally advocated in several ways by many authors (Glickman, 1991; Dey et al., 2007) although it remains a challenge in practice for many teachers. Emerging tools and techniques would make it easier to deal with this challenge. Teachers need to perform their teaching with sufficient agility in order to adjust their strategies to problem solving environments and learner's goals, styles and preferences. The unique combination of PDT and agile teaching makes it resilient so that it thrives in a wide variety of environments. In order to measure the success of APDT it is necessary to conduct experimental studies in controlled environments. With deeper understanding of the issues, we are now better prepared for conducting our experimental studies on the effectiveness of our teaching methodology. Initial indications are that students as well as teachers benefit from APDT.

References

Adamowski, T., Frydecka, D. & Kiejna, A. (2007). "Problem based learning (PBL) – Possible adaptation in psychiatry". Psychiatria Polska, 41 (2), 163-169.

Bailey, T. R. (2003). *Working knowledge: Work-based learning and education reform*. New York: RoutledgeFalmer.

Barab, S. A., Hay, K. E., Barnett, M., & Squire, K. (2001). Constructing virtual worlds: Tracing the historical development of learner practices, *Cognition and Instruction,19*(1),47–94

Barell, John (2006). Problem-Based Learning: An Inquiry Approach, Corwin Press, 2nd Edition.

Brain Based Learning (2008) http://eduscapes.com/tap/topic70.htm

Borman, K. M. (2005**).** *Meaningful urban education reform : confronting the learning crisis in mathematics and science.* Buffalo: State University of New York Press

Cashing, W. E. (1990). Student ratings of teaching: Recommendations for use (IDEA Paper No. 22). Manhattan: Kansas State University, Center for Faculty Evaluation and Development.

CISSP (2009). http://www.isc2.org/, 15 February 2009 23:15 UTC.

Cohen, D. (1997) Introduction to Computer Theory, 2nd Edition, John Wiley & Sons, Inc., 1997.

Common Teaching Methods (1992)
http://honolulu.hawaii.edu/intranet/committees/FacDevCom/guidebk/teachtip/comteach.htm

Cunningham, I., Dawes, G., & Bennett, B. (2004). *The handbook of work-based learning.* Adershot: Gower Publishing.

Dey, P. P., Amin, M., Bright, M., Datta, A., Akhtar, S., Cruz, A., Wyne, M. F., Kennedy, P., Olson, P., Snyder, R., Farahani, A., Afuwape, S., & Badkoobehi, H. (2008)**.** Multi-model multi-strategy teaching/learning in science, engineering and technology**.** *Proceedings of the International Computer Science and Technology Conference* (ICSTC-2008)**,** San Diego**.**

Datta, A., Caesar, J., Schuman, J., Cammer, S., Daphne Rainey, D., Crasta, O. (2009) CIBRED: Engineering Education on Cyberinfrastructure with a multidisciplinary approach for Non-Engineering Students, ASEE/PSW, March 19-20, San Diego

Duch, B. (2008). Problem-based learning. Retrieved February 10, 2008 from http://www.udel.edu/pbl/.

Educational Broadcasting Corporation.. (2008). Inquiry-based learning, Retrieved February 10, 2008 from https://www/thirteen.org/edonline/concept2class/inquiry/inded.html

Eick, C. J., & Reed, C. J. (2002). What makes an inquiry oriented science teacher? The influence of learning histories on student teacher role identity and practice. *Science Teacher Education*, *86*, 401–416.

Experience Based Learning Systems (2008) Experience Based Learning, Retrieved December 4, 2009 from http://www.learningfromexperience.com/

Farahani, A. & Uhlig, R. (2009) Use of Tablet PCs to Generate Class Discussion and Facilitate Deeper Understanding, Submitted for publication.

George Lucas Educational Foundation, The. (2008). Project based learning, Retrieved February 10, 2008 from http://www/edutopia.org/projectbasedlearning

Glickman, C. (1991) Pretending not to know what we know, *Educational Leadership*, 48(8), 4-10, 1991.

Helic, D., Maurer, H., & Scerbakov, N. (2004). Knowledge transfer processes in a modern WBT system. *Journal of Network and Computer Applications*, *27*(3), 163–190.

Information Assurance (2009). http://en.wikipedia.org/wiki/Information_assurance, 10 February 2009 05:06 UTC.

Instructional methods: Advantages and Disadvantages http://www.adprima.com/teachmeth.htm

Kaminski, J. A., Sloutsky, V. M., & Heckler, A. F. (2008) Learning Theory: The Advantage of Abstract Examples in Learning Math, Science, Vol. 320, 25 April 2008.

Kay Hunt, S. & Lathrop, S. (2008) Campus Champions – bringing HPC to your campus (2008). Panel discussion, Proceedings of the SC08, the International conference for high performance computing (HPC), networking, storage and analysis, Nov 15-21, Austin (TX)

Kearsley, G. & Shneiderman, B. (1999) A framework for technology-based teaching and learning, Retrieved Dec. 11, 2008, http://home.sprynet.com/~gkearsley/engage.htm

O'Neil, H. F., & Perez, R. S. (2006). *Web-based Learning: Theory, research, and practice*. Hillsdale, NJ: Erlbaum.

Owens, T. R., & Wang, C. (2008). Community-based learning, Retrieved February 10, 2008 from http://www.nwrel.org/scpd/sirs/10/t1008.html

Prensky, M. (2004). *Digital game-based learning*. New York: McGraw-Hill.

Queens (2009). Good Practice, http://www.queensu.ca/ctl/goodpractice/problem/index.html, 11 February 2009 15:45 UTC.

SharePoint (2009). http://office.microsoft.com/en- us/sharepointdesigner/HA101009081033.aspx, 14 February 2009 09:26 UTC.

The National Academies (2005) *Rising Above The Gathering Storm: Energizing and Employing America for a Brighter Economic Future*, Retrieved on February 11, 2009 from: http://commerce.senate.gov/pdf/augustine-031506.pdf

Trondsen, E. (1998). Technology-based learning, Retrieved February 10, 2008 from http://www.sric-bi.com/LoD/summaries/LoDRpt01.shtml

Van, R. E. (2008). Game-based learning. *EDUCAUSE Review, 41*(2), 16–30.

Wallis, C. (2008, February 25). How to make great teachers? *Time, 171* (8).

Wilson, J. (2007) Problem-Base Learning Revolution: Reformation or an empty promise? A review of the Constructivist Approach to Problem-Based Learning in Undergraduate Medical Education - Middlesex University Occasional Papers in Education & lifelong Learning, Vol. 1, No. 1, 2007, pp44-57.

Young, L., K. Thearling, S. Skiena, A. Robison, S. Omohundro, B. Mel, and S. Wolfram, Academic Computing 2 (1988) 8-12, 62-65.