AC 2011-1047: WORK-IN-PROGRESS: COLLABORATIVE AND REFLECTIVE LEARNING IN ENGINEERING PROGRAMS

Neelam Soundarajan, Ohio State University

Dr. Soundarajan is a faculty member in the Computer Sc. & Eng. Dept. at Ohio State. His interests include topics in Software Engineering and Engineering Education.

Work-in-Progress: Collaborative and Reflective Learning in Engineering Programs

Abstract

The importance of well developed *team-working* skills as well as *reflective* or *metacognitive* skills among engineering graduates is widely recognized. Many undergraduate engineering programs, as well as K-12 programs, have introduced a number of activities throughout the curriculum, ranging from a variety of team projects to the requirement of maintenance of *portfolios* of a student's work as he/she goes through the program, to develop both sets of skills.

At the same time, there has been a heated debate between some researchers who have developed these approaches on the one hand and others who have focused on *knowledge-centered* approaches on the other hand about the relative merits of these two sets of approaches. What has been missing is a suitable way to *integrate* knowledge-centered considerations and collaborative/reflection-centered considerations, with the focus shifting from one to the other as needed. In this paper, the author presents an innovative way of achieving such an integration. The paper situates the work within the *how people learn* framework[1,2].

1. Introduction

Over the last few decades, a number of different approaches have been developed to exploit IT tools and systems in education at all levels from K-12 to undergraduate engineering programs through corporate training. While each of these systems and approaches has its own particular goals and uses its own specific methods to achieve them, they can be broadly classified into two groups which may be called *knowledge-centered* approaches and *activity-centered* approaches respectively. As the name suggests, the central focus of knowledge-centered approaches is on items of knowledge and ensuring that students acquire the knowledge in question. Examples of systems in this group include the work of Anderson[3] which takes an AI approach to the problem and culminated in the powerful ACT (automated cognitive tutor) system[4]; the work of Merrill and colleagues[5] who introduced the notion of *instructional transaction*s and systems based on them; various systems that have been characterized as *multimedia learning* systems[6]; and, most importantly for our work, the many systems that are based on the notion of *learning objects*[7].

By contrast, the focus of *activity-centered* approaches is on such student activities as collaborative team-work, building models of various kinds in constructionist[8,9] systems, and having students engage in *reflection* via such activities as maintaining *portfolios* of their work, etc. The intent, of course, is that these activities will help students develop deep knowledge and understanding of the topics of study; but the main focus, nevertheless, is on the activities. Examples of systems in this group include the CSILE system of Scardamalia et al.[10]; *mindtools* of Jonassen[11]; *CoWeb* of Dieberger and Guzdial[12]; and the work of Bransford and the CGTV group[13].

Although the ultimate goal of both groups of approaches is to effectively exploit IT tools and systems to improve education, the difference in focus has lead to considerable discussion about their respective effectiveness. Indeed, there has been a long-running, sharp debate between the two groups on this question; see, for example, Greeno[14], Jonassesn and Churchill[15] for the activity-centered group's arguments and Merrill et al.[16], Anderson et al.[17] for the knowledge-centered group's counter-arguments. Sfard[18] frames the debate in terms of the *acquisition metaphor* (AM) versus the *participation metaphor* (PM). In AM, we think of knowledge as something the learner wishes to *acquire* from such sources as a teacher, books, or the Internet; in PM, we think of a learner as a *participant* in such activities as reflecting, communicating, inquiring, building models, etc. Sfard presents many examples to demonstrate that it is not that one metaphor is correct and the other wrong; but that in certain situations and at certain stages in the development of a learner, one serves us better than the other. Cobb and Bowers[19] take a similar position.

While there have been calls such as those by Sfard and Cobb & Bowers to adopt a balanced position between the knowledge-centered and activity-centered approaches, there doesn't seem to be any system that actually does so in an *integrated* and *seamless* manner, i.e., a system that effectively exploits IT tools and mechanisms and shifts the focus of the student at the right moments and in a seamless manner from knowledge-centered activities such as working on specific problems on specific topis to such activities as collaborating with a team or reflecting on his/her understanding of earlier related topics and back. Our goal is to develop precisely such an approach.

Our approach is based on the concept of *learning objects*. The idea of learning objects, henceforth LOs, was introduced by Hodgins[20] in the early nineties. While different authors have assigned slightly different meaning to the term, the consensus definition is the one offered by Wiley[21]: "[t]he main idea of learning objects is to break educational content into small chunks that can be reused in various learning environments, in the spirit of object-oriented programming"; we will return to the relation to object-oriented programming in Section 3. Saum[22] offers a historical perspective of LOs. One main focus of the various LO-based approaches has been on *reuse*. In order to enable wide reuse, standards have been introduced that specify how LOs should be *tagged* with *meta-data* so that search engines can easily find them; and repositories of LOs such as MERLOT (www.merlot.org) and ARIADNE[23] have been created. The idea is that a learner or an instructor can, using appropriate meta tags, search for LOs that will meet particular requirements and put them together to create even complete new e-learning systems. A somewhat different approach has focused on defining a suitable structure for LOs. Thus, for example, in the Cisco model, Cisco being an early, influential adopter of the approach, an LO consists of an "overview" component, a "summary" component, 7 to 9 "information objects", and an "assessment" component. But, again, the consideration of reuse has been central.

In effect, the idea is that an LO encapsulates an item of knowledge and it and the associated system are set up to "deliver" that knowledge to interested students. This very much falls in the "acquisition metaphor" (AM) of Sfard. How do we enrich it to allow for the "participation metaphor" (PM) when required? The answer, as we will see in detail later in the paper, is a novel extension of the LO idea so that an LO becomes a *dynamic* object rather than being a *static* entity as it is currently (in all of its various versions). This will enable us to maintain and update information about the specific activities that *individual* students/student groups engage in as they work with a given LO. In effect, an LO will not just be a static item of knowledge but, rather, will keep track of *individual* students' engagement with it over time. Thus each time works with a given LO, the system will have access to that particular student's prior involvements with that LO and, hence, will be able to tailor the activities presented or the manner in which they are presented, etc., based on weaknesses/misunderstandings that the student may harbor about the underlying concepts, as revealed from his/her prior involvements. Moreover, the system will have access not just to information about the student's prior engagement with *that particular* LO but also his/her engagement with earlier, related LOs, including those in prerequisite courses. Further, the dynamic nature of LOs in our approach will enable the system to allow students (suitably constrained) access to their teammates' engagement in the same or related activities and learn from that. As we will see, the approach will even enable instructors to reflect on their experiences with teaching specific concepts and refine their teaching practices based on such reflection.

The paper is organized as follows. In Section 2, we summarize background work. We consider both systems whose primary concern is knowledge acquisition as well as those that focus on student participation in (team and individual) activities. We also consider some underelying learning theories and the *how people learn* (HPL) framework[1,2]. In Section 3, we consider the details of existing LO-based systems, consider a serious shortcoming in these systems which makes them AM-oriented systems, describe a novel approach to overcome the shortcoming, and how the approach will allow us to build rich systems that can account for both AM and PM considerations. We also show how the approach is consistent with the HPL framework. In Section 4 we consider various questions concerning the design of such a system that we are currently addressing.

2. Background: Learning Theories and Instructional Systems

Cognitive information processing: Recent work in cognitive science has focused attention on learning and cognition and such questions as how information is stored in the brain, what happens when a learner's skill with a particular topic increases, etc. A framework for these developments was provided by the work of Anderson[3],[24]. Anderson identifies three distinct stages of development, each being a step up from the previous stage, these being *declarative*, where various facts related to a skill are stored independently and have to be retrieved one at a time; *knowledge compilation*, where related facts are compiled into larger groups and accessed as such; and *procedural* where the groups are applied almost automatically, i.e., without much conscious effort by the individual. Several systems have been built based on this framework.

Instructional transactions: Merrill and colleagues (1996) introduce the concept of *instructional transaction*, the set of interactions necessary for a student to acquire an item of knowledge or skill. The idea is to identify various patterns of transactions and develop algorithms that implement the patterns. The algorithms are based on three types of instructional strategies: presentation (of information); practice (working on problems of varying difficulty); learner guidance (in the form of feedback). The ITT algorithms work with entities known as *knowledge objects* (KOs). A KO contains a name, one or more multimedia objects that represent the knowledge in question, and a free-form description that the KO's author can use to provide information about the KO. There are four types of KOs: *entities* corresponding to things like places, persons, or symbols; *properties* of entities; *activities* that are actions that a learner can perform; and *processes* that change properties of entities. The claim is that suitable transactions, based on the intended learning outcomes, can be defined, given appropriate KOs. A general software "engine" can perform these transactions in the appropriate fashion. Although they were developed indepently, KOs are clearly a type of LOs.

Principles of multimedia learning: The term "multimedia learning" represents the fact that this kind of learning involves multiple streams of input to the learner, for example, words on a screen, pictures and/or animations on the screen, a voice reading some information, etc. The work in this area[25] identifies key principles that should be adhered to if multimedia learning system are to be effective in achieving *meaningful learning* (as against *rote learning*) in various domains. These principles provide guidance to designers of instructional systems. An example may be useful. The *split-attention principle* states that it is important to avoid formats that require learners to split their attention between, and mentally integrate multiple sources of information[26]. Ayres and Sweller[26] present an example: a geometry problem is presented as a diagram showing a polygon with additional internal lines with vertices labeled with various letters; a textually separate description specifies the angles of the polygon using the vertex names listed in the figure. This requires the learner to move attention back and forth between the figure and the text description - a violation of the split-attention principle. A simple change in which the angles are given as part of the figure eliminates the problem and enables better learning. Although not directly related to our approach, these ideas will be of value in designing components of our system.

Goal-based scenarios (GBS): In this approach[27,28], students learn particular skills and knowledge by pursuing, in an on-line simulation, a specific goal. The goal is typically something that makes sense in real life such as learning to reserve a room in a hotel in a foreign country or starting a new business. A scenario, including a cover story and a clear statement of the goal, is presented to the learner. The system includes videos depicting portions of the scenario and, based on the learner's responses to questions posed thus far, further portions are shown or other information is provided to help the learner make progress toward achieving the goal. The approach mainly focuses on knowledge acquisition but caters to a limited extent to PM-considerations.

Knowledge-building systems: Scardamalia and Bereiter[29] (see also Scardamalia et al.[10]) make the case for what they call *knowledge-building*, as against *knowledge-reproduction*, in educational programs. Knowledge-building focuses centrally on the development of deep understanding as against simply memorization and organization of knowledge. They argue that most IT-based educational systems have focused on supporting knowledge-reproduction by providing facilities for such actions as copying, deleting, storing, retrieving, displaying, etc. This criticism applies also to more advanced systems that might provide tailored instruction based on the needs of the individual learner. The point is that such systems "cast the student into the role of recipient of knowledge rather than active constructor of it"[29]; this is, of course, the AM/PM debate.

The approach of Cognition Technology Group at Vanderbilt (CGTV)[13] has some similarities to the CSILE system of Scardamalia et al.[10]. The key point of the theory underlying CGTV is to situate learning, via problem solving, in the *context of realistic problems* rather than present one issue at a time to the learner. Hence learners learn to make connections between various ideas in the domain and, occasionally, across domains. Rather than mechanically learning and applying various formulas to compute various things without developing the ability to determine which formula to apply under what conditions, students are forced to analyze problems carefully, identify the steps that must be taken to solve them, and decide how each step can be performed.

While CSILE and the CGTV-group's systems impose important constraints on how students engage in their learning activities, *CoWeb*[12] uses a different model. As in the case of the Wiki[30] on which it is based, a CoWeb is a collection of pages that (typically) do not enforce a strict author/reader distinction; instead, every user has the same rights and may edit any page or add new pages. Thus, depending upon how learners use these abilities, they will be able to build knowledge collaboratively. CoWeb provides a set of facilities that facilitates the typical kinds of changes that users might make but they can be tuned to the needs of the specific course or whatever. In addition, it is possible to impose restrictions on what changes to a given page can be made by which users, etc. This ability to fine tune its facilities, the fact that a user needs only access to a standard Web browser to use it, combined with the fact that students and instructors are almost universally familiar with web pages, are attractive aspects of the approach that we will adopt in our system.

Constructivism/Constructionism: The idea underlying constructivism is that learners are not "empty vessels" to be filled with knowledge by the instructor[31]. Rather, learners understand a new topic by constructing mental models of the item. To this, constructionism[8,9,32] adds the idea that this can happen most effectively when they make tangible models, possibly in software, of the item in question. Jonassen[11] makes a compelling case for how students can benefit by using various IT tools from spreadsheets to databases to construct models of a wide range of types of items. The main point is that many IT tools enable students to build interesting, possibly simplified, models of various things. And, by doing so, students develop a deep understanding of the item in question.

The How People Learn (HPL) Framework: Much understanding has emerged over the last few years about how people learn, especially with regards to science, mathematics and engineering; see, for example, Bransford et al.[1]; Donovan and Bransford[33]; the National Research Council report on the topic[34]; Pellegrino et al.[35]; and Redish and Smith[36]. A pictorial representation of the HPL framework appears in Fig. 1. The figure is based on one in[1]. Metacognitive skills play a key role in the framework. Metacognitive skills are activities that allow us to reflect on and control performance in a useful and efficient manner. A report of the NRC Committee on the Foundations of Assessment[35] stresses the importance of metacognitive skills: "One of the most important aspects of cognition is meta-cognition – the process of reflecting on and directing one's own thinking." In order to help students develop these skills, the learning environment as well as the students' activities must be designed to "focus on making students' thinking visible to both their teachers and themselves."

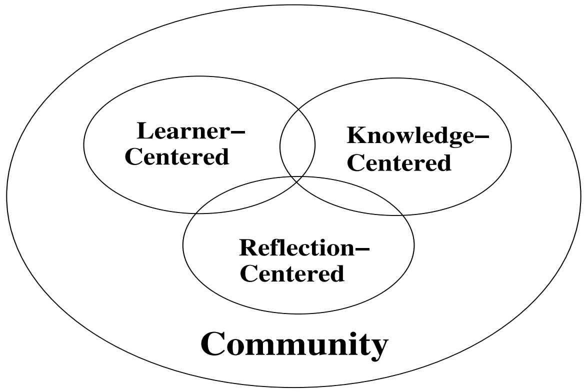

Figure 1: HPL Framework

One change from the original is that the oval labeled "reflection-centered" in Fig. 1 is labeled "assessment-centered" in the original. We made the change because the essential purpose of assessment is to provide, timely feedback that helps the students to *see for themselves* exactly how well developed their skills are with respect to various items of knowledge and ability in the given discipline, and what they need to do to move to the next level. As Royer et al.[37] put it, "metacognitive skills are cognitive activities that allow an individual to reflect on and to control performance in a useful and efficient manner. Skilled performers within a domain possess the capability of planning their activities, and altering behavior in accordance with the monitoring activity. Less skilled performers are far less proficient at this monitoring process and, correspondingly, less successful at applying the skills they do possess." In a real sense, it is the ability to reflect on one's performance, learn from the results of that performance, and refine one's knowledge or skill that not only helps improve the performance but marks the transition to becoming a professional in the discipline. The overlap in the ovals labeled knowledge-centered and reflection-centered corresponds to the fact that the reflective activities by the student are based on the assessments of the knowledge items represented by the knowledge-centered components.

Consider next the oval labeled "learner-centered" in Fig. 1. A key observation regarding student learning[2,36] made in recent research on how people learn is that "[s]tudents come to the classroom with preconceptions –often incorrect– about how the world works, which include beliefs and prior knowledge acquired through various experiences ... effective teaching [must] elicit the preexisting understanding and provide opportunities to build on, or challenge, these initial understandings." This poses an important problem since the "incorrect preconceptions" and the "preexisting understanding(s)" may vary from one student to another. How do we account for these variations? As we will see, the ability of the LOs in our system to save information about *individual* students will allow us to address this problem directly. Note also the overlap between these three ovals.

The final aspect of HPL, the one that, as depicted in the figure, ties the others together, is *community-centered*-ness. The importance of *teamwork* has been widely recognized in the last few years with graduates' adequate ability to engage in teamework being one of the key requirements of the accreditation criteria[38] that undergraduate engineering programs must meet. The various team projects that engineering programs have introduced at various points in their curricula are intended to develop these skills. However, as Johnson and Johnson[39] note, "simply placing students in groups and telling them to work together [which is essentially what many programs have done] does not in itself promote higher achievement". The richer model of LOs that forms the basis of our approach, as we will see in Section 3, will enable students not just to work in teams but to learn from each other and to help each other in the central fashion implied by HPL.

The key ideas of HPL may thus be summarized as follows as a set of conditions that the learning environment should be designed to satisfy:

1. *Learner-centered*: To account for the knowledge, skills, preconceptions, and common misconceptions of the learners;
2. *Knowledge-centered*: To help students learn with understanding by thinking qualitatively and organizing their knowledge around key concepts;
3. *Reflection-centered*: To find ways to monitor progress and provide formative, timely feedback to help students be aware of their own understanding and gaps in that understanding;
4. *Community-centered*: To foster norms that encourage students to learn from one another.

3. Richer Model of Learning Objects for Enabling Learning and Teaching

We start with a discussion of an important weakness in the current approaches to learning objects. The problem does not seem to have been discussed in the literature but it is one that seriously impacts the usefulness of the approach in higher education including engineering programs. The problem may be less significant in corporate training and, indeed, this may well be why LOs, as they exist currently, have found widespread acceptance in that setting but despite considerable effort, few university curricula in any field use LOs to any large extent.

Before considering the details of the problem and our solution, it may be useful to summarize some background material from Computer Science, especially the field of object-oriented programming– with apologies to any reviewers/readers with technical background in Computer Science! "Objects" in software engineering has a well-defined technical meaning. An *object* is an instance of what is called a "class", a *class* being essentially a *template* that specifies what information an *instance* of the class, i.e., *objects* of this type, will contain; and defines a set of operations (often also called "methods") that can be performed on these objects. One key point is that while different instances (i.e., objects) of a given class share the methods, each instance has its own copy of the information it contains. This information is called the "state" of the object; and this state can vary over time. Indeed, when we apply a method on an object, that will typically change the state of *that* object, and not that of any other instance of the same class. A simple example might serve to make the concepts concrete. A common use of object-oriented programming is to build software simulations of real-life situations. For example, consider a modern office whose functioning is to be simulated. One of the *classes* in such a simulation might be called `CopyMachine`. When the simulation actually runs, *multiple* instances of this class would be created, each representing one of the copy-machines in the building. Suppose, for example, there are three copy-machines M1, M2, and M3, in the actual office; when the simulation runs, three instances, I1, I2, and I3, of the `CopyMachine` class would be created and would correspond respectively to M1, M2, and M3. The information in each instance would simulate the state of the corresponding machine, for example, the amount of paper left in the machine, the number of copies that have been made on that particular machine, etc. Methods defined in the `CopyMachine` class will be applied to particular instances. For example, applying `makeCopies(10)` on the object I2 would correspond to making ten copies on the machine M2 and would update information about the number of copies made, the state of the toner cartridge, etc., that is maintained in the internal state of I2; the states of I1 and I3 will remain unchanged during this operation.

Learning objects, as currently used (in any of the existing approaches), are not like this. LOs do have something like methods but one does not think in terms of the "state" of the object as changing with time. One might ask, so what? Why would we want the state of an LO to change over time? If an LO just defined, for example, a collection of methods that can take a student through a set of activities (such as solving increasingly complex problems) over a topic such as velocity and acceleration as part of a course in physics, and contained some methods that can be used to assess how well the student has learned the topic, why should such an object change over time? The answer is that if we had a separate learning object for *each* student who goes through the particular lesson, then the state of the particular object could, if it can change over time, contain information about *that* student's *progress* through the lesson. So, for example, if a student went through part of the lesson in one session and then came back to it at a later point, the system will automatically know, by examining the state of the object associated with that student, where he/she had left off, and the system can start from there. Or, if the information contained in the state of the object suggests that the student had difficulties with some earlier part of the lesson, the system can determine that it might be appropriate to use some methods to help the student *review* those parts of the lesson; etc. Essentially the states, at any given time, of the particular objects associated with individual students enrolled in the course can provide detailed information about the *specific* students' progress (or lack thereof!) upto that point in time, in mastering the topic in question. The existing LO approaches do not do this. Instead, they focus on the topic knowledge to be conveyed and do not allow for recording of student-specific information about individual students' progress in their understanding and skill with respect to the topic. In one sense, learning objects, as they exist now, essentially ignore the *learner*!

The above discussion should also suggest the natural solution to the problem, the one we have adopted in our work: Replace what are currently called "learning objects" by *learning classes*. Each learning class will correspond to a particular topic and will consist of resources such as notes related to the topic, suitable assignments and exercises for the students to work on, etc. The class will provide appropriate methods that will allow students to engage in the related activities (or, where appropriate, allow instructors to guide students though the activities). For each student who goes through the topic represented by the given learning class, we create a *learning object* whose state will be used to record information about the progress of *that* student through *that* topic. To see this in more detail, consider a course, say, the physics course mentioned above. Suppose $T1, \ldots, Tn$ is the list of topics in the course. Corresponding to each of these topics, we will have a separate learning class; thus, we will have classes $L1, \ldots, Ln$, that correspond respectively to $T1, \ldots, Tn$. For each student enrolled in the course, we will have a collection of *learning objects*, these being instances of the classes $L1, \ldots, Ln$; henceforth in the paper, unless otherwise noted, we will use the term "learning object" and its abbreviation LO in this new sense. More precisely, when a student S first goes through the topic represented by, say, $L3$, our system creates a corresponding learning object s_{l3}. Information about S's performance in this particular topic will be recorded in the state of s_{l3}. It is important to note that s_{l3} will contain a *temporal* record of S's performance in this particular topic. That is, once s_{l3} has been created (when S first engaged with this topic), each time S interacts with $L3$, for example to solve additional problems related to the topic, review some course notes, etc., information about this new interaction will be added to s_{l3}. Most importantly, each time this student S interacts with $L3$, information in s_{l3} will be used to *fine-tune* the interaction. For example, if the information in s_{l3} indicates that this particular student (S) had difficulties solving a certain type of problem related to the topic in question ($T3$), the system can present additional problems of that type to him during this interaction, rather than moving on to more advanced aspects of $T3$. Conversely, if the information in the object s'_{l3} corresponding to a different student S' indicated that that student had mastered the initial aspects of $T3$, the system can present to her problems related to the more advanced aspects of $T3$.

This is only a start. Typically, the different topics $T1, T2$, etc. in a course will have close relations between each other with one topic building on one or more earlier topics. And, typically, how a student performs in the activities that are included as part of a given set of topics will give a good indication of which aspects of a follow-up topic he or she will have problems with and which aspects of that topic the student will find easy to understand and master. What this means is that the information contained in a particular student's learning objects corresponding to those earlier topics contain valuable information that our system can use to tailor the activities of the later topics when presented to that particular student. Equally important, the *student* (possibly with the help of the instructor) can review the information contained in these learning objects to identify specific weaknesses that the student may have in his or her knowledge and skills related to the topics in question; of course, in order to allow students and instructors to comfortably engage in these activities, the system will have to provide appropriate browsing and navigation facilities and we will return to this in the next section. This can be of value not only when the student is still enrolled in the particular course but also later such as when deciding which other follow-up courses to take. In effect, the learning objects associated with an individual student as he or she goes through the various courses in the curriculum provide a complete record, in a real sense the *learning history*, of the student's activities through the engineering program. If appropriate browsing and navigation facilities are provided, they will enable the student, when engaged in an activity in any course, to look back and reflect on his or her work in the most directly related activities in the current course as well as in earlier courses in the curriculum. Thus this type of reflection is not something that is tacked on as an additional component of the curriculum but is naturally integrated into the activities of various related courses in the program and serves directly to support the student's learning. Moreover, the student sees this activity of looking back at the work on related activities in earlier courses as directly helping master the knowledge and skills in the current course rather than as an additional activity distracting from his or her focus on the current course. Over time, reflection over earlier relevant activities becomes a natural part of the student's intellectual makeup.

It might be worth noting here that, in some respects, what we have described above has some similarities to *computerized adaptive testing* (CAT) (see, for example, Wainer[40]). But in CAT, the adaption of questions/problems presented to the student are based on the student's performance in earlier questions/problems presented during the *same* test. There is nothing analogous to our learning objects that enable the system to store information about a student's performance in various activities not in just one course but over all the courses in the curriculum and to utilize this information to fine-tune the activities presented to the student. Nor is there any notion of enabling the student to reflect on the relevant earlier activities.

Let us now turn to the *collaborative* aspect of HPL. As noted earlier, simply requiring student groups to work in teams, as many engineering programs have done, does not necessarily develop team skills. Thus, for e.g., Barron's[41] work showed that team performance depends critically on the kind of interactions that the team engages in. Her work showed that given two teams, members of which were academically equally gifted, with one team's members engaging in each other's thinking the members of the other being focused primarily on their own ideas, the former was much more likely to arrive at higher quality solutions to complex problems than the latter. Thus, to enable collaborative learning, we must ensure that students in a team have a good understanding of each others' points of views and the ideas that those views are based on. Given this, in order to enable collaborative learning, we adopt the following approach in our system. The interface that the system presents to students in engineering coures will be a specialized *wiki* that borrows ideas from systems such as CoWeb[12] but is built around the *learning histories* of the students. The interface will allow students not just to look back at their own activities in the current and earlier courses but also to learn from each other and from each others' histories.

In more detail, when creating a new learning activity in a given course, the instructor will create a corresponding page in the wiki that will contain (in it or in pages it links to) relevant notes, assignments, practice activities, etc. When a student engages in any part of this activity, a new page will be created for him/her which will record the student's learning history with this activity; i.e., the page will represent the *learning object* for this student for this activity. Moreover, the student will have a "course page" for each course he/she has registered in. This page will provide information about the course, a "my team" link that will link to the course pages of other students in the class who are on his/her team, and links to the pages corresponding to the various activities in the course. If the student has not yet engaged in a given activity, the corresponding page will be a default page containing standard information about the activity provided by the instructor and easily accessible buttons to start engaging in the activity. Once the student does some of the work involved in the activity, this page will be modified to include information about what he/she did. This will include a description of what he/she did, the instructor's comments and feedbacks about the work and suggestions for the student might do next, *and* comments from team mates. When he/she wants to review the activity, the page will allow the student to go over what he/she did in full detail. Thus even in the case of learning activities that would not normally be considered "team-activities", the system will enable and encourage students to learn collaboratively.

The system will provide analogous facilities for the instructor so that he/she has a "teaching portfolio" organized by activity, including links to information about earlier related activities possibly in other courses and containing additional notes he/she may wish to add – including links to particular student's pages to highlight special difficulties that some students or some groups of students may have, etc. This will allow the instructor to be alert to potential problems the next time he/she teaches the same course. Again, reflecting on the teaching and using of that information to improve course activities will become second nature to instructors. Moreover, the system will provide links in instructor's pages to pages of other instructors of the same course. This will facilitate collaborative work among instructors in the same way as among students.

4. Discussion

Over the last several years, much work has been done on using IT technology in innovative ways to improve student achievement. While this work had its beginnings in the sixties/early seventies, it has accelerated with the wide availability of the Internet and, especially, the Web. Along a different thread, much has been learned over the last few years about how people learn and how to improve the effectiveness of educational programs. A key result of this work has been the development of the *HPL* learning framework. The HPL framework and related work has highlighted the importance of *reflective* and *collaborative* learning in ensuring strong student outcomes.

The point of departure for our approach is the deliberate focus on facilitating reflective and collaborative learning to such an extent that they become integrated components of student learning. An innovative aspect of the approach is the manner in which it ties together, in a very natural way, the idea of *learning objects*, which thus far has been used primarily in approaches that have focused on the *knowledge-centered* component of HPL, with reflective and collaborative activites. Three key ideas underlying our approach may be summarized as follows. First, for each student engaging in a learning activity, we create a separate learning object that records information about that individual student's work, over time, with the particular activity. This information makes it possible for the system to tailor the individual student's further interactions with that activity such as presenting more/less challenging activities based on that student's current understanding and skill with earlier components of the activity. Second, the collection of learning objects associated with an individual student for the various activities he or she goes through in a given course, and the various courses in the curriculum that the student has gone through so far, provides a detailed history of the student's curricular activities. With the help of suitable browsing and navigation tools that allow the student to easily pinpoint those parts of the history that are most relevant to the task that the student is currently working on, the student will be able to use this history to recall what he or she learned earlier and apply and build on that knowledge; this is precisely the point of reflective learning. Third, carefully controlled access to the learning histories of some of the other students and tools that allow the student to focus on the relevant parts of those histories, make it possible to learn not just from other students but from their past learning experiences as well.

We conclude with a brief mention of some important questions that need to be addressed in implementing a prototype of our system. First, the browsing and navigation tools that allow students access to key parts of their (and other students') learning histories must be carefully designed so that they are unobtrusive and, at the same time, provide the right facilities. This, as one of the referees noted, will be challenging. It will require considerable experimentation and fine-tuning. Similarly the tools that provide instructors who may not be IT-experts facilities to create suitable wiki pages for various activities and to access individual students' learning histories will also require experimentation and fine-tuning. Finally, development of suitable methods for assessment of the actual impact of the tools on improving students' reflective and collaborative skills is part of our plans for future work. Our plan is to develop the system for use by Computer Science and Engineering students and instructors since these users can be expected to be more tolerant of interfaces that non-computing users might find less friendly. We use the approach of *design experiments*[42,43,44] to evaluate and modify the system's design over a number of iterations. The idea underlying design experiments is to use formative evaluation of the tools to revise the afforances it provides; and is very appropriate for the type of system we are building. If this paper is accepted for presentation at this and other ASEE conferences, the discussions at the conferences should provide further ideas for tuning the system.

References

1. J Bransford, A Brown, and R Cocking. *How people learn: Brain, mind, experience, and school*. National Academy Press, 2000.

2. J Pellegrino. Rethinking and redesigning curriculum, instruction, and assessment. Paper commissioned by the National Center on Education and the Economy, 2006. available at: www.skillscommission.org.

3. J Anderson. Acquisition of congitive skill. *Psychological Review*, 89:369–406, 1982.

4. J Anderson, A Corbett, K Koedinger, and R Pelletier. Cognitive tutors: lessons learned. *Journal of the Learning Sciences*, 4:167–207, 1995.

5. M Merrill and ID2 Research team. Instructional transaction theory. *Educational Technology*, 36:30–37, 1996.

6. R Mayer. *Cambridge handbook of multimedia learning*. Cambridge Univ. Press, 2005.

7. D Wiley. *The instructional use of learning objects*. Association for Instructional Technology, 2002.

8. D Jonassen, K Peck, and B Wilson. *Learning with technology: A constructivist perspective*. Merrill/Prentice-Hall, 1999.

9. S Papert. Computer-based microworlds as incubators for powerful ideas. In *The computer in the school: Tutor, tool, and tutee*, pages 203–210. College Press, 1981.

10. M Scardamalia, C Bereiter, RS McLean, J Swallow, and E Woodruff. Computer supported intentional learning environments. *Journal of Educational Computing Research*, 5:51–68, 1989.

11. D Jonassen. *Modeling with technology: Mindtools for conceptual change*. Merrill/Prentice-Hall, 2006.

12. A Dieberger and M Guzdial. CoWeb - Experiences with Collaborative Web spaces. In *From Usenet to CoWebs: Interacting with Social Information Spaces*, pages 1–13. Springer, 2002.

13. Cognition Technology Group at Vanderbilt (CGTV). From visual word problems to learning communities. In *Classroom lessons: Integrating cognitive theory and classroom practice*, pages 157–200. Cambridge, 1994.

14. J Greeno. On claims that answer the wrong questions. *Educational Researcher*, 26:5–17, 1997.

15. D Jonassesn and D Churchill. Is there a learning orientation in learning objects? *Int. Journal on E-Learning*, 3 (2):32–41, 2004.

16. MD Merrill, L Drake, MJ Lacy, and JA Pratt. Reclaiming instructional design. *Educational Technology*, 36(5): 5–7, 1996.

17. J Anderson, I Reder, and H Simon. Situative versus cognitive perspectives. *Educational Researcher*, 26:18–21, 1997.

18. A Sfard. On two metaphors for learning and the dangers of choosing just one. *Educational Researcher*, 27(2): 4–13, 1998.

19. P Cobb and J Bowers. Cognitive and situated learning perspectives in theory and practice. *Educational Researcher*, 28:4–15, 1999.

20. W Hodgins. Into the future: A vision paper. http://www.learnativity.com/download/MP7.pdf, 2000.

21. D Wiley. Connecting learning objects to instructional design theory. In *Instructional use of learning objects*, pages 3–23. AIT/AET, 2002.

22. R Saum. Abridged history of learning objects. In *Learning objects for instruction*. Information Sc., 2007.

23. E Forte, Haenni, and Warkentyne. Semantic and pedagogic interoperability in ”ARIADNE”. In *SIGMOD*, pages 20–25, 1999.

24. J Anderson. *Architecture of cognition*. Harvard University Press, 1983.

25. R Mayer. Cognitive theory and the design of multimedia instruction. *New directions in teaching and learning*, 89:55–71, 2002.

26. P Ayres and J Sweller. The split-attention principle in multimedia learning. In *Cambridge handbook of multimedia learning*, pages 135–146. Cambridge, 2005.

27. RC Schank, A Fano, B Bell, and M Jona. The design of goal-based scenarios. *Journal of the Learning Sciences*, 3:305–346, 1993.

28. R Schank, T Berman, and K Macpherson. Leraning by doing. In *Instructional design theories and models, II*, pages 161–181. Lawrence-Earlbaum, 1999.

29. M Scardamalia and C Bereiter. Technologies for knowledge-building discourse. *Comm. of the ACM*, 36(5):37–41, 1993.

30. B Leuf and W Cunningham. *The Wiki Way: Quick collaboration on the Web*. Addison, 2001.

31. M Driscoll. *Psychology of learning for instruction*. Allyn and Bacon, 2000.

32. S Papert and I Harel. *Constructionism*. Ablex, 1991.

33. S Donovan and J Bransford. *How students learn history, science and mathematics*. National Academy Press, 2005.

34. NRC Report. *Learning and understanding: Improving advanced study of mathematics and science*. National Academy Press, 2002.

35. J Pellegrino, N Chudowsky, and R Glaser. *Knowing what students know: Science and design of educational assessment, Report of the NRC Committee on the Foundations of Assessment*. National Academy Press, 2001.

36. E Redish and K Smith. Looking beyond content: Skill development for engineers. *Journal of Engineering Education*, 97(3):295–308, 2008.

37. J Royer, C Cisero, and M Carlo. Techniques and procedures for assessing cognitive skills. *Review of Educational Research*, 63(2):201–243, 1993.

38. ABET. Engineering accreditation criteria. ABET website at: http://www.abet.org, 2007.

39. DW Johnson and RT Johnson. Cooperative learning and achievement. In *Cooperative learning: theory and practice*, pages 23–37. Praeger, 1990.

40. H Wainer. *Computerized adaptive testing*. Routledge, 2000.

41. B Barron. When smart groups fail. *The journal of the learning sciences*, 12(3):307–359, 2003.

42. A Brown. Design experiments: Theoretical and methodological challenges in creating complex interventions in classroom settings. *Journal of Learning Sciences*, 2:141–178, 1992.

43. A Collins. Towards a design science of education. In *New directions in educational technology*, pages 15–22. Springer, 1992.

44. P Cobb, J Confrey, A deSessa, R Lehrer, and L Schauble. Design experiments in educational research. *Educational Researcher*, 32(1):9–13, 2003.