

2009

A Framework for Problem-based Learning of Systems Development and Integration

Sandeep Purao
Penn State University

Vijay Vaishnavi
Georgia State University

Richard Welke
Georgia State University

Lisa Lenze
Penn State University

Follow this and additional works at: <http://aisel.aisnet.org/amcis2009>

Recommended Citation

Purao, Sandeep; Vaishnavi, Vijay; Welke, Richard; and Lenze, Lisa, "A Framework for Problem-based Learning of Systems Development and Integration" (2009). *AMCIS 2009 Proceedings*. 772.
<http://aisel.aisnet.org/amcis2009/772>

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 2009 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

A Framework for Problem-based Learning of Systems Development and Integration

Sandeep Purao

Penn State University
sandeep-purao@psu.edu

Richard Welke

Georgia State University
rwelke@ceprin.org

Vijay Vaishnavi

Georgia State University
vvaishna@gsu.edu

Lisa Lenze

Penn State University
llenze@ist.psu.edu

ABSTRACT

The focus of learning and teaching about systems development and integration is expanding from monolithic systems to systems of systems. Skills and competencies needed for the next generation of systems development and integration professionals further stretch requisite knowledge (technical, design) and skills (teamwork, organizational). Accommodating this expanded coverage requires multiple epistemologies, making *problem-based learning* a potentially effective approach. In this paper we develop a framework that can be used for adapting problem-based learning (PBL) to information systems development and integration, based on a review of research from several contributing themes and a retrospective analysis of our own classroom practices, we illustrate the why and how of problem-based learning and the use of our framework with templates derived from several systems development and integration courses across two very different university settings.

Keywords

Systems of Systems, Systems Development, Systems Integration, Problem-based Learning, Framework, Templates

INTRODUCTION

Systems development is changing from the design and development of stand-alone software applications, to interventions in organizations that require integration of existing systems -- referred to as "systems of systems" (Fisher 2003). A recent report argues that traditional software engineering principles may be inadequate in these settings, suggesting that "although we can conceive these solutions, we are poor at building them" (Brownsword et al. 2006). A consequence is that the skills required for the next-generation of systems development and integration professionals further stretch the already wide spectrum of skills and competencies that are considered necessary. They include not only knowledge about technology but also existing and emerging standards; not only the design skills related to modeling of software applications but also cross-functional business processes; and not only the ability to deal with multiple users but also the ability to manage potential conflicts among stakeholders likely to be in control of different applications in need of integration. This wide spectrum of knowledge, skills and abilities demands multiple epistemologies for learning (Nilsen and Purao 2005).

One important set of techniques that brings these epistemologies together is called problem-based learning (PBL) (Barrows 1986). Although definitions vary, PBL recognizes that problems precede answers (Adams et al. 1988). It suggests an inversion of traditional pedagogy: students attempt to solve problems before learning concepts; in effect, problems act as key motivators, and concepts are learned as-needed. Proponents of PBL argue that it can lead to deeper learning and greater retention (Springer and Borthick 2007). Because of its potential to combine epistemologies, PBL has been described (Hughes et al. 1997) as a nexus of several theories of learning (e.g. Bruner 1966, Vygotsky 1978, Piaget 1929). As a result, prescriptions to implement PBL vary significantly. Barrows (1986) calls the term PBL a "genus for which there are many species and subspecies." Many acknowledge that PBL does not represent a binary decision; instead, it presents a spectrum of possibilities. The research question addressed in the paper is: *Which key dimensions define the spectrum of possibilities for designing problem-based learning in the domain of systems development and integration pedagogy?* This paper develops dimensions that can define this spectrum. The inquiry, therefore, builds on prior work in theories of learning and PBL as well as research related to systems development and integration.

PRIOR WORK

Systems Development and Integration

Although practice and technology continue to change, prior research has contributed some fundamental insights into the nature of information systems development (Mathiassen and Purao 2002). For example, Vitalari and Dickson (1983)

suggested that design behaviors such as analogical reasoning, goal-setting and strategy formulation were keys to success. Later studies (e.g. Vitalari 1985; White and Leifer 1986; Tan 1994) revealed a wider list including: business knowledge, communication skills, technical expertise, analytical skills and organizational skills. Others (Guindon et al. 1987; Curtis et al., 1988; Waltz et al., 1993) found that past experiences and context-sensitive learning were also important contributors. More recently, Iivari et al. (2004) argued for a body of knowledge (extending Freeman (1987) and Jones and Walsham (1992)) that should include five areas: technology, application domain, systems development process, organizational, and IS application.

Over the last several years, this already wide spectrum of skills has been challenged due to the move from engineering of stand-alone applications to those requiring integration of systems that support cross-functional processes (Smith et al., 2002). These efforts, referred to as systems of systems, integrate system functionalities (Lee et al., 2003) spread across multiple units to align processes and combine data. The next-generation of systems development and integration professionals are likely to engage in these efforts and, therefore, require new skills. The need is further emphasized by Dhar and Sundarrajan (2007), who, drawing on conversations with deans of business schools, argue that deep interaction between technology and organizational elements is essential in business schools.

In spite of such acknowledgments, translations to actionable plans continue to be problematic. Such efforts require, among other things, adjusting pedagogical practices to account for the multiplicity of epistemologies and increasing demands on students. The next section reviews theories of learning to help understand the choices.

Theories of Learning

Traditional pedagogy may be described as stimulus-response (Skinner 1968), often rendered with lectures (Leidner and Jarvenpaa 1995). Beyond this, research in pedagogical practices may be distilled in two broad groups that subscribe to competing assumptions. The first, called constructivist theories (Piaget 1929), assumes pedagogy should allow students to form their own abstract concepts, rather than their being fed these externally. It interprets learning as a process in which students construct new ideas or concepts, while engaged in new experiences, based on their current and past knowledge (Bruner 1966). The second, called social/cultural theories (e.g. Lave 1988), does not see the goal of learning as the formation of abstract concepts. Instead, it argues that knowledge cannot be divorced from the historical and cultural background of students. Situated learning theory (Lave 1988) argues that learning occurs when students engage in a ‘community-of-practice’ via legitimate, peripheral participation (Lave and Wenger 1990). The knowledge constructed is thus local, specific to a context, i.e. situated, and appropriate for the individual. PBL suggests the possibility of integrating the seemingly incommensurable positions of these two camps, allowing for multiple epistemologies (Nilsen and Purao 2005).

Problem-based Learning

Problem-based learning (PBL) is the simple idea that problems precede answers (Adams et al. 1988). Instead of providing concepts that students are expected to memorize, they explore a problem to understand what they need to know, and can question instructors to learn (Savery and Duffy 1995). PBL involves open student conversation, where students unlearn and learn via problem-solving. Fundamentally, PBL is “a conception of knowledge, understanding and education profoundly different from the more usual conception underlying subject-based learning” (Margetson 1998). It is the difference between a ‘knowing what,’ subject-based conception of expertise (‘covered’ as propositional knowledge), and a ‘knowing how’ conception of expertise, i.e., the ability to “make judgments about what is problematic in a situation ... and to know how to go about solving” (Margetson 1998). It presupposes propositional knowledge but does not equate expertise with it arguing for a greater integration of the two (Margetson 1998). Since its first use in the schools of Medicine, PBL has been applied to many disciplines (Barrows, 1996) such as accounting, computer science and others (with similarities such as the case method (Garvin 2007)). PBL provides a potential mapping to elements of *both* theoretical groups outlined earlier. Problems can be designed to allow exploration, leading to the formation of abstract concepts for students. They can also be designed to be contextual, allowing a more direct map to learning-how. Figure 1, adapted from Hughes (1987) highlights *key* theoretical influences on PBL.

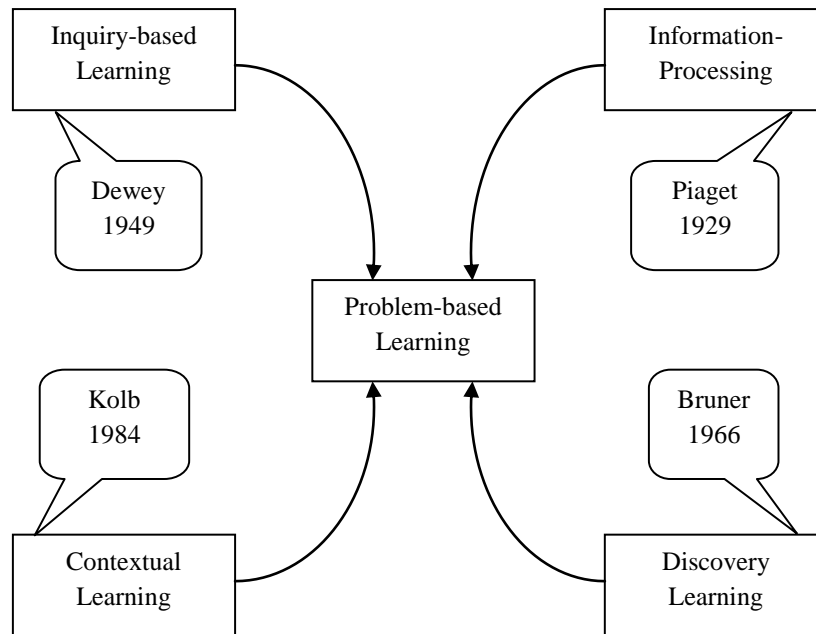


Figure 1. Selected Theoretical Influences on PBL

A logical consequence of the multiple theoretical influences on PBL is that adoptions of PBL are not binary decisions. Instead, they represent a range of approaches from traditional to those requiring imaginative uses of pedagogy and choices drawing on general descriptions (e.g. Margetson 1988, Barrows 1996). It is this space that is the focus of this paper, with a particular emphasis on the domain of systems development and integration.

RESEARCH APPROACH

The research follows an analytical and reflective approach that is difficult to succinctly define because it contains both, an integrative review of prior research combined with reflections on the authors' own practices. Following Webster and Watson (2002), we describe the integrative review as one that highlights key theoretical positions in research related to systems development and problem-based learning. Following Mathiassen (1998) and Schön (1983), we describe reflection-in-action as similar to that in the context of systems development practice (e.g. Naur 1985). In the same domain, Stolterman (1992) points to local rationality, where a practitioner may move from an open, unstructured situation to a structured perception of the situation. Our research approach resembles such reflection, informed by the integrative review of prior work. The non-adherence to a specific research method is analogous to the systems developers' lack of following the technical rationality ideal. The move to a "thinking mode" from the "performing mode," then, resembles a similar move described in other contexts (Schön 1983). The combination of integrative review and reflection allows us to discover dimensions for adopting PBL in the context of systems development and integration.

Research Setting

The impetus for these moves from 'performing' to 'thinking' is the outcome of efforts to implement PBL in different courses. These courses are offered in different settings, to different audiences, at different universities. The first is a set of courses aimed primarily at systems integration, offered at a college of information in a large traditional research university, located in a rural setting. The second set of courses covers a larger spectrum, including a database course and a process modeling course, offered at a business school in a large research university, located in an urban setting. Not only do they target different student populations and include different topics; they also accommodate different levels of maturity in the underlying body of knowledge and instructor preparation. The discussions carried out over the last three semesters among the larger set of investigators in this research effort provided several opportunities for reflection against actual implementations by the authors. These were juxtaposed against the integrative review of research to produce the framework described in the next section.

A FRAMEWORK FOR PBL IN SYSTEMS DEVELOPMENT AND INTEGRATION

The framework leverages other, although not as comprehensive, efforts reported elsewhere in areas such as Medicine and K-12 education.

Dimensions from other efforts to structure PBL

The *first* effort to structure PBL (Barrows 1986) identified two critical dimensions. One, which we label Control, captures whether the learning episodes are teacher-directed, teacher-student shared or student-directed (this continues to be a theme (e.g. Baptiste 2003)). The other deals with the Nature of the problem, i.e. a complete case vignette, partial problem simulation or free inquiry. The *second* effort (Margetson 1988) surfaces the distinction between problems to facilitate learning of propositional knowledge (subject-based, know-what) versus problem-solving (know-how) with the latter described as pure PBL. The *third* effort (Duch 2001) describes prototypical models instead of dimensions. The first “medical school model” is described as a technique used in medical schools that facilitates learning basic science concepts in the context of clinical cases, where students are assigned in small groups to a faculty member who leads the discussion as students work through a problem. The second “floating facilitator model” employs a portion of class time to group discussion as the instructor floats between groups asking questions and probing for understanding with activities such as each group reporting to the whole class, small lectures and whole class discussions. The third “peer tutor model” uses peer or near-peer tutors to extend the instructor’s ability to check the functioning of groups and probing for deeper understanding. A combination of peer tutors as floating facilitators allows working with large number of groups. The *fourth* effort (Uden and Beaumont 2006) extends the work from Barrows (1986) to present multiple models meant to clarify the nature of PBL. Their first model, learner-based cases, describes the teacher presenting students information in lectures followed by cases to demonstrate relevance. It does not require student inquiry beyond attempting to apply the concepts. The second model, case-based lectures, involves presentation of cases to the students before the lecture. The students analyze the cases using prior knowledge before new knowledge is provided via the lecture causing student-oriented structuring of information. The third model, case method, involves giving the students a complete case for study and research in preparation for subsequent class discussion. The discussion combines student-teacher-shared learning requiring hypothesis generation and data analysis with more active structuring of information. The final model, pure PBL, presents the students with an authentic problem. Students then try to solve the problem. The teacher acts as a coach to activate students’ prior knowledge. Students reflect on information used, evaluate their prior reasoning and knowledge; multiple rounds follow until the problem is solved.

The Proposed Framework: PBL-SoS

We propose PBL-SoS (pronounced *Peeble-Sauce*) that describes dimensions for structuring PBL efforts for learning and teaching Systems of Systems (SoS), a term that includes systems development and integration. The framework is primarily aimed at undergraduate (and some graduate) education. It targets key questions that an educator is likely to face as s/he considers the use of PBL for a course offering. This scope contrasts with others that have focused on the design of class sessions as well as institutional changes with PBL. A final distinguishing characteristic of this framework is that it is clearly focused on the domain of systems development and systems integration. It does not claim to be a generic solution for all disciplines nor for all topics within IS, e.g., purely managerial or organizational concerns (although we do not preclude potential applications or extensions to these areas).

The key issue that motivates the framework is of the need for multiple epistemologies to address the range of knowledge and skills required for systems development and integration. These include synthetic rules about how information technology *should* be designed and implemented; as well as discussions related to individual and group work practices, teamwork and organizational skills. Although some may be open to classic PBL such as hypothesis generation, evidence gathering, and reasoning; others represent accumulated wisdom codified in synthetic rules that must be learned because they cannot be inferred via hypothesis generation and testing. We argue that integrating these under PBL requires a framework that goes beyond classical versions. Figure 2 outlines key dimensions that define PBL-SoS. While some can be traced to prior work, or considered evident, others are the outcome of the research approach described earlier. Although the dimensions are conceptually different, we acknowledge they overlap in terms of pragmatic actions.

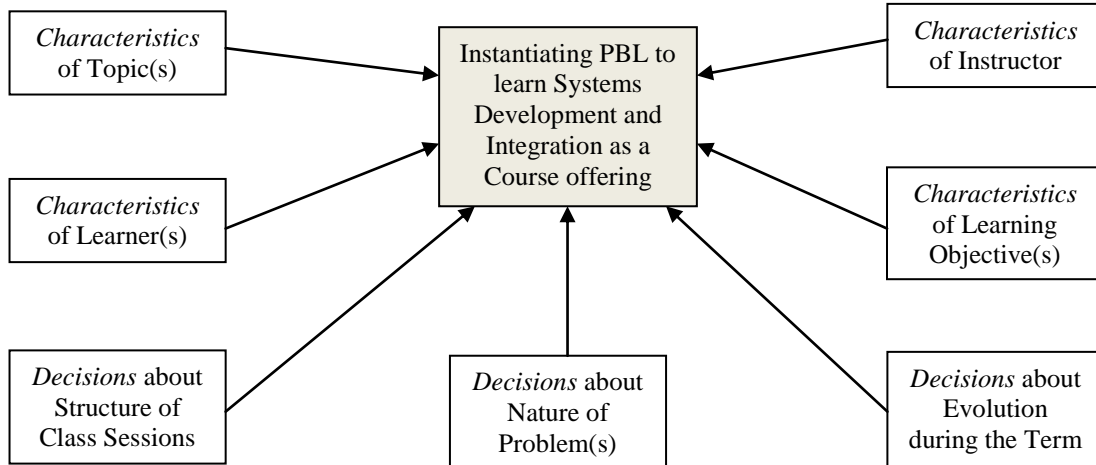


Figure 2. PBL-SoS: A Framework to Structuring PBL to Learn SoS

The dimensions are shown as either *Characteristics*, i.e. properties or attributes that the instructor is likely to inherit based on the educational environment; or *Decisions*, i.e., choices s/he will make about effecting PBL. Together, they outline the form and content of the specific PBL instantiation. Table 1 describes each in more detail and traces it to supporting theories and bases.

Dimension	Description	Prior work in PBL, Learning Theories	Prior work in IS
Characteristics of Topics	Knowledge (technology, design) and Skills (teamwork, organizational)	Know-what versus Know-how (Margetson 2001)	Diverse body of knowledge for ISD (Iivari et al. 2005); empirical studies (e.g. Curtis et al. 1988)
Characteristics of Learners	Traditional versus non-traditional learners, proximity to graduation and learning styles	Myers-Briggs Types and Learning Styles (Felder and Silverman 1988)	None
Characteristics of Instructor	Self-assessed depth of knowledge, experience in pedagogy and content, propensity for risk-taking	Sharing control with students (Barrows 1986; Baptiste 2003)	Nature of expertise and deep knowledge (Vitalari 1995)
Characteristics of Learning Objectives	Learning to integrate information technology and business	Learning objectives (e.g. Bloom 1956), and PBL goals (Barrows 1986)	Desirable behaviors and competencies (Vitalari and Dickson 1983; Tan 1994)
Decisions about Structure of Class Sessions	Balance between lectures, problems and other activities, sequencing, and individual versus team efforts	Models of PBL (Duch 2001) and sharing of control (Barrows 1986)	None
Decisions about Nature of Problems	The nature of problem itself, including characteristics such as size and complexity (which may require assigning prior to class sessions)	The spectrum from cases (Garvin 2007) to PBL (Uden and Beaumont 2006)	Authenticity, experiences and context-sensitive learning (Curtis et al. 1988)
Decisions about Evolution during the Term	Static models for use of PBL versus evolution through the term	None	None

Table 1. Dimensions Underlying the PBL-SoS Framework

The framework can be seen as a set of interacting scales. The *Characteristics* require a priori assessment and, in effect, narrow the range of choices available for the *Decisions*; that is, they are not mutually independent. They have the effect of providing an N-dimensional space, highlighting dimensions that can and cannot be adjusted.

THREE INSTANTIATIONS

We describe three instantiations of the PBL-SoS framework. It is only with hindsight, with the framework in place, that we are able to reflect on the choices reflected in these instantiations. This insight allows us to position the instantiations as partial validation: we can describe the instantiations by mapping them against the dimensions. A more complete validation would require use of the framework with other educators. This paper, instead, describes the three instantiations. One caveat is in order: although the instantiations are described as if they are applied consistently, the student sample and their abilities can and did vary across semesters. PBL, then, allows continuous evaluation, and allows adjusting specific rendering for the term.

Instantiation 1: Enterprise Integration

The first example instantiation comes from an Information School in a large university in a rural setting with enterprise integration as the focus of learning. The course, taught to seniors in the major, was designed with PBL in mind although it has evolved over six years from a case-based teaching practice to a greater emphasis on PBL.

- Topics: Significant propositional knowledge as well as teamwork and organizational skills with topics that range from integration basics, process models, middleware and organizational change, mature research streams and marketplace
- Learners: Traditional students, class size 25-50, close to graduation, potentially varied learning styles
- Instructor: Tenured faculty, considerable knowledge of content, track record of incremental improvements to pedagogy, amenable to risk-taking
- Learning Objectives: Following Bloom’s taxonomy, with increasing emphasis on higher level learning
- Structure of Class Sessions: Wiki-based questions to explore readings, problems for deeper engagement, student-lead sequence, minimal presentations from instructor although slides available, floating facilitator model in class with attempts to move towards peer tutoring, few discussions with complete class
- Nature of Problems: Mix of problems with specific technologies, tools and methods that have specific answers (use of synthetic rules); open-ended problems that require hypothesis generation and exploration; problems solved in dyads or small teams
- Evolution during the Term: Moving from conceptual learning with a focus on synthetic rules to open-ended problems

<p>Example Class Session</p> <ul style="list-style-type: none"> • Students attempt to work, as a dyad, to solve a small set of problems. • Resources (slides, readings and other), made available after first attempt. • Students adjust solutions, discussing outcomes with the instructor who floats between teams. • Student dyads submit final solutions, instructor provides feedback. 	<p>Example Problem 1</p> <p>Consider the organization Painters R Us that we have encountered. You may refer to the description made available earlier. Describe everything that happens from the time someone calls in with a request to paint, and the paint job commences:</p> <ul style="list-style-type: none"> • Who performs each task that you have identified? • Who decides the order in which these tasks must be performed? 	<p>Example Problem 2</p> <p>A new reporting requirement is introduced to qualify for state funding. ABC State University must now integrate the degree audit application with another application hosted and run at ABC Department of Education.</p> <ul style="list-style-type: none"> • What options can you think of to integrate the two applications? Describe three options. • What would be the relative efforts for each option? Which might prove easier to implement?
--	---	--

Figure 3. Instantiation 1, Enterprise Integration

Instantiation 2: Process Design

The second template comes from an IS department at a large university in an urban setting, aimed at learning and teaching about process modeling and design. The course, taught to undergraduate students in the major, was offered from its inception as a PBL-infused course that has evolved over three terms with an increasing emphasis on PBL.

- Topics: Include significant propositional knowledge and teamwork with specific topics such as process modeling and redesign, emerging research streams, unstable marketplace of vendor offerings
- Learners: A 50-50 mix of traditional (full-time) and non-traditional (part-time, working) students, close to graduation,

- potentially varied learning styles.
- Instructor: Tenured faculty, expert knowledge of content, track record of contributions to curricula design, amenable to risk-taking
- Learning Objectives: Becoming entry-level process analysts knowing open-standard models and methods
- Structure of Class Sessions: Student-teams of median size three. Teams discuss problem from last class (done at home, individually). Team comes up with merged, “best” solution. Two teams present solutions that students and instructor critique. Next topic overview and discussion of new topic follows, within-class team exercise related to next topic. Problem is assigned to each student to work at-home, becomes the basis to start next session.
- Nature of Problems: Scenario-style, “in-class” less ambiguous than “at-home” problems
- Evolution during the Term: The pattern of in-class and at-home problems varies based on the concepts being examined

Example Class Session	Example Problem 1 (In-Class)	Example Problem 2 (At-Home)
<ul style="list-style-type: none"> • First segment: teams merge individual solutions to at-home problems from previous class. Two present, critique follows. • Second segment: instructor introduces new topic • Third segment: students work on a simpler problem in teams, instructor answers questions from teams 	<p>Diagram the following three scenarios using BPMN 1.1:</p> <ol style="list-style-type: none"> 1. Task A is followed by Task B. If Task A takes longer than 24hrs, terminate it and proceed to C. 2. Task A is followed by Task B. If B is not completed within 24 hours of start of A, proceed to C 3. Continuing the insurance claim scenario: <ul style="list-style-type: none"> • After a claim is received and validated it is sent to a claims adjusting service for obtaining an estimate • While waiting for the estimate, the response document is prepared. Upon receipt of the estimate, the response document is completed and sent. • If a response is not received in two business days from the claim estimation service, escalation activity is initiated. 	<p>At North State College, James initiates the class registration process by completing a Course Request (CR) form that lists courses he wishes to take that semester. It is first received by James’ guidance counselor, who determines whether it is consistent with James’ program of study, retrieved from the student information system (SIS). If the request is not consistent, or if there is no Program on file, the CR is returned to James stating what must be done. If the CR is approved by the counselor, it moves to the registrar, who checks James’ financial record from SIS. If James doesn’t pass the check, a notice is sent to him indicating this and requesting he remedy it. If It he passes this check, the registrar adds James in each course listed on the CR. If the course is full, James is placed on stand-by. Based on the courses enrolled, a pro forma tuition bill is prepared by Accounts Receivable. The registrar informs James about the outcome (courses enrolled and stand-by); Accounts Receivable sends James the pro forma tuition bill.</p> <ul style="list-style-type: none"> • Use Tibco Business Studio 3 to represent this process. Use swimlanes for each of the roles identified.

Figure 4. Instantiation 2, Process Design

Instantiation 3: Data and Databases in Organizations

The third template comes from a large university in an urban setting, aimed at learning and teaching about database management, broadly mapped to the traditional database course offering in IS departments. The course, taught to juniors and seniors in the major, has been offered for many years in different versions, employing traditional pedagogy.

- Topics: Include significant propositional knowledge and teamwork with specific topics such as data modeling, database design and performance, mature research streams and stable marketplace of vendor offerings
- Learners: Many non-traditional students, not necessarily close to graduation, varied learning styles although no formal assessments done
- Instructor: Tenured faculty, considerable knowledge of content, track record of contributions to curricula design, amenable to moderate risk taking
- Learning Objectives: Following Bloom’s taxonomy, with increasing emphasis on higher level learning
- Structure of Class Sessions: First segment: take-home problems assigned in prior class session, individual solutions brought together in class to form a consensus solution, team presentations, critique by instructor and students. Second segment: a mini-review of the topic. Third segment: In-class team solutions of a new problem related to the current topic, instructor provides any needed assistance; a take-home problem is assigned for individual work.
- Nature of Problems: A mix of problems focused on data modeling (requiring use of synthetic rules) and database design; as well as open-ended problems that require hypothesis generation and exploration; problems solved by individuals before discussion in small teams in class
- Evolution during the Term: Moving from an emphasis on questions to assess learning of concepts to open-ended problems involving application and synthesis

<p>Example Class Session</p> <ul style="list-style-type: none"> • First segment: students compare and merge individual solutions in teams to at-home problems from previous class. Two teams present solutions followed by critique. • Second segment: instructor provides review of new topic. • Third segment: teams work on in-class problem based on current topic. Instructor floats between teams and answers questions. 	<p>Example Problem 1 (In-class)</p> <p>State Pacific (SP) has created a database to support a run/walk event that thousands participate in to support the importance of healthy bodies. You have access to the metadata. Your task is to help SP send Thank-You packages and awards.</p> <ul style="list-style-type: none"> • In yesterday’s event somebody with last name starting with G or C won the race. To help in preparing the First Place Award thank you letter, write a query for displaying information on race times for those whose last name starts with “G” or “C”. • GP wants to send Thank You letters to everyone who participated in yesterday’s event. They will need information that may be scattered in different tables. Write a query that brings together and displays all the information available on the participants. 	<p>Example Problem 2 (Take-Home)</p> <p>Little Bobby Joe has just started a car race track company, Bobby’s Race Track. So far, he is using spreadsheets, tables, paper based forms to keep track of race data, race times, and other related information. As his company grows, he is realizing that he needs a database to manage the data. You are asked to help by developing the database. To start, you would like understand the requirements.</p> <ul style="list-style-type: none"> • Analyze the available forms, reports, data sources and other requirements you can infer to create an initial conceptual model for the database. • How can you use the created to model to find out if a database based on this model would meet the data requirements of the company?
--	---	---

Figure 5. Instantiation 3. Data and Databases in Organizations

The instantiations demonstrate diversity in terms of characteristics of educational environments faced as well as decisions to adapt PBL. They also point to the considerable latitude that educators have for adopting and adapting PBL to the situations faced by instructors as they try to balance the demands of synthetic technology rules and organizational skills.

CONCLUDING REMARKS

The paper has outlined a framework to formalize the characteristics and decisions involved in employing PBL for systems development and integration courses. The work reported draws on an integrative review of prior work in systems development and integration, theories of learning, and the pedagogy of PBL; combined with a retrospective analysis and reflection on the authors’ own practice. For the increasingly wide spectrum of skills necessary for next-generation systems development and integration professionals, PBL provides an effective foundation. There is, however, a significant gap between conceptual prescriptions about the importance and need for PBL and their operationalization. We believe this is partly due to the multiple theoretical influences on PBL. The framework outlined, and instantiations described, provide a start towards actionable approaches for implementing PBL in the context of systems development and integration, answering the call (Dhar and Sundarajan 2007) for a deeper integration between IT and Business for education.

ACKNOWLEDGMENTS

The work reported has been funded by the National Science Foundation under award numbers 722112 and 722141. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation (NSF).

REFERENCES

1. Adams, L., J. Kasserman, A. Yearwood, G. Perfetto, J. Bransford, and J. Franks (1998). The effect of fact versus problem oriented acquisition, *Memory and Cognition*, vol. 16, 167-175.
2. Baptiste, S. (2003). *Problem-based Learning: A Self-directed Journey*. SLACK Incorporated, 2003
3. Bloom, B.S. (Ed) (1956). *Taxonomy of Educational Objectives: The Classification of Educational Goals*, 201-207. Susan Fauer Company, Inc. 1956.
4. Brownsword, L., D. Fisher, E. Morris, J. Smith, and P. Kirwan. (2006). System-of-Systems Navigator: An Approach for Managing System-of-Systems Interoperability. *Technical Note CMU/SEI-2006-TN-019*.
5. Bruner, J. (1966). *Toward a Theory of Instruction*. Cambridge, MA.: Harvard University Press.
6. Curtis, B., H. Krasner and N. Iscoe (1988). A field study of the software design process for large systems. *Communications of the ACM*, 31, no. 11, 1268–1287.

7. Delisle, R. (1997). *How to Use Problem-based Learning in the Classroom*. Association for Supervision and Curriculum Development (ASCD)
8. Dewey, J. and A. Bentley (1949). *Knowing and the Known*. Beacon Press, Boston.
9. Dhar, V., and A. Sundarrajan (2007). Information Technologies in Business: A Blueprint for Education and Research. 18(2), June, 125-141.
10. Duch, B. (2001). Models for Problem-based Instruction in Undergraduate Courses. In *The Power of Problem-based Learning*, B. Duch, S. Groh and D. E. Allen (eds.).
11. Felder, R.M., and L.K. Silverman (1988). Learning and Teaching Styles in Engineering Education. *Engineering Education*, 78(7), 674-681
12. Fisher, D.A. (2003). An Emergent Perspective on Interoperation in Systems of Systems. CMU Software Engineering Institute, *Technical Report CMU/SEI-2006-TR-003*
13. Freeman, P. (1987). *Software Perspectives: The System Is the Message*. Addison-Wesley, Reading MA
14. Garvin, D.A. (2007). Teaching Executives and Teaching MBAs: Reflections on the Case Method. *Academy of Management Learning & Education*, vol. 6, no. 3, September.
15. Guindon, R., H. Krasner, and B. Curtis (1987). Breakdowns and processes during the early activities of software design by professionals. In: *Empirical Studies of Programmers, Second Workshop*. Ablex Publishing. Norwood, NJ, 65–82.
16. Hughes Caplow, J.A., J.F. Donaldson, C. Kardash, and M. Hosokawa (1997). Learning in a problem-based medical curriculum: students' conceptions. *Medical Education*, 31, 440-47.
17. Iivari, J., R. Hirschheim, and H. Klein (2004). Towards a distinctive body of knowledge for information systems experts. *Information Systems Journal*, vol. 14, 313–342.
18. Jones. M. and G. Walsham (1992). The limits of the knowledge: organizational and design knowledge in systems development. In: *The Impact of Computer Supported Technologies on Information Systems Development*, Kendall, K.E. (ed.), 195–213. Elsevier Science Publishers, B.V (North-Holland), Amsterdam.
19. Kolb, D.A. (1984). *Experiential Learning*. Englewood Cliffs, NJ: Prentice-Hall.
20. Lave, J. (1988). *Cognition in Practice: Mind, Mathematics, and Culture in Everyday Life*. Cambridge, UK.: Cambridge University Press.
21. Lave, J. and E. Wenger, (1990). *Situated Learning: Legitimate Peripheral Participation*. Cambridge, UK.: Cambridge University Press.
22. Lee, J., K. Siau, and S. Hong. (2003). Enterprise integration with ERP and EAI. *Communications of the ACM* 46(2) 54-60.
23. Leidner, D.E. and S.L. Jarvenpaa, 1995. The use of information technology to enhance management school education: a theoretical view, *MIS Quarterly*, vol. 19, pp. 265-291.
24. Margetson, D. (1998). Why is Problem-based Learning a Challenge? In: *The Challenge of Problem-based Learning*. D. Boud and G. Feletti (eds.), 36-44. Routledge.
25. Mathiassen, L. (1998) *Reflective Systems Development*. Dr. Techn. Thesis, Aalborg University.
26. Mathiassen, L. and S. Purao (2002). Educating Reflective Systems Developers. *Information Systems Journal*, vol. 12, issue 2, April, 81-102.
27. Naur, P. (1985). Intuition . In: *Software Development*, Ehrig, H. et al. (eds). Lecture Notes in Computer Science 186, 60–79. Springer Verlag, Berlin.
28. Nilsen, H., and S. Purao (2005). Balancing Objectivist and Constructivist Pedagogies for Teaching Emerging Technologies: Evidence from a Scandinavian Case Study. *Journal of Information Systems Education*, vol. 16, 281-293.
29. Piaget, J. (1929). *The Child's Conception of the World*. New York, NY: Harcourt, Brace Jovanovich.
30. Ross, B. (1998). Towards a Framework for Problem-based Curricula. In *The Challenge of Problem-based Learning*, D. Boud and G. Feletti (eds.), 28-35. Routledge.
31. Savery, J.R. and T.M. Duffy (1995). Problem Based Learning: An Instructional Model and Its Constructivist Framework, *Educational Technology*, vol. Sept-Oct, p31-38.
32. Schmidt, H.G., W. D. Dauphinee, and V.L. Patel (1987). Comparing the effects of problem-based and conventional curricula in an international sample. *Journal of Medical Education*, 62, 305-315.

33. Schön, D.A. (1983). *The Reflective Practitioner — How Professionals Think in Action*. New York: Basic Books
34. Skinner, B.F. (1968). *The Technology of Teaching*. New York, NY: Appleton-Century-Crofts
35. Smith, D., L. K.K. O'Brien, and M. Barbacci (2002). Enterprise Integration. *The Architect*.
36. Springer, C.W. and A.F. Borthick (2007). Improving performance in accounting: Evidence for insisting on cognitive conflict tasks, *Issues in Accounting Education*, vol. 22, 1-19.
37. Stolterman, E. (1992). How systems designers think about design and methods. Some reflections based on an interview study. *Scandinavian Journal of Information Systems*, 4, 137–150.
38. Tan, M. (1994). Establishing mutual understanding in systems design: An empirical study. *Journal of Management Information Systems*, vol. 10, no. 4, 159–182.
39. Uden, L., C. Beaumont (2006). *Technology and Problem-based Learning*. Idea Group Inc (IGI Publishing).
40. Vitalari, N.P. and G.W. Dickson (1983). "Problem Solving for Effective Systems Analysis: An Experimental Exploration," *Communications of the ACM* (26:11), 248-956.
41. Vitalari, N.P. (1985). Knowledge as a basis for expertise in systems analysis. An empirical study. *MIS Quarterly*, 9 (3), 221–241.
42. Vygotsky, L.S. (1978). *Mind in Society*. Harvard University Press, Cambridge, MA.
43. Waltz, D.B., J.J. Elam, and B. Curtis (1993). Inside a software design team. Knowledge acquisition, sharing, and integration. *Communications of the ACM*, 36, no. 10, 63–77.
44. Webster, J. and R.T. Watson (2002). Analyzing the Past to Prepare for the Future: Writing a Literature Review. *MIS Quarterly*, 26(2).
45. White, K.B. and R. Leifer (1986). Information systems development success. Perspectives from project team participants. *MIS Quarterly*, 10(3), 214–223.