CPATH CB: A community for lab-centric computer science instruction

1 Vision, Goals, Objectives, and Outcomes

We propose to transform undergraduate computing education on a national scale by understanding, growing, and providing an online space for a community of stakeholders in *lab-centric* instruction. Lab-centric instruction is a significant extension to the widely advocated use of supervised closed-labs as a required part of the course. In lab-centric courses, lecture and recitation time is traded wholly or in large part for supervised labs that incorporate collaboration, embedded assessments, and tutoring-like interventions by the instructor.

Our vision is a successful community that will significantly expand the use of lab-centric instruction throughout the CS curriculum, and can therefore achieve on a national scale the benefits we have observed within the lower-division at Berkeley[1]. We propose a three-year project to achieve a set of related milestones:

- Assemble a diverse team of leaders in communities related to lab-centric instruction: CS curriculum, pedagogy, teacher education, instructional tools, digital repositories, and educational technology. Over the course of the project, this team will refine the vision and goals of the lab-centric community.
- Evaluate the state of lab usage and barriers to adoption of lab-centric formats in courses across the range of higher-educational instructional contexts.
- Create an online space to support the growth and functioning of the lab-centric community.
- Recruit instructors to develop and refine a small number of lab-centric segments for inclusion in upper-division CS courses. This will serve both as a pilot for our online space as well as exemplars for upper-division lab-centric courses.
- Provide outreach for the lab-centric community and the online center and dissemination of the project results through a variety of educational forums.

A lab-centric format differs substantially from its more traditional counterpart, with students as well as staff taking on new educational roles. In lab-centric versions of an introductory programming course for non-majors and a data structures course, we have observed numerous benefits over and above those claimed for lab-augmented instruction:

- a wider variety of activities, including frequent assessments and online and offline collaborations;
- frequent opportunities for staff to engage students in targeted tutoring to counter confusion;
- evidence of enhanced student performance, especially among students traditionally at risk in CS courses;
- a much enhanced ability to detect previously unrecognized misconceptions;
- a cooperative and relaxed classroom climate in which student learning is the main focus.

---

[1] This effort is part of the UC-WISE project, the University of California Web-based Instruction in Science and Engineering, partially funded by NSF ("Online curricula for monitored, closed-lab first-year CS courses", DUE 0443121, $249,000, May 2005-May 2008). This project has contributed to lab-centric curricula development in the lower division. Relevant publications include Ryan(2006).

1.1 Lab-centric instruction: what is it?

Parker *et al.* (Parker 1990) note that "from the beginning, the discipline of computer science has been a laboratory science [standing] alongside biology, chemistry, geology, and physics." Consequently, a formal lab session—scheduled, supervised, with tasks specifically intended for completion during the session—is a component of many introductory programming courses. (Such a section is referred to as a "closed lab", in contrast to an "open lab" where attendance and tasks are left up to the students.) The lab session is usually two to three hours long, meeting once a week. Its purpose is mainly to *supplement* or *complement* classroom presentations. We will apply the term *lab-augmented* to courses or course segments organized in this way.

Lab-centric instruction trades lecture and recitation time for more hands-on lab work. Instead of playing a secondary role to in-class lectures or discussions, the lab sessions are the *primary* resource for student learning. Lectures and discussions, where provided, arise from and supplement the lab content rather than vice versa. Table 1 below summarizes the differences between a lab-centric course and its more traditional counterpart.

Traditional (lab-augmented) course	**Lab-centric course**
The student attends 2-3 hours per week of lecture.	The student attends 0-1 hours per week of lecture.
The student works through a set of short exercises, perhaps with a partner, in 2 hours per week of lab.	The student works through a wide variety of exercises in 4-7 hours per week of lab.
The student attends 1 hour per week of "discussion", in which examples similar to upcoming homework exercises are covered.	Students engage in structured collaboration activities; occasional impromptu discussions may arise in lab with a neighboring student, or between lab instructor and a group of students.
The student works on homework, not always clear about skills and techniques that need to be applied. Help, when available, is found through a class newsgroup, fortuitously scheduled teaching assistant office hours, or communication with fellow students.	Most exercises formerly done as homework are now completed in lab. Relevant context is clear from earlier exercises. Help from instructors is always available in the lab section.

Table 1: Comparison between traditional and lab-centric courses, from a student's perspective.

A course may be entirely lab-centric, as described in Table 1, or it may have lab-centric segments. For instance, a project group in a software engineering course may meet with an instructor for a tutorial on collaborative work, for a design review, or for a code walkthrough. A lab-centric segment in an operating systems course might focus on concurrency, including activities visualizing thread activation, debugging a race condition, or brainstorming on a design to split processing equally among threads.

1.2 Why use lab-centric instruction: the potential for revitalizing undergraduate computing education

Any closed lab is better than none, we claim. (Research, as surveyed by McCauley *et al.* (McCauley 2004) generally agrees.) Hofstein and Lunetta (Hofstein 2003), surveying characteristics of labs in science courses, note that they provide an environment for experimentation and inquiry, they facilitate student community and collaboration, and they foster positive and improved student attitudes and interest in science. These characteristics apply to computer science closed labs as well.

Parker *et al.* (Parker 1990) note in addition that closed lab activities facilitate the integration of theory and practice, and alleviate the tension between learning and evaluation.

The UC-WISE project (see footnote 1) has for the last four years been experimenting with lab-centric organizations of our CS 1- and CS 2-level courses. Each has one lecture hour and six hours of supervised lab per week. All lab activities are delivered online, using a learning environment that provides facilities for assessment, reflection, and collaboration. We have noticed several advantages in the resulting "learner centered" environment (Bransford 2000) over and above those provided by lab-augmented instruction.

- Numerous commentators, e.g. (McKeachie 2005; Elbe 1976), have observed that a lecture is likely to lead at best to *passive* "learning". Our replacement of most lecture time by hands-on lab time instead engages students in active learning, apparently without increasing the total workload in the course. We have some evidence (Clancy 2003) that learning improves as well.
- Lab activities include frequent embedded assessments that keep students apprised of their progress in the course, provide opportunities for staff intervention when students are confused, and motivate students to seek help if necessary. The lab format, together with the frequent assessments, allow instructors to engage in one-to-one and group tutoring sessions, fostering a kind of apprenticeship learning (Collins 1989). Tutoring has long been known to be the best method of instruction, and has been shown to work without extensive training on the part of the instructor. The main drawback of tutoring—that it is very time-intensive and, therefore, costly—is reduced by the lab format. Students spend most of their time learning from the materials and their collaborators, while the tutoring is efficiently used when confusion occurs.
- Research (Johnson 1991; Hoadley 2004; Dubinsky 1997) has shown that collaborative activities have a myriad of benefits in all phases of learning and coursework. Good lab-centric activities should include collaboration, both on- and offline. In UC-WISE, "gated collaboration" presents students with a prompt; after answering, they can review the responses of their classmates. (Contrast this with a question posed to students in a recitation section. Ideally, all students consider the question and contribute their answers to discussion; in practice, only a few consider the question seriously, and only a few—always the same few—contribute answers.) Focused discussion activities provide opportunities for sharing strategies for debugging or avoiding particular kinds of errors, for comparing design or development approaches, and for criticizing and defending alternatives.
- Activities are structured so that the increase in difficulty as students move between activities is small. This contrasts with the significant differences in complexity between lab exercises, homeworks, and projects in lab-augmented courses. The fine-grained activities expose a substantial amount of information about student conceptions and misconceptions. We have learned in several years of our lab-centric introductory programming course of a variety of previously unrecognized misunderstandings, and have modified or invented activities to address them (Ryan 2006).
- The lab-centric format helps staff to get to know the students, and encourages students to get to know and help each other. Opportunities for communication patterns that lead to a competitive and defensive climate (as described by Barker *et al.*, 2002) are limited. Evaluation ratings for lab teaching assistants have been uniformly good. Moreover, qualitative analysis (from surveys and interviews) suggests that UC-WISE technology and pedagogical approaches promote positive views towards computer science by women and non-CS majors.

- Our courses provide a much wider variety of activities than their traditionally taught counterparts, allowing more students to shine and supporting a wider variety of learning styles and cognitive abilities. Shy students are encouraged to contribute in collaborative exercises, and receive help without seeking it out. Students who might otherwise drop out are kept in the course with the support provided by staff and the learning environment: our courses have succeeded with groups of students typically at risk for success in computer science.
- The lab-centric format fosters more efficient interaction with staff. Consider the obstacles presented to a student seeking help in a traditional course. Only short focused questions are appropriate for the class newsgroup. Staff office hours may not coincide with the student's free time. A staff member, presented with the student's question, will lack context—why is the student asking this question?—and history—what does this student already know?—on which to base a response. In contrast, students in our lab-centric courses ask questions in the context of the day's lab activities, consulting a staff member who most likely has been observing the student's work throughout the term.
- Lab-centric instruction provides benefits to instructors as well. The reduced amount of lecture frees up time to spend on activities such as t.a. training, review of student work to diagnose problems, and inventing new pieces of curriculum. Instructors may thus concentrate on their strengths, devoting efforts to the aspects of teaching that they do best and enjoy most. Lectures that remain can be tightly focused on *actual* rather than *assumed* student needs. (Techniques such as peer instruction have the same goal, but in our opinion achieve it less effectively.)
- Teaching assistants in our lab-centric courses work essentially the same number of hours as in the traditional version. Extra time needed to supervise lab sections is offset by reducing or eliminating office hours; reducing or eliminating preparation for recitation section; and, because most questions are answered in lab, a drop in class-related electronic mail and newsgroup traffic.

Thus, we have identified and implemented an important new model for undergraduate computing education. The next step—the CPATH vision—is embodied in this proposal: to begin to replicate this model across a variety of programs and institutions, forming partnerships among academic institutions and engaging national leaders to help.

2 Implementation Plan

During this three-year project, we will undertake six main tasks:

[1] Work with the leadership team.
[2] Evaluate the state of lab usage in CS instruction.
[3] Develop an online center for the lab-centric community.
[4] Sponsor the creation of exemplars of upper-division lab-centric course segments and associated materials.
[5] Engage in outreach and dissemination.
[6] Evaluate this project and the growth of the lab-centric community.

These tasks involve many subtasks spread across the length of the project. Figure 1 provides a timeline of our intended progress, with indications of who will engage in each task and how they relate to the main outcomes of the project. The rest of this section will describe each of the above tasks in more detail, and conclude with descriptions of the leadership and management teams.

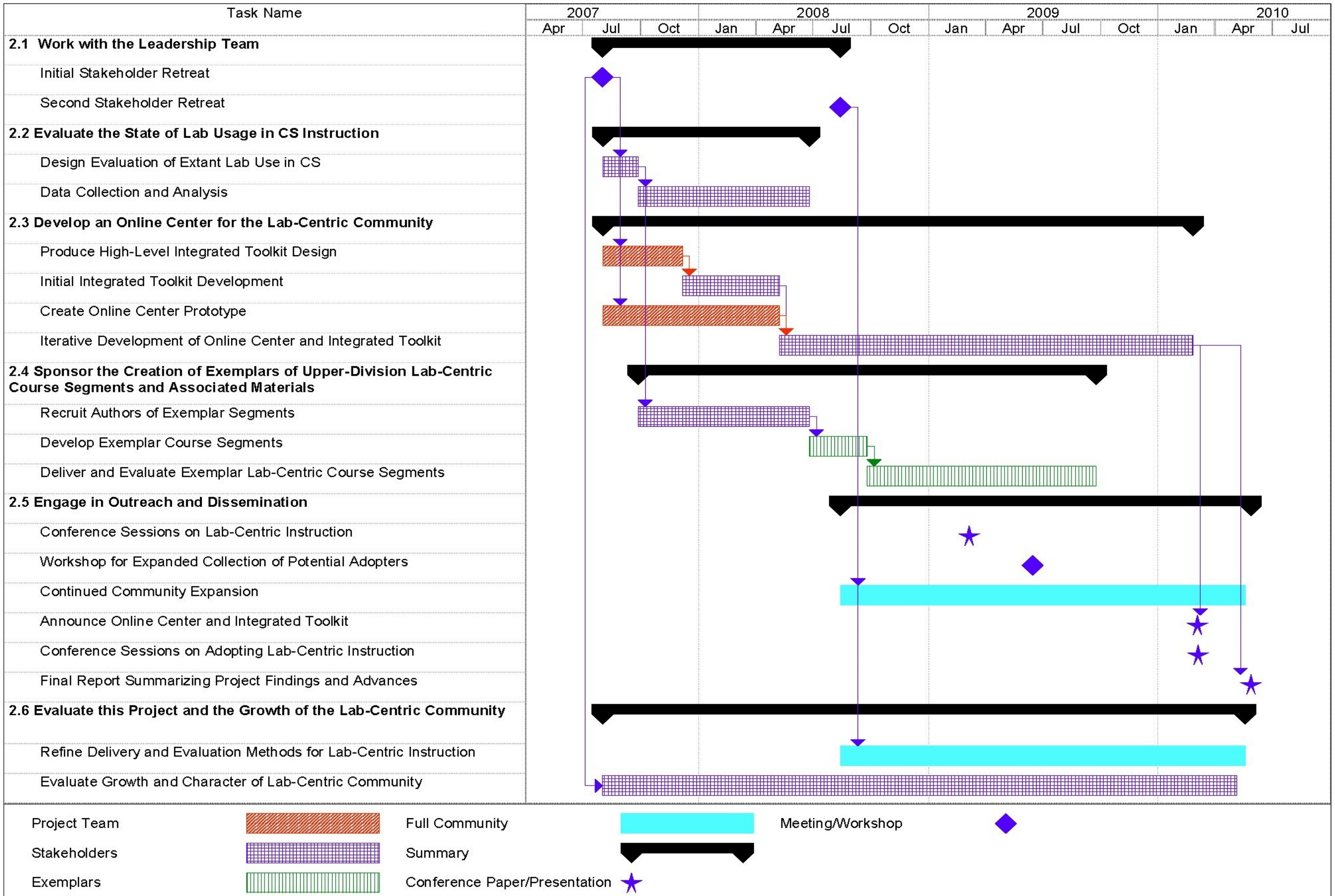

Figure 1: Project schedule referenced by responsible party

2.1 *Activity 1: Work with the leadership team*

We will assemble our leadership team in both summer 2007 and summer 2008 in Berkeley. (Section 2.7 discusses the membership of this team). The initial retreat will last three days. While the members are all familiar with the general issues in lab-based courses in CS, an issues packet will be mailed one month prior to the meeting to ensure a strong start.

There are four main goals to accomplish at the initial summer 2007 retreat:

1. Define what a successful lab-centric community should embody, including what should constitute "success", how it can sustain itself, what sorts of roles will comprise the community, and who else might appropriate for the leadership and general membership. Section 1 above contains our initial thoughts on some of these issues; we expect this beginning to be expanded and made more precise during the initial retreat. Part of this goal will involve initial thoughts on activity 3, the online center; more work will be undertaken in the second retreat. Also related is the goal of initiation, through the interaction of this diverse team of leaders, of new lines of research related to lab-centric instruction.
2. Fully define the benefits of lab-centric instruction and the constraints that hinder its adoption in CS instruction. Information will be collected to understand the full range of lab use in CS courses, and to delimit which uses we consider lab-centric. This will include advice on activity 2, evaluating the state of lab usage in the CS curriculum.
3. Begin work on evaluation metrics for the use of labs in CS instruction, with particular attention to differences between courses without labs, with lab augmentation, and with lab-centric formats.
4. Identify tools that support lab-centric instruction, delivery, and authoring. These will primarily be existing tools, although some effort will be taken to identify tools that don't exist but should. Additionally, ways to integrate these tools will be examined with the aim of committing resources to particularly promising possibilities.

The leadership team will communicate throughout the academic 2007/08 year, and reports will announce commencement of or progress on other project activities. The second retreat in summer 2008 will last two days; as new stakeholders are identified, it is likely to include more members than the initial retreat. It will focus on three main goals:

1. Reassess findings from the initial retreat, with particular attention to the results of the surveys on lab usage.
2. Review work that has taken place on the tool integration and the online space for the lab-centric community. Ensure that new courses of action that became possible over the previous year have been considered.
3. Support the exemplar course development (action 2.4 below), which should be in progress. (Exemplar developers will probably include members of the leadership team.)

Beyond the course of this project, we expect members of this team to be active leaders in the lab-centric community, as researchers or practitioners.

Characteristics of the lab-centric community

A community that successfully promotes lab-centric instruction will need active participation from a range of different participants. The collective experience of a successful lab-centric community should include the following.

- Teaching at all areas and levels of the undergraduate CS curriculum: lower-division and upper-division courses; stable and cutting-edge courses; theory, software, architecture, and applications; community colleges, liberal arts colleges, research universities, technical schools, and industry.
- Pedagogical expertise: with active learning techniques such as inquiry, collaboration, and assessment; with curricular mechanisms for apprenticeship learning such as case studies and reciprocal teaching; with management of individual differences such as learning styles and common misconceptions; and with evaluation.
- Teacher education, especially relating to use of technology in the classroom.
- Tool and infrastructure development expertise: with instructional tools such as simulations, visualization, or microworlds; with pedagogical technology such as software for collaboration, communication, and assessment; with curriculum authoring and analysis tools; with infrastructure for storage and access of curriculum segments; and with user interface evaluation.
- Commercial content providers.

To keep the community active and responsive to new developments, participants need places and times to organize and communicate. These should include workshops for prospective instructors; regular exposure at major conferences, and perhaps smaller dedicated conferences; online forums for novices and experts to meet and discuss relevant issues; and evolving repositories of materials ranging from tutorials, white papers, and curriculum at various grain sizes.

2.2 Activity 2: Evaluate the state of lab usage and barriers to lab-centric adoption.

McCauley and colleagues (McCauley 2002; McCauley 2006) have surveyed CS departments starting 1995, and collected data about the use of closed labs. Table 3 presents these data. Each table entry contains the number of institutions reporting use of closed labs and the percentage of all respondents that this number represents.

year	CS 1	CS 2	CS 3	other
1995	35 (60%)	33 (57%)	10 (17%)	17 (29%)
1996	39 (70%)	23 (41%)	8 (14%)	22 (39%)
1997	39 (74%)	28 (53%)	11 (21%)	18 (34%)
1998	58 (71%)	37 (45%)	—	28 (34%)
1999	44 (70%	31 (49%)	—	21 (33%)
2001	33 (73%)	24 (53%)	—	15 (33%)
2004	123 (85%)	—	—	—

Table 3: Data on prevalence of closed labs

The main weakness of this data is lack of any information about the character of lab use. We will gather richer data from institutions using labs in order to understand their instructional milieu and to measure the potential for and the barriers to adoption of lab-centric formats. Dr. Titterton will head this activity, with advice from the leadership team. Most of the data collection will take place in fall 2007, with additional data collection and analysis stretching into spring 2008. The resultant data will be used to inform the outreach and dissemination efforts (activity 5), by identifying instructors and researchers interested in participating in the wider lab-centric community.

Data collection will involve searches of public information, electronic surveys, and phone interviews. The unit of analysis will be higher education computing departments, although we may get responses from multiple faculty at a single institution. Population membership will be gathered via lists of higher education institutions, SIGCSE mailing list, and McCauley's (2002) population (derived from CSAB).

We plan two rounds of surveys, a smaller initial survey with no sampling and a larger, more comprehensive survey sent to a randomly drawn sample with follow-ups to increase response rates. We believe that phone interviews to a smaller sample will also be worthwhile, both to pilot the comprehensive survey and to gather further information.

Refinement of the instruments will occur during the initial leadership retreat, We have preliminarily identified constructs and items for institutions that run closed labs as follows: history and rationale for setting up lab courses, which courses use labs, whether labs are required, the delivery and authoring technologies used, contact hours and durations of sessions and office hours, lab curriculum development practices, the types of activities that students do in the labs, the role of staff, faculty and/or staff satisfaction, how teaching credit is determined, and description of the physical and IT facilities. Constructs and items for institutions that do not run labs will differ, of course.

Comments on barriers to lab-centric instruction

Thus far, lab-augmented instruction in CS is mainly a feature of introductory courses. Moreover, it has not been universally adopted, in marked contrast to the role of labs in physics, chemistry, biology, and engineering curricula outside CS. What follows are some initial thoughts about the absence of labs, and accompanying responses.

Shortage of resources (facilities, staff): Indeed, current closed labs require seats in a room outfitted with computers. As more of the typical campus becomes accessible via wireless networks and more students have laptop computers, however, the restriction to specially configured lab rooms disappears.

We have found at Berkeley in large-enrollment courses with multiple sections run by teaching assistants that the shift from lab-augmented instruction to lab-centric instruction is revenue-neutral. In courses with insufficient lab instructor resources, a lab-centric curriculum could be designed to include more collaborative exercises to help students teach one another.

At many institutions, particularly smaller liberal arts colleges, instructors teach small class sessions of 25 or fewer students. If computers are accessible in the classroom, this teaching situation can easily be converted to a format that includes lab-centric instruction. (The number of class lab hours will of course depend on instructor availability.) If not, the reality of ubiquitous computing will soon eliminate this constraint.

Satisfaction with status quo: There are still numerous CS departments with *no* experience with structured labs. Especially in upper-division courses, there are few alternative models to emulate. Instructors at these institutions claim that their students are learning just fine. We ask, how do they know? The confusion and blind alleys that students encounter in homework are almost invisible to instructors who teach in a traditional system. Moreover, the preference for the *status quo* ignores the current nationwide problems of reduced enrollment, especially among populations traditionally at risk (women, minorities).

Admittedly, there are few good metrics for success of labs (Ma 2006). Soh *et al.* (2005) report progress on a framework for evaluating programming labs, and researchers involved in science education are also making progress in this area (e.g. Psillos 2002). We expect that our proposed community will have some ideas for effective evaluation.

We also note that the time for experimentation with alternative models of instruction is now, when enrollments are relatively low.

Student inconvenience: The time a student spends in lab in lab-centric courses is comparable to lab time in a variety of courses in biology and chemistry. We have already noted the learning benefits that result from attendance at a supervised closed lab.

Teaching credit determination: Lecture is partly replaced in a lab-centric course by a variety of other instructor activities (Clancy 2003), and thus a strong argument can be made for preserving teaching credit while switching to the lab-centric format.

High buy-in cost: That's one of the major issues that our proposed community will address: justification for increasing structured lab sessions throughout the curriculum, advice and assistance for instituting labs, exemplar lab-centric course segments, help with educating lab instructors about unpracticed methods of interaction with students, access to online advice, and support for authoring quality activities.

2.3 Activity 3: Online center development and tool integration.

Authoring and delivery tools: After the initial leadership team retreat, a list of instructional tools and possibilities for integration will be delineated. For example, the leadership team may find that tighter integration between Sakai and NSDL's Engineering Pathway digital library would provide specific benefits. Resources will be directed to the most promising integration candidates in fall 2007. This work will continue through the course of the project; however, sufficient infrastructure should be available to be piloted by the designers of the exemplar materials (activity 4 below) in summer 2009. Iterative development and new integration efforts will continue after this date.

An incomplete list of tools likely to be of interest include Sakai, UC-WISE, Engineering Pathway/SMETE (Agogino 2004), CITIDEL, PACT (Carle 2006; see below), JHAVE (Naps 2005), and INFACT (Tanimoto 2000).

The online space for the lab-centric community: Work on the online space for the lab-centric community will begin in fall 2007, after the initial leadership retreat. The online space will contain white papers and tutorials; access to repositories; community spaces (wikis, mailing lists, and so forth); guides and practical manuals for the use of instructional tools suited to lab-centric formats; professional development resources; and evaluation instruments and tools. We will strive to integrate and adapt existing facilities rather than reinvent wheels. Existing virtual community sites for educators will serve as models for our efforts, e.g., (TappedIn 2004).

The online space will be piloted in summer 2009 by the exemplar developers, and will be rolled out in fall 2009 (see activity 5 below).

The goal of the online space is to support the functioning of the lab-centric community. One important aspect of this process is developing a common vocabulary to talk about the challenges in lab-centric instruction and their solutions. To simplify this process, we will seed the curriculum design aspects of the online center with references to *pedagogical* patterns, analogous to software design patterns. Work in this area (Sharp 2000) was initiated in an effort to duplicate the success of of the software design pattern movement within the space of curriculum design. The goal is to identify outstanding teaching practices and record them in a pattern format that facilitates a common vocabulary for pedagogical research and practice, is accessible to novice instructors, and encourages the repurposing and reuse of techniques that are solidly grounded in modern pedagogy. With this information in hand, a novice instructor or a newcomer to a particular style of teaching (such as lab-centric instruction) should be able to determine if the pattern applies to her situation.

To facilitate the adoption of best pedagogical practices and guide community members towards the use of pedagogical patterns, we propose using the Pattern-Annotated Course Tool (PACT) (Carle 2006; Carle 2007) as a key piece of the online center. PACT is a visual curriculum editor designed to unify a number of curriculum design tasks under a common platform that steers the user towards best practices in pedagogy. In PACT, collections of learning objects or course activities can be arranged into temporally sequenced groups representing the in-class delivery of the curriculum. The main feature of PACT is the ability to annotate these course activities with references to pedagogical patterns and other design metadata. The resulting artifact, a pattern-annotated course, is a reification of both the curriculum and the instructional expertise that was needed to create it. This direct connection between theory and real-world practice allows PACT to present the knowledge of experts in a way that is useful for other course designers.

It is in this capacity as a learning tool that PACT has the most utility for building a community of lab-centric computer science instructors. Instructors participating in the community (both experts at lab-centric curriculum design and relative novices) can build descriptions of their courses in PACT. Annotated references to pedagogical patterns and other metadata will reveal a great deal about the designer's thought process and intentions to the rest of the community.

Pattern-annotated courses created by experts are rich resources for novice instructors in the process of learning a new pedagogy or curriculum platform. Each course is a real-world structure that the learning instructor can relate to directly. She has likely taken a course very similar to the one annotated or may even be familiar with the expert instructor that designed and annotated it. For these reasons, the pedagogical patterns used in the curriculum take on a concrete meaning that cannot be achieved in the abstract. Similarly, the annotated courses created by educators who are new to lab-centric instruction are likely to reveal misconceptions or other errant design decisions that can be identified by more experienced members of the community.

Further, these annotated courses will serve as an evolving metacognition tool for the growing community. The stakeholders involved in the project will learn more about which techniques for lab-centric instruction work and which don't as the community advances. These discoveries will be cataloged in the growing repository of annotated courses contained within PACT. This repository will prove beneficial throughout the community membership. Pedagogical pattern authors can carefully inspect the course designs of other experts to look for new robust patterns. Curriculum designers can learn from the designs of others, both highly successful courses and those that failed for well-documented reasons. Professional development specialists can use the repository to train new instructors in best practices for lab-centric instruction. Finally, it is likely that unanticipated uses for the repository will emerge from other groups within the community during the course of the project.

2.4 Activity 4: Exemplar lab-centric materials in upper-division courses.

Starting in fall 2007, we will identify three instructors interested in designing and integrating lab-centric segments (the longer, the better) into their upper-division courses. (Compensation for this curriculum development is only a small percentage of the total project budget.) Efforts will be made to include at least one instructor with a sizeable population of traditionally underrepresented students. The development of the lab-centric materials will be supervised by the leadership team, with appropriate training given. The courses will be run in the following academic year with the lab-centric component(s), and evaluated.

This curriculum development process will contribute to the project in several ways:

- Quality, exemplar curriculum for lab-centric instruction will be created and publicly available. Few lab materials exist for upper-division courses, let alone lab-centric materials.
- Early evaluation of the lab-centric format as defined by the leadership committee, especially for upper division courses, can take place in these courses. Baseline data (surveys, exams/homeworks, and projects) will be collected in the 2007-2008 academic year.
- The lab-centric community will grow as the curriculum developers gain experience creating and running lab-centric courses.
- Exemplar developers will use pilot versions of the online space and integration toolkit and push their refinement.

Curricular opportunities for lab-centric instruction

Lab activities are appropriate in every area of computer science. The report "Computing Curricula 1991" (Tucker 1991) divided the areas of computing knowledge into "knowledge units", most of whose descriptions included accompanying lab activities. Table 2 lists the nine areas described in the 1988 report "Computing as a Discipline" (Denning 1989), along with the number of knowledge units in each area for which lab activities were suggested.

Area	**Knowledge unit count**	**Number of knowledge units for which lab activities were suggested**
Algorithms and data structures	9	9
Architecture	7	4
Artificial intelligence and robotics	2	2
Database and information retrieval	2	2
Human-computer interaction	2	2
Numerical and symbolic computation	2	1
Operating systems	10	5
Programming languages	12	7
Software methodology and engineering	5	5

Table 2: Knowledge units in "Computing Curricula 1991" for which lab activities were suggested.

For the knowledge units in programming languages and operating systems for which no lab activities were mentioned, it is not too difficult to devise some; for instance, an ITiCSE working group report (Joyce 1997) describes a disk scheduling simulation lab project along with other sample lab activities for use in upper-division courses. Discrete mathematics did not appear in the "Computing Curricula 1991" list; however, courses in this area may also be taught with lab activities (Hein 1993; Fenton 1999; O'Donnell 2006).

Basically, supervised lab exercises are appropriate for *any* of the following learning activities:

- acquiring and solidifying understanding of a computing agent (language or application environment, used either directly or through simulation): constructing and testing a model of computation, acquiring and managing multiple representations of the model;

- acquisition and use of implementation techniques within an environment;
- evaluating and analyzing artifacts constructed in an environment;
- scaffolded design; and
- reflection (learning from experience).

The apprenticeship learning embodied by lab-centric instruction, if implemented in advanced courses, will surely "contribute to the development of a diverse, agile workforce with the computing knowledge essential to U.S. leadership in the global innovation enterprise."

2.5 Activity 5: Outreach and dissemination.

An ITiCSE working group chaired by Joyce and Knox (Joyce 1997) laid out a design for a repository for CS lab material. We feel that the proposed community will be able to improve on this design in our online center. For example, the center will allow access to a collection with links to the Engineering Pathway, CITIDEL, and Computer Science Teaching Center digital libraries. Community members will be able to catalog, evaluate, discuss and download curricular materials associated with lab-centric CS instruction. PI Clancy will serve as editor of this collection.

The collection will contain lab-centric courses and course segments, with accompanying PACT annotations. It will also contain position papers on various aspects of lab-centric pedagogy, authoring tools, community spaces (wikis, mailing lists, and so forth), and evaluation tools.

Our online resource will also make available material with which prospective teachers of lab-centric pedagogy can familiarize themselves with the differences between the new and the old formats. Devising appropriate quality activities that align with existing tools is a challenge. Teachers require guidance and mentoring to help them make the best use of the lab-centric format (Slotta 2004). For example, we have observed (Clancy 2003) that some lab instructors are reluctant to use technology for tracking student activity, considering it "spying" on the students. Students in a lab-centric course benefit greatly from a proactive lab instructor; shy teachers may initially have difficulty in this role. Many teachers find challenges in interacting with individuals or small groups that encounter problems.

To publicize the project, we plan to run several information sessions at conferences starting in fall 2008 on the current and near-future state of lab-based and lab-centric instruction. In summer 2009 we will run a larger workshop, targeting around forty people, for instructors interested in lab-centric formats (i.e., non-leadership members of the community). Attendance will be free, but no travel costs will be reimbursed. Finally, we will run sessions on adopting lab-centric instructions at conferences.

2.6 Activity 6: Evaluation of project success and the lab-centric community

There are several evaluative efforts in this project that have already been discussed: evaluating the state of lab usage in CS departments (activity 2), assembling metrics for evaluating lab-centric materials (activity 1) and using them to evaluate the success of the exemplar courses (activity 4).

This section, however, discusses evaluation of the lab-centric community to determine which of our efforts are working well and which need to be reconsidered. The success of this project is best measured in terms of the growth and strength of the community. Andy Carle will head this task, which will take place over the course of the project. (This work is expected to become part of his doctoral dissertation). We will employ the Communities of Practice (CoP) framework (Lave and

Wenger, 1991), frequently used as a unit of analysis in Human-Centered Computing projects. Wenger identified three descriptive dimensions of CoPs:

What is it about? The *joint enterprise* that the community pursues, as understood and continually renegotiated by its members. In this project, the joint enterprise will begin as a loosely defined focus on lab-centric instruction. Over time, the group is likely to hone in on specific aspects of this topic that appear intrinsically compelling to the community.

How does it function? The *mutual activities* and collective communication that make the group a unique social entity. Example activities might include the discussion of best practices in lab-centric instruction, the communal review process for newly created curricula, and reflection on designs to create new pedagogical patterns. We will track this dimension with analysis of both face-to-face meetings and contributions to the online center.

What has it produced? The *communal resources* that the group has created over time. The creation of communal resources is a major goal of this project and will provide an excellent means of tracking the progress of the community. These artifacts will include developed lab-centric curricula, PACT annotated course descriptions, pedagogical patterns and other curriculum design guidelines, and conference papers and sessions.

Our evaluation of these three dimensions will focus on the community of lab-centric instructors and the interplay between these individuals and the technology that supports the virtual community between face-to-face meetings. The tools provided by the online center, particularly argumentation tools such as PACT and the forum, will be crucial to all three dimensions of the community. A significant aspect of our evaluation will be determining the usefulness and usability of these tools in facilitating the development of joint enterprise, mutual activities, and communal resources.

An example potential data point in this analysis could be the movement of new members of the community from the periphery of participation towards full group membership. Understanding how smoothly this process occurs along with the factors that influence it will help us improve the online center and the leadership habits of the people directing the community.

2.7 Leadership team

Our initial recruitment of stakeholders was guided by the CPATH goals of engaging national leaders and promoting partnerships among institutions, and by our determination (see Section 2.1) of the collective range of expertise required in the group. Among candidates familiar with CS pedagogy, we sought particular expertise in the constraints that hinder the adoption of labs in CS courses, considering both lab-centric and the better known lab-augmented formats.

This group listed below will comprise our leadership team for the first retreat. During the retreat and as the project ramps up, we expect that new stakeholders will be identified, and that the leadership team will grow. For instance, there is currently no community college or industrial representation.

- Tom Naps (curriculum, pedagogy, technology): Professor of CS at University of Wisconsin, Oshkosh. Author of many intro programming textbooks; long-time advocate of labs; co-PI of an NSF-funded project to integrate algorithm visualization into CS education; served on numerous ITiCSE working groups considering resource collections for CS education and evaluating the educational impact of algorithm visualization.
- Owen Astrachan (curriculum, pedagogy): Professor of the Practice of CS and Director of Undergraduate Studies at Duke University. A leader in introductory programming education;

textbook author; co-PI on an NSF-funded grant to provide modules and courses for ubiquitous and mobile computing; member, ACM Education Council.

- Joe Bergin (curriculum, pedagogy. technology): Professor of CS at Pace University. One of the leaders in the Pedagogical Patterns and the Extreme Programming communities; served on numerous ITiCSE working groups on resources for CS 1 education, visualization, and teaching of object-oriented programming.
- Mark Guzdial (curriculum, pedagogy, technology): Professor of CS at Georgia Tech. Leader in CS Education research; member, ACM Education Board and ACM Education Council; substantial experience with collaborative technology; designer of innovative introductory programming courses in media computation.
- Shawn Newsam (curriculum, technology): Assistant professor and founding faculty at University of California, Merced. Instructor, UC-WISE version of CS 1 and CS 2.
- Marcello Kallman (curriculum, technology): Assistant professor and founding faculty at University of California, Merced. Instructor, UC-WISE version of CS 1.
- Dan Garcia (curriculum, pedagogy, technology): Lecturer in CS at Berkeley. Member, ACM Education Council; PI, Werner Lecture Archives project (http://wla.berkeley.edu); instructor, UC-WISE version of Berkeley's version of CS 1.
- Dan Klein (curriculum): Assistant professor of EECS at Berkeley. Recent instructor of CS 188, "Introduction to Artificial Intelligence".
- James Demmel (curriculum): Professor of Mathematics and EECS at Berkeley. Co-designer and recent instructor of CS 267, "Applications of Parallel Computers".
- Alice Agogino (pedagogy, teacher education, technology): Professor of Mechanical Engineering at Berkeley. A leader in research in engineering education and diversity; PI for the NEEDS (National Engineering Education Digital Library System) and the new Engineering Pathway digital library.
- Marcia Linn (pedagogy, teacher education, technology): Professor in the Graduate School of Education at Berkeley. Pioneer in the research of learning technologies and of gender-related educational issues; currently co-PI of NSF-funded TELS Project ("Technology-Enhanced Learning in Science; www.telscenter.org), which brings university researchers together with middle school and high school educators to improve instruction in science.
- James Slotta (technology, pedagogy, teacher education): Associate professor of Curriculum, Teaching, and Learning, University of Toronto. Long time involvement with WISE project on which UC-WISE is based; co-PI of TELS project.

2.8 Project Management Team

We have a strong management team that can achieve all of the desired outcomes in this project. All three individuals are committed to the full length of the project.

Michael Clancy

PI Michael Clancy is a Senior Lecturer in the EECS Department at University of California, Berkeley. Clancy has strong links to all the communities from which our stakeholders are drawn.

- CS curriculum: Clancy has coordinated the evolution of Berkeley's CS lower-division curriculum since 1977, and has taught courses in the upper division as well. He was chair of the Advanced Placement CS Test Development Committee from 1987 to 1992.

- Pedagogy: Clancy is one of the leading CS education researchers in the U.S., and his experience spans a wide variety of pedagogy: self-paced instruction, labs, case studies, pair programming, peer instruction, and lab-centric courses. He has been involved in several NSF-supported projects. He has been on the program committees for both International Computing Education Research conferences, for the 1999 SIGCSE symposium, and for the Seventh Workshop on Empirical Studies of Programmers.
- Teacher education: Clancy worked with high school teachers while on the AP CS committee. As Faculty Advisor for Graduate Student Instructors in the CS Division, he designed courses for training teaching assistants and for course design. (He also won the campus Faculty Award for Outstanding Mentorship of Graduate Student Instructors in 2004.) His current collaboration with Marcia Linn includes serving on dissertation committees of two of her graduate students whose research involves issues of professional development.
- Technology: With Marcia Linn, Clancy devised online learning environments for novice programmers learning LISP in a project funded by NSF. More recently, he has led the UC-WISE project (also partially NSF-supported), which aims to provide technology and curricula for laboratory-based higher education courses. He is currently working with Dan Garcia to expand the use of the Werner Lecture Archive in UC-WISE courses. With Alice Agogino, he helped write two unfunded NSF ITR proposals. He is co-advising Andy Carle.

Nathaniel Titterton, Ph.D.

Researcher Dr. Nathaniel Titterton is a specialist in the EECS Department at University of California, Berkeley. He received his doctorate at U.C. Berkeley in Education, focusing on instructional technology and statistical understanding. His graduate work included the design of computer-based statistical learning environments and research on the learning management and content management systems. He has a M.A. in Statistics from U.C Berkeley, and has consulted as an evaluator on several large projects.

He has taught the lab-centric version of U.C. Berkeley's introductory programming course for non-majors for several semesters. He was involved in the design of UC-WISE since 2002, and currently maintains the system and supervises undergraduate students getting credit for evaluation, design, and implementation of UC-WISE related projects.

Andy Carle

Andy Carle is a Ph.D. student of Human Computer Interaction in the Berkeley Institute of Design and Computer Science Division at the University of California, Berkeley. Andy specializes in instructional and educational technologies and is the lead designer and evaluator for the PACT project. He has authored papers on PACT and other computer science topics as an active member of the ACM's SIGCHI and SIGCSE organizations. He is well versed in the relevant literature of Human Computer Interaction and Human-Centered Computing, including evaluation techniques, user-centered design, communities of practice, computer-mediated communication, and social computing. In addition to his HCI research, Andy has interests in pedagogical reform and has taught lower division CS courses for several summers at Berkeley. He will work as the graduate student for the duration of the project. This project will contribute directly to his dissertation work on computer-mediated professional development for university professors, and we are lucky to have committed interest from him.