PROJECT DESCRIPTION

1. Project Vision, Goals, Objectives and Outcomes

In the narrowest sense, "computational thinking" is the mindset that students need to acquire in order to work effectively with computational systems. More broadly, however, it is a way of understanding the world—one that transcends mere methodology and which is likewise transportable across a wide variety of human endeavors. Exploring the implications of this notion, and of its practical and theoretical applications, has always been the core subject matter of computer science. Our computational systems have changed over the decades, but the fundamental habit of mind has not. If anything, it has become more and more apparent that computer science is not about the machine.

The synopsis for the CPATH program notes that "despite the deep and pervasive impact of computing and the creative efforts of individuals in a small number of institutions, undergraduate computing education today often looks much as it did several decades ago." On the one hand, we regard such stasis as unproblematic, to the extent that computational thinking, as outlined above, has remained largely invariant throughout its history. However, we take very much to heart a later statement, which declares that computing has broadened and now "require[s] integration of multidisciplinary domains." In our view, computer science curricula that do not address this broadening—which seek change through minor adjustments to the languages, systems, and engineering paradigms taught to undergraduate students—do not address the ways in which computational thinking pervades life in the modern world.

We therefore propose what we consider a radical re-thinking not only of our core curriculum in computer science, but of the role of computer science at the university level. In our conception, "computational thinking" is neither easily separated from other endeavors nor easily balkanized into a single department. We therefore imagine a computer science program that is always inextricably linked to other domains. We further understand these domains to include not only the subjects ordinarily thought of as cognate with computational thinking (like bioinformatics or computational physics), but with such notions as "humanities computing," "arts computing," and "music computing."

Our term for this new approach, "**Renaissance Computing**," is intended to evoke that period of history in which computational thinking arguably first began—a period marked by the single lack of hard and fast lines between subjects that has now overtaken university curricula. We imagine an undergraduate program in which students are prepared for the challenges of any one computational domain by virtue of their exposure to multiple domains in which computational thinking plays prominent a role. Our program imagines discrete tracks into computational thinking, but also emphasizes learning environments in which students are able to see the broad transferability of computational notions into other areas.

With this Concept Development and Planning proposal, we seek funding in order to (1) evaluate the needs of academic units outside of CS proper, (2) create specifications for a cohesive Renaissance Computing curriculum, (3) initiate a pilot program to put these specifications into practice, and (4) study the responses of students and faculty to these courses and their attendant ideas. We will also galvanize institutional support for our broader vision and develop a highly-competitive Transformative Implementation (TI) proposal for submission to NSF CPATH.

2. Intellectual Basis/ Related Work

2.1. How Renaissance Computing Benefits CS Majors

CS majors will have the opportunity to apply computational thinking skills to a plethora of disciplines that are traditionally close to CS such as sciences and engineering (Doom *et al.* 2003; Sung *et al.* 2003), as well as those that have already become prevalent, such as arts and humanities. CS majors will be motivated to understand the challenges in those disciplines, and in turn, be "forced" to truly drill down on problem solving using their computational thinking skills in order to find a good solution. Interdisciplinary problem solving and teamwork also allow CS to bring in students of diverse backgrounds and interests, helping our CS majors to better prepare for the real world (Klawe and Shneiderman 2005).

Further, we argue that exposing CS majors to topics in digital arts and humanities also trains them on another front. Creativity and imagination are highly valued in arts and humanities. Unlike science or engineering-oriented applications that usually adhere to strict specifications, problems in digital arts and humanities are much more open-ended and at times with moving targets. Therefore, having interdisciplinary courses such as those proposed in Section 3 will further train our CS majors with their creative thinking skills.

Finally, as the world becomes increasingly digital, CS graduates will likely encounter interdisciplinary projects in sciences, engineering, arts, and humanities in their careers. With our emphasis in interdisciplinary problem solving and collaborative learning, Renaissance Computing will better prepare our CS graduates with the skills they need to succeed in industry. Take for example the areas such as bioinformatics and computational biology, which are now providing a huge playing ground for computer scientists, where they can apply their skills to manage and exploit data. It also provides challenges to computer scientists where new algorithms need to be developed. However, collaboration between computer scientists and biologists are often hampered due to the lack of common language and background knowledge. Interdisciplinary training of CS students in early stages of their education would facilitate the removal of such barriers and broaden their career paths.

2.2. How Renaissance Computing Benefits Other Disciplines

Understanding how the proposed Renaissance Computing program can benefit other disciplines is critical. First, this understanding will allow us to build meaningful and engaging real-world applications that provide interesting and yet challenging assignments and discussions for CS majors. Second, this understanding will allow us to identify the specific computational thinking and computing skills that non-CS majors need, which in turn help us motivate them to take more CS courses and to minor in CS. Here, we use two illustrations, in sciences and humanities.

Sciences. We will use biology as an example to illustrate how Renaissance Computing can benefit students in sciences. Recent advances in molecular biology techniques have changed the research landscape in biology. Instead of examining each gene and protein one by one, we now study the entire set of genes and proteins in an organism and groups, and try to understand their relationships and their roles in larger systems. Such genome-wide or systems-level research requires us to process a large amount of information both efficiently and accurately. Naturally, computational work is integrated into modern biology labs as one of the routine tasks. However, how biologists use computational methods is often limited as they usually try to follow provided recipes found in manuals or cookbook-style textbooks. This happens mainly because the majority of biologists learn how to use computational methods by themselves out of necessity at a later stage of their career, but not as part of their biology education. Even though many biology educators now realize the necessity of integrating computational techniques as part of their biology curricula (e.g., Honts 2003), it is still largely experimental and not regularly done except for computational biology or bioinformatics programs. One of the main reasons for the lack of such integrated education is the lack of expertise in offering such courses in biological sciences. Interdisciplinary collaborative education solves this problem. This will be a win-win situation on both sides. Biology students will receive education in computational thinking that will benefit not only the bioinformatics/computational biology fields but also any life science research that requires logical thinking (e.g., Bialek and Botstein, 2004; Pevzner, 2004; Rice et al. 2004; Wing, 2006) (this is actually true for any life science as well as many other disciplines).

Humanities. It is now uncontroversial to say that over the next century, the entire human record will be put into digital form. This epochal transformation of the traditional objects of humanistic study has, indeed, already begun. Today's literary scholars, historians, linguists, philosophers, and their students have access to vast text and image resources reflecting the primary objects of study in their respective fields. It is obvious that these kinds of digital resources —distinguished not merely by their comprehensiveness, but by their tractability—allow one to ask questions that have literally never been asked before, and so digital scholars in the humanities are increasingly turning toward analysis of this material, which involves subjects that are quite solidly in the realm of computer science and engineering—including data mining, ontology-based storage, agent-based simulation, statistical computing, and natural language processing. Yet despite this abundance and these opportunities, most humanities students remain unequipped to deal effectively with these materials beyond using them as finding aids. Few students have the skills necessary for undertaking comprehensive analysis of these corpora as corpora, by using them to form new kinds of questions and create new kinds of tools. And to make matters worse, their inability to do so means that the potentiality of these resources—the new tools, interfaces, and modes of critical thinking they suggest—will undoubtedly be left in the hands of engineers who, understandably, may have no domain knowledge of the material in question. Our program endeavors to create specialist engineers that can be a part of the digital revolution that has already begun in the humanities, and we believe we can form that community from both sides—by giving engineers a sense of the problems unique to the humanities, and by giving humanists a deeper sense of the computational. The goal is to educate a new generation of students who are adept not merely at using tools, but at creating them and understanding the nature and implication of that creation.

2.3. Why Collaborative Learning

"Collaborative learning" is learning stemming from collaboration that involves the construction of a solution that otherwise could not be produced (Hansen et al. 1999). In Renaissance Computing, we emphasize collaborative learning to yield even more significant impact in student learning of their respective discipline-specific topics and of their interdisciplinary problem solving and teamwork skills. This is especially true as students of diverse backgrounds, interests, and disciplines will encounter conflicts during the joint construction of understanding process. These conflicts could lead to improved learning (Howe et al. 1992).

Increasingly popular is the use of computer-supported collaborative learning (CSCL) systems (Lehtinen et al. 2001) to increase the likelihood of improved learning in participants through more guided or structured collaborative learning, fueled by the advent of Web 2.0 and the renewed opportunities for innovations in teaching and learning (Alexander 2006) and that today's students learn in groups and in a social setting more often, and they solve problems from gathering information online more often than students of yesterday who more likely solved problems individually from information they memorized (Hartman et al. 2007). Research has shown that CSCL can provide re-organized social contexts that promote active and on-task learning, especially when increasingly more instruction and learning are performed online (MacDonald 2003). For example, Schellens & Valcke (2005) found that interaction in the discussion is very task-oriented, stays task-oriented and reflects high phases in knowledge construction. CSCL can also potentially break the isolation of distance learners, and quantitative and quantitative studies of collaborative learning have shown higher levels of learner satisfaction, improvements in knowledge, self-awareness, understanding of concepts, and achievements of course objectives in medicine (Ruiz *et al.* 2006).

However, there are reported pitfalls, such as taking social interaction for granted and restricting social interaction to cognitive processes with these CSCL systems (Kreijns et al. 2003) and the risk of overscripting CSCL systems such that they disturb the natural collaborative interactions and problem solving processes and increase cognitive load on the use of the CSCL itself (Dillenbourg 2002). Thus, for our Renaissance Computing paradigm, the primary use of CSCL is to support and *not* to replace actual collaborative activities, to improve student learning of their discipline-specific topics, and of teamwork and interdisciplinary problem solving skills, and to systematically track and support classroom management across different courses.

3. Implementation Plan

3.1. The Basic Renaissance Computing Framework

Our basic framework of the proposed Renaissance Computing curriculum is one that revolutionizes undergraduate computing education for CS majors and minors, emphasizing interdisciplinary course contents for introductory, middle, and capstone CS courses, and collaborative learning activities. Figure 1 shows the framework for CS majors, CS minors, and non-CS majors. It includes a set of CS1 courses and a unifying CS2 course, sets of technical electives (so-called CS depth courses), and a culminating, integrative capstone course. This design also provides us with longitudinal stewardship over the evolution of these courses, including tracking and assessment of student performances and instructional content. The framework is flexible: it accommodates regular CS majors, as well as CS minors from other departments, and fulfils other majors' technical elective requirements. The capstone course also allows students of other disciplines to collaborate with our CS majors and minors in interdisciplinary group projects, further exposing the students to exciting aspects of solving interdisciplinary problems. Finally, the proposed framework is customizable by individual students, providing different pathways to suit student needs and interests, either as a CS major or a CS minor.

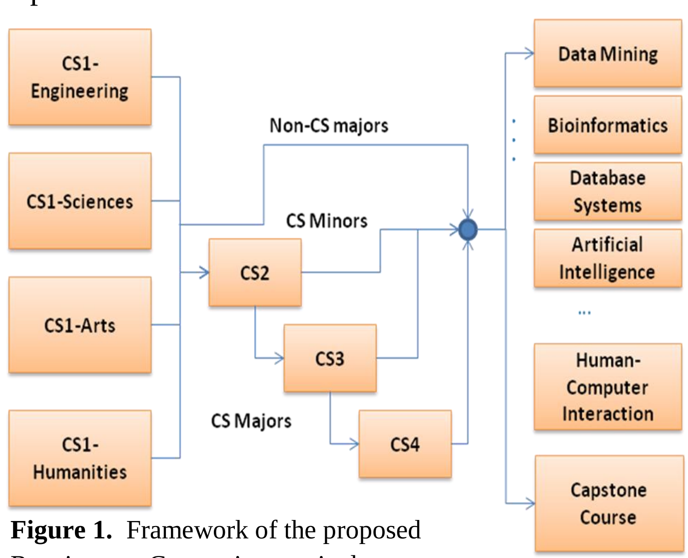

Figure 1. Framework of the proposed Renaissance Computing curriculum

The framework will incorporate collaborative learning activities with interdisciplinary projects or assignments for groups of students from different courses, allowing students of different CS1 courses to interact, and students of different levels (from freshmen to seniors) to interact. The collaborative learning environment will support student social networking and facilitate teamwork using a computer-supported collaborative learning (CSCL) system called I-MINDS. The system will also help teachers monitor and manage student groups.

3.1.1. The Renaissance Computing CS Courses.

1. **The CS1 "Funnel":** The proposed framework has a set of CS1 courses, each tied to a different non-CS area. Each course may use a different programming language and have a unique emphasis. For example, CS1-Engineering may use C or C++ as the programming language and with assignments and examples in engineering applications, while CS1-Arts may use Python or Java as the programming language, and with assignments and examples in multimedia applications pertaining to, say, music, dance, and digital arts. CS1-Sciences will be relevant to life sciences (e.g., physics, chemistry, and biology) and natural resource sciences (e.g., geosciences and hydrology); CS1-Humanities will be relevant to history, English, and so forth. Each of these CS1 courses will contain the same basic core of CS topics (a subset of those in IEEE/ACM Computing Curricula 2001), including the problem-solving paradigm of object-oriented programming, at least covering the concepts of abstraction, encapsulation, and the "object" view when constructing a solution. Each CS1 course will also introduce students to database design. CS1 will be required of all CS majors and minors.

2. **CS2**: The framework has one CS2 course, to cover data structures (e.g., stacks, queues, linked lists), other traditional CS2 search and sorting algorithms, and the concepts of object-oriented programming further: polymorphism and inheritance. Further, OO concepts such as event-driven programming and exception handling will be covered. CS2 will be required of all CS majors and minors.

3. **Depth Courses**: In addition to the set of depth courses already being offered (e.g., Bioinformatics, Data Mining, Artificial Intelligence, Database Systems, and Human-Computer Interaction), new topics such as simulation, computer visualization, embedded systems, and autonomic computing will be considered. These depth courses will serve as technical electives for CS majors and minors.

4. **Capstone Course**: The proposed framework requires all CS majors and minors to take a capstone, project-based course. All projects will be interdisciplinary in terms of the problem to be solved and also the team members. This capstone course will also meet an elective requirement of all majors of participating departments.

3.1.2. Discipline Non-CS Majors. We envision the CS1 "Funnel" courses as establishing a foundation for broader implementation of computational thinking within the academic disciplines. Within the Renaissance Computing Framework in Figure 1, we anticipate that these courses will ultimately be required of all students in their respective major fields. We also anticipate that the capstone course will become, if not required, at least a broadly-taken senior elective. As a result, Renaissance Computing will become an integral component of the University's general core academic requirements. While all students will take CS1 and most capstone, we anticipate that many of these students will take additional CS2 and depth courses to gain additional computational knowledge and skills to apply in their field, with perhaps many pursuing a CS minor. One key objective of this planning grant is to develop the initial content for the CS1 courses. We will engage in planning efforts to develop the topic areas and curriculum for the CS1 courses both in terms of disciplinary content and for expanding the basic CS1 core for CS students. We will also determine the *feasibility of requiring CS1 for **all** students of the participating academic units regardless of majors and minors*, based on our Pilot Study, workshops, and meetings with the students, faculty, and administrators.

3.1.3. Technology-Based Learning Platforms. We see two specific technology-based learning platforms as critical to the implementation of Renaissance Computing courses and the incorporation of collaborative learning activities. These technology platforms could facilitate course delivery at UNL and provide potentially cost-effective ways to disseminate and replicate the Renaissance Computing approach at other institutions.

Learning Objects. Learning objects (LOs) are small, stand-alone, mediated, content "chunks" that can be reused in multiple instructional contexts and serve as building blocks for lessons. The value of the learning object approach has been recommended by the Department of Defense (ADL 2003), business and industry (ASTD 2000), public schools (Grunwald 2002; Nugent 2005; Pasnik & Keisch 2003), and higher education (Koppi & Lavitt 2004; Francia 2003; van Zele et al. 2003). Major strengths cited were reusability, ease of updates and content management, customization, interoperability, and overall flexibility. Research on LOs has also verified their instructional value (Boster et al. 2002; Bradley & Boyle 2003; Nugent et al. 2006a).

PI Soh and Co-PI Samal are leading an NSF-funded project called the Intelligent Learning Object Guide (iLOG). The long-range goal is to augment LOs with empirical usage intelligence—how a learning object should be used, how it has been used, and how it has impacted instruction and learning. The project is currently designing and producing learning objects—which are compliant with the Shareable Content Object Reference Model (SCORM) standard (ADL 2004)—on CS topics that uses active learning (Bransford and Schwarts 1999) and feedback. In contrast to passively listening to a lecture, active learning requires students to dynamically make decisions and choices, which, in turn, influence the sequencing and instructional presentation. Feedback is another critical design dimension. Studies of learning, transfer, and development show that feedback is extremely important and that, usually, it should be immediate (Bransford et al. 2000; Mory 2004). The value of LOs as an instructional tool has been recognized by the National Science Foundation (NSF) as, in addition to our iLOG project, the NSF has funded learning objects-oriented projects (Gehringer 2006; Wiley 2002; Steuck 2002; Recker and Dorward 2000).

Because the CS1 "Funnel" courses share common CS core components, LOs would be a cost-effective and efficient way to share this common core curriculum across the courses. As part of this proposal, we will create LOs for *selected common CS core components* that will be used across the courses to test the feasibility of this approach. We will also create LOs for *selected discipline-specific content* to determine student reactions to the LO approach for curriculum within their discipline. The flexibility of LOs makes them a potentially viable approach to supplement the delivery of Renaissance Computing courses and the ultimate dissemination of CS1 courses to other institutions. As indicated in the Budget Justification, LOs created from the iLOG project will be adapted to create the above LOs.

Computer-Supported Collaborative Learning: I-MINDS. Collaborative learning is a foundational principle of Renaissance Computing. To support collaborative learning, across the courses as well as the disciplines, we will utilize a computer-supported collaborative learning (CSCL) tool: the Intelligent Multiagent Infrastructure for Distributed Systems in Education (I-MINDS) (Soh et al., to appear). I-MINDS was developed by PI Soh and his research group at

UNL, and has been deployed at UNL and Bellevue University (the second largest online university in the United States). I-MINDS employs a system of intelligent software agents, representing individual students and the instructor (or teaching resource in the case of an asynchronous course or lesson). Briefly, an *agent* is a software module that observes and receives input stimuli from its environment, makes autonomous decisions based on these stimuli, and actuates actions to carry out these decisions, which, in turn, change the environment (Weiss 1999). A multiagent system, in turn, consists of agents working together to solve joint tasks or achieve global goals. In I-MINDS, each student agent serves a student, profiling the student's behavior in his or her structured and non-structured collaborative activities (Khandaker et al. 2006). The student agents exchange information to form peer groups that are compatible to help students collaborate (Soh et al. 2006). For each group, there is a group agent that provides scaffolding and monitors the progress on accomplishing tasks assigned by the instructor. The instructor is supported by a teacher agent that displays statistics of students and groups, identifies problems in group activities (for example, a student being too dominating or too shy), and manages the Q&A sessions by ranking questions posed by the students. The teacher agent also administers quizzes and supports multithreaded forum discussions. Our educational studies show that agent-mediation can help improve student performance (Soh et al. 2004) and form more effective and efficient groups (Khandaker et al. 2006; Soh et al. 2006).

For this proposal, we will examine how to incorporate collaborative learning into our CS1 "Funnel" courses so that students of various disciplines could interact without having to be in the same class and students of different levels (from freshmen to seniors, and CS majors to minors) can interact. To promote teamwork and collaborative learning, I-MINDS also provides a unique function: it is one of only a handful computer-supported collaborative learning (CSCL) systems that tracks and records all student activities among themselves and with the teacher through its graphical user interface. I-MINDS enables instructors to track and capture student participation in collaborative activities, allowing, for example, researchers to study student pedagogy of students of diverse backgrounds. Since student participation is highly accountable through I-MINDS, instructors would be able to better motivate students to collaborate, which is important for our Renaissance Computing framework.

3.2. Specific Activities and Roles

Table 1 lists the interdisciplinary team of PIs involved in this project. There are faculty from sciences (Moriyama and Meyer), computer science and engineering (Soh, Samal, Scott), arts (Moore) and humanities (Ramsay and Thomas), and education psychology (Shell). Most of the PIs have been involved in curricular activities or interdisciplinary research.

3.2.1. Year 1: Course Planning and Development Phase

We will initiate planning activities with a one-day "Renaissance Computing: Curriculum Planning" (RC-CP) Workshop. The objective of this workshop will be to develop a planning framework for identifying course content. In addition to the project PIs, other faculty from Computer Science and the participating disciplines will be invited. Professor Kenneth Goldman (PI of an NSF CPATH TI project, from Washington University in St. Louis) has tentatively agreed to attend the workshop to provide guidance on what Transformational Projects entail and suggests for planning for transformation. At the workshop, work groups will be formed for each of the disciplines: sciences, engineering, arts, and humanities, with each led by PIs of corresponding disciplines (Moriyama, Meyer, Moore, Ramsay, and Thomas). The CS PIs (Soh, Samal, and Scott) will assist each work group. At the workshop, initial planning discussions in the work groups will focus on three specific threads: (1) aspects of computational thinking that expand the problem solving skills of students in non-CS majors, (b) aspects of interdisciplinary problem solving that better engage potential CS majors and prepare CS majors for a career in computing, and (c) aspects of collaborative learning that affect student motivation, aptitudes, and attitudes towards working with students in other disciplines.

Name	**Affiliations**
Leen-Kiat Soh, PI	Associate Professor, Chair of Curriculum Committee, Department of Computer Science and Engineering, College of Arts and Sciences (CAS), and College of Engineering (CoE); member of College of Engineering's Curriculum Committee
Ashok Samal, Co-PI	Associate Professor, Member and former chair of Curriculum Committee, Department of Computer Science and Engineering, CAS and CoE
Stephen Scott, Co-PI	Vice Chair and Associate Professor, Member and former chair of Curriculum Committee, Department of Computer Science and Engineering, CAS and CoE
Etsuko Moriyama, Co-PI	Associate Professor, School of Biological Sciences, CAS
Steve Ramsay, Co-PI	Assistant Professor, Department of English, CAS; Member of the Undergraduate Program and Curriculum Committee of the Department of English.
George Meyer, Co-PI	Professor, Department of Biological Systems Engineering, College of Agricultural Sciences and Natural Resources (CASNR) and Institute of Agriculture and Natural Resources (IANR); Representative of IANR's Computing Steering Committee
Brian Moore, Co-PI	Professor and Chair, Department of Music Education, School of Music, Hixson-Lied College of Fine and Performing Arts
William Thomas, Co-PI	John and Catherine Angle Chair in the Humanities and Professor, Department of History, CAS;
Duane Shell, Co-PI	Research Associate Professor, Department of Education Psychology, College of Education and Human Sciences

Table 1. List of PIs.

Following the RC-CP workshop, each work group will regularly meet to develop the specific content of its respective CS1 course. This includes identification of computational topics, core computational knowledge and tools, interdisciplinary problems and collaborative problem-based learning (PBL) activities, and additional content. Monthly, there will be a meeting of the entire planning group to discuss progress and cross-course issues such as common curriculum and activities and cross-course student collaboration.

At the end of the first six months, we anticipate work groups completing their course planning. At the end of this planning phase, we will have identified specifications to create the specific curricula and materials for each of the CS1 courses. Also, we will have developed specifications for the common core principles that will define a Renaissance Computing course.

We will then shift work group activity to development of the specific curricula and materials for each course. We will fully develop the CS1-Sciences and CS1-Humanities courses for implementation in the pilot study. For these courses, we will complete full implementation of content topics, PBL activities, and collaborative activities. During this phase, we will employ a graduate student to assist with the creation of the specific LOs for these courses. These will be SCORM-compliant to allow maximum portability. We will develop common LOs for the two courses where possible. We set up I-MINDS for collaborative learning activities in the two courses. The CS1-Sciences and CS1-Humanities work groups and the graduate assistance will meet as frequently as they determine to complete full course implementation. Although the engineering and arts CS1 courses will not be implemented during the project, these groups will continue to develop their specific content and activities during this time so that these courses will be ready to implement during the TI grant.

There will be monthly meetings of the entire planning group to assess progress, examine fidelity to the common set of Renaissance Computing principles, and ensure that common components are being implemented consistently. At the end of the first year we will have completed the CS1-Sciences and CS1-Humanities courses for the pilot study.

3.2.2. Year 2: Pilot Study Phase

The CS1-Sciences and CS1-Humanities courses will be offered during the first semester of Year 2, with students recruited primarily from our freshmen CS majors and minors and students from the School of Biological Sciences, the Departments of History and English, and secondarily from other participating academic units. Students participating in these courses and the instructor will be asked to consent to participation in research and evaluation activities in accordance with Institutional Review Board (IRB) guidelines. We will collect data on student outcomes and technical aspects of course implementation. For student outcomes we will use pre- and post-tests to assess students' knowledge and skills in computational thinking and tool use, attitudes and motivation about computational thinking and CS, attitudes and motivation about interdisciplinary collaboration, impact of the course on students' self-regulation, and students' perception of the classroom environment. In addition, we will embed specific assessments within the LOs. These will capture both student learning and immediate student attitudinal and motivational reactions to the LO. At the end of the course, we will conduct a short on-line interview with students to gather their self-assessment of their learning, motivation, and experience in the course.

Feedback on technical aspects of course implementation will be obtained from students and the instructor using an end-of-the-course on-line interview. Students and the instructor will be asked to provide feedback on satisfaction, course effectiveness, and suggestions for course improvements. They will also be asked to rate the utility and interestingness of specific course components. We will also capture data from interaction with the LOs and I-MINDS to identify how students were interacting with these learning technologies. Instructors also will be provided with an on-line diary to record immediate reactions and comments concerning the course.

Details of the data and analysis are provided in the Evaluation Plan (Section 6). Data from the Pilot Study will be summarized and provided to the entire planning team, including all work groups. The CS1-Sciences and CS1-Humanities work groups will meet to examine the results in detail and draft recommendations to the entire planning group. The planning group will then meet to develop final recommendations for CS1 course specifications and implementation.

While the Pilot Study is being conducted during the first semester of Year 2, CS faculty, led by the CS PIs (Soh, Samal, and Scott), will develop specifications and content for the CS2 and depth courses that will be part of the Renaissance Computing framework. They will create work groups for each course. These work groups will develop the specifics of content and materials for these courses. The work groups will examine these courses to align them with Renaissance

Computing principles and create new Renaissance Computing components where relevant. They will examine where aspects of each course could be programmed into LOs and where CSCL activities would be useful. They will examine where interdisciplinary content can be integrated. On a monthly basis, the combined CS workgroups will meet to update progress, examine fidelity to the common set of Renaissance Computing principles, and insure that common components are being implemented consistently. These activities will ensure that the spectrum of Renaissance Computing courses is ready to implement during the TI Grant.

3.2.3. Year 2: Grant Writing Phase

Following the Pilot Study, the project team will focus on four tasks: development of the capstone course, gaining formal institutional commitment to the implementation of Renaissance Computing at UNL, identification of an external evaluation team for our TI proposal, and completion and submission of a highly-competitive TI proposal to NSF CPATH. At this point, we will have specific course specification for the CS1 courses, CS2, and depth courses. We will begin a work group for the capstone course. This work group will examine data from the pilot study of the CS1 courses to gain insight into appropriate interdisciplinary, collaborative principles for the capstone course. They will work during the final semester of the grant to develop the specifications for the capstone course.

Following the pilot test, evaluation data will be compiled into a report for UNL administrative units. We will send the report to department chairs, deans, and upper administration. The report will describe the Renaissance Computing vision and data that support its effectiveness and feasibility. We will supplement these reports with a half-day “Renaissance Computing: Demonstrations” Workshop. We will invite faculty and administrators in the relevant content areas, colleges, and upper administration to a session where Renaissance Computing will be explained and demonstrations of course content, learning objects, and collaborative activities will be presented. We will also present relevant evaluation results. This meeting will be directed at obtaining buy-in and support for the institutional transformation envisioned in the TI Grant. We will follow-up with visits by the PIs to relevant administrators in their respective disciplines. We will also hold informational meetings with faculty in the disciplines. Through these activities, we will be able to spread information about the Renaissance Computing idea across the campus and gain support from faculty and administration for implementation.

The final activity will be to identify a proposal-writing team for the writing of the TI Grant. This will include identification of an external evaluator, as required in the TI Guidelines. The team will work during the final three months of the grant to produce the proposal.

3.2.4. Roles

PI Soh will lead the Renaissance Computing team. Co-PIs Samal and Scott will represent the CSE Department and also serve as CS experts on the team. Soh, Samal, and Scott will be involved in organizing the workshops. Co-PIs Meyer (Sciences) and Moriyama (Sciences), Moore (Arts), Ramsay (Humanities), and Thomas (Humanities) will represent the four key areas of disciplines. They will lead the working groups for the “Renaissance Computing: Course Planning” Workshop. They will also work with the CSE PIs to finalize the set of topics for the interdisciplinary CS courses. Co-PI Shell will lead the Pilot Study. CS PIs will work with Moriyama (CS1-Sciences), Ramsay (CS1-Humanities) to create these two courses, and a pair of PIs will co-teach each course (e.g., Scott and Moriyama for CS1-Sciences, and Soh and Ramsay for CS1-Humanities). All PIs will also be liaisons with their respective academic units in interactions with the students, faculty, and administrators.

3.2.5. Timelines and Milestones

Activities	Year 1				Year 2			
	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4
Course Develop. & Planning Phase								
Workshop Organization	ΔW1							
Discussions & Reporting		ΔR1						
Pilot Study Phase								
Educational Research Design			ΔIRB					
Course Design				ΔCS				
Data Collection (Courses)					Courses offered			
Workshop Organization							ΔW2	
Data Analysis & Evaluation								ΔR2
Grant Writing Phase								ΔP
Project Evaluation				ΔAR1				ΔAR2
Project Dissemination								

* The milestones are: ΔW1 = 1-day Renaissance Computing: Course Planning (RC-CP) Workshop, ΔR1 = report on topic identification, ΔIRB = the educational research design and with IRB-approval, ΔCS = contents ready for the CS1-Sciences and CS1-Humanities courses, ΔW2 = half-day Renaissance Computing: Demonstrations (RC-D) Workshop, ΔR2 = report on the pilot study, ΔP = TI proposal to NSF CPATH, ΔAR1 and ΔAR2 = annual reports to the NSF.

4. Project Expertise

At UNL, the PIs of this proposal have been involved in curricular initiatives of various scales, with most focusing on individual disciplines. Here we list several significant initiatives and programs that the PIs or UNL have undertaken or have been involved in and how they relate to the proposed Renaissance Computing. More importantly, these activities indicate the readiness—in terms of experience, expertise, and collaborative relationships—of UNL and this team to embrace such a radical change to our approach to undergraduate computing education as the proposed Renaissance Computing.

The Reinventing CS Curriculum Project. In 2002, the University of Nebraska-Lincoln (UNL) Department of Computer Science and Engineering (CSE), in collaboration with the College of Education and Human Sciences (CEHS) and under the auspices of the National Center for Information Technology in Education (NCITE), undertook a major effort in redesigning its undergraduate curriculum, including aligning it to the ACM/IEEE-CS Computing Curricula (2001). This effort was motivated by the need to keep pace with a growing and rapidly-changing field and to offer an effective learning experience to attract and retain students. Called the *Reinventing Computer Science Project*, the goal was to better prepare students for the dynamic and challenging workplace in the computer science (CS) and information technology field. Funded from a U.S. Department of Education grant, this project resulted in a number of major outcomes: (a) introductory CS course elements that include a placement test (Nugent et al. 2006a), interactive multimedia LOs (Nugent et al. 2006b), and active-learning laboratories (Lang et al. 2006, Soh et al. 2005), (b) a framework to conduct systematic research on instructional approaches and their effectiveness in both the short term and longitudinally (Samal et al. 2005; Soh et al. 2007), and (c) a strong collaboration of an interdisciplinary group of researchers and educators with experience and insights to improve CS education. We followed the cyclic model for knowledge production and improvement of practice (Ball, 2003) where research-based learning material (LOs) and strategies were systematically developed and carefully evaluated, leading to new insights and research questions. Courseware (e.g., LOs, laboratory assignments, and homework assignments) produced by this project will serve as templates for the new courses in Renaissance Computing and as "idea blueprints" for faculty of other disciplines. Further, the CS1 framework, educational research design, insights, experience, and strong collaboration will be leveraged for the concept development and planning of Renaissance Computing.

Digital Humanities. UNL has made a substantial commitment to Renaissance Computing by hiring humanities scholars who specialize in the creation of digital tools. We are undoubtedly one of the few institutions in the country that offers advanced courses in programming, software engineering, and digital archive creation to English and History majors that are taught not by computer scientists, but by humanities scholars who specialize in these areas. We are likewise one of the few institutions in which computer scientists regularly collaborate with humanities scholars on research questions germane to both disciplines. It would be difficult to understate the importance of this particular aspect of our local culture, which, in our experience, is exceedingly rare.

***J.D. Edwards Honors Program*.** The J. D. Edwards Honors program is a joint, interdisciplinary program between the Department of Computer Science and Engineering (CSE) and the College of Business Administration (CBA), born out of a multi-million dollar endowment. The program focuses on developing leaders in interdisciplinary CS and business management education for high ability students. Students are recruited from around the country and provided room and board for four years. Regardless of their majors, students are required to take more than 24 credit hours of CS courses. A significant number of the core CS courses have been adapted to include flavors of business-oriented applications, such as a finance calculator and a transactions account tracker. The students also must register in two years of Senior Design Studios (akin to capstone courses) to work in interdisciplinary teams on real-world applications—actual projects paid for by IT companies, from problem specification to solution proposal, from design to implementation, from customer satisfaction to product evaluation. These studios allow the students from CS and CBA to learn invaluable interdisciplinary teamwork skills, and are extremely engaging. While the model of the J. D. Edwards Honors program might not be financially feasible for the general UNL campus, key aspects such as curricular design, instructional content, and course delivery that have taken place at the program provide critical insights to our proposed activities. For example, PI Soh has been involved in developing some of the interdisciplinary CS core courses, and is presently designing a new course, "Artificial Intelligence Applications," for the honors program, focusing on AI fundamentals and their use in business applications.

Achievement-Centered Education (ACE). UNL has recently adopted an "Achievement-Centered Education" (ACE) program as part of a campus-wide general education reform initiative. At the heart of this reform is a desire for an approach to general education that is elegant and simple, based on student outcomes, and integrates (and is reinforced within) all undergraduate majors. The ACE program has been accepted by the entire UNL campus and asks ***all*** students, regardless of major, to demonstrate specific skills and abilities, including reasoning, inquiry and civic capacities. Four institutional objectives are: (1) Develop intellectual and practical **skills**, including proficiency in written, oral, and visual communication, inquiry techniques, critical and creative thinking, quantitative applications, information assessment, teamwork, and problem-solving; (2) Build **knowledge** of diverse peoples and cultures and of the natural and physical world through the study of mathematics, sciences and technologies, histories, humanities, arts, social sciences, and human diversity; (3) Exercise individual and social **responsibilities** through the study of ethical principles and reasoning, application of civic knowledge, interaction with diverse cultures, and engagement with global issues; and (4) **Integrate** these abilities and capacities, adapting them to new settings, questions, and responsibilities. Our proposed Renaissance Computing aligns with the ACE initiative and provides a concrete example of the integration of diverse disciplines in support of one another.

Graduate Programs in Bioinformatics. The CSE Department at UNL has an MS and a PhD program with a Bioinformatics Specialization. Its objective is to provide qualified candidates with the opportunity to pursue a course of study that will bring them to the frontiers of knowledge in an area of bioinformatics—an interdisciplinary research area merging CS and Biology. Co-PIs Scott and Moriyama have been instrumental in creating this program, have co-advised CS students (both at the undergraduate and graduate levels) for bioinformatics projects, and have produced five M.S. graduates and one Ph.D. student (soon to defend his dissertation), and has fostered tighter collaboration between CSE and the School of Biological Sciences. Moriyama was also instrumental in developing and installing the **Interdisciplinary Bioinformatics Specialization** at UNL. This interdisciplinary specialization (both at the MS and PhD levels) encompasses 12 departments, and CSE's specialization in Bioinformatics specialization has been incorporated as one of the two tracks (the second track being for the Life Sciences). Further, Scott and Moriyama have independently developed and have been teaching bioinformatics courses for CSE department and School of Biological Sciences, respectively, for the past five years. Both courses are taken by both senior-level undergraduate as well as graduate students, and attracting students outside of their own disciplines.

5. Institutional/Community Support

Presently, we have institutional support from the CSE Department, the English Department, the History Department, the School of Biological Sciences, the School of Music, and the College of Agricultural Sciences and Natural Resources to support the various activities proposed for this concept development and planning project. Please see the letters of support in the Supplementary Documents section.

6. Evaluation Plan

Evaluation will consist of both formative and summative components. Co-PI Shell will identify and create appropriate evaluation instruments and finalize research designs and analytic techniques for the summative evaluation during the summer prior to the pilot study after course specifications have been finalized. Formative and summative data will be compiled into the Annual Reports to NSF CPATH. Summative data on outputs and student outcomes will be compiled into a written report for dissemination to administration and faculty at UNL. Evaluation data will be reported in professional presentations and publications.

Formative evaluation will examine project progress in meeting objectives and completing planned activities in relation to grant timelines. Data sources will include attendance logs and summary minutes from the Workshop and work group meetings, with a specific focus on documenting broad participation both from the disciplines and from CS faculty. Attendance logs and minutes will be reviewed by project PIs to ensure that work groups are meeting with adequate frequency and that planning and development activities are being completed at a suitable pace to realize project outcomes. Pilot study participation will be documented from course enrollment. Summary minutes of work group meetings will be used to capture *process documentation* on issues, barriers, controversies, and problem solving used to realize the final course content structure. Working papers and drafts of course components along with minutes will be archived to create a "documentary" of planning activities.

Summative evaluation will determine achievement of project outputs and outcomes. For Course Planning and Development, realization of anticipated outputs will be documented by final completion of the courses. The course curricula for the CS1 courses will document the computational topics, core computational knowledge and computing tools, disciplinary problems, PBL activities, and ancillary knowledge and skills present in each course. Learning Objects and CSCL activities will be documented by their completion and implementation in the courses. Course curricula will also document the Renaissance Computing changes to topics, knowledge, and activities in the CS2 and depth courses and the final structure of the capstone course. Documentation of institutional support will be provided by administrative commitment to the TI Grant, which will itself serve as documentation of project success.

For the pilot study, we will examine student outcomes using a pre-test/post-test design. At the start of the class, students will complete an assessment battery consisting of a knowledge and skills test, attitude and motivation scales, self-regulation instruments, and classroom environment instruments. Students will complete the assessment battery again at the end of the semester. Within-class changes will be analyzed by appropriate repeated measures analyses such as dependent *t*-test and Chi-Square. The pilot classes will be compared for attitudes, motivation, self-regulation, and classroom environment using appropriate between-group analyses such as *t*-test and Chi-Square. Students will complete the on-line interview at the end of the semester. Student on-line interview responses will be summarized, grouped and content coded.

For technical data, students and instructors will complete an on-line interview at the end of the semester. Students' satisfaction, course effectiveness, and suggestions for course improvement responses will be summarized, grouped, and coded. Instructor responses will be coded. Descriptive statistics of student and instructor ratings of the utility and interestingness of specific course components will be compiled. Student interaction data captured from LOs and I-MINDS will be compiled and analyzed for emergent patterns of use within and across LOs and collaborative interactions.

7. Dissemination Plan

The targets of our dissemination plan include parties on campus (administrators, faculty, and students), colleagues at CPATH, and a larger audience of CS educators at national and international conferences. For parties on campus, through our two proposed workshops, we expect to share information about Renaissance Computing and the results of our Pilot Study with the UNL campus. Material (e.g. learning objects and self-studies) in electronic forms will be disseminated, and a dedicated website will be set up, modeled after our current Reinventing CS Curriculum Project website. Keeping in touch with existing CPATH grantees (for example, PI Professor Goldman has tentatively agreed to be one of the two invited speakers at our first workshop) and attending the annual CPATH PI meetings will allow us to share our experiences and results of the Course Development & Planning and the Pilot Study phases. Finally, we plan to publish our results at SIGCSE, the premier annual conference on CS education.

8. Prior NSF Results

NSF Advanced Learning Technology (ALT): "Embedding and Validating Empirical Usage Intelligence in Learning Objects," Contract number: IIS-0632642, 09/15/07-07/31/09. The project started last year and has progressed steadily. We developed a comprehensive tracking mechanism that wraps around learning objects to capture student interactions. The mechanism works with standard learning management systems such as Blackboard and SCORM-compliant learning objects. We have also developed a suite of templates for creating animated components in learning objects. Finally, we have produced three learning objects and are currently conducting our educational research studies with students in introductory CS courses.

9. Intellectual Merit

The intellectual merit of this concept development and planning proposal is twofold. First, the Renaissance Computing framework is holistic: it covers introductory, depth, and capstone courses, and targets both CS majors and minors. Further, it is aimed to develop interdisciplinary courses in sciences and engineering (which are traditionally aligned with CS) and arts and humanities (which are gaining significance as key applications areas for CS). In addition, the framework embraces collaborative learning, which has been shown to improve learning performance. Finally, the framework recognizes the flexibility of learning objects to supplement course delivery, affording students the luxury of perusing modules of other courses of other flavors. Second, the time is ripe for this highly-motivated group to conduct the concept development and planning phase *now*. The PIs have had significant experience and involvement in curricular changes and degree programs and in teaching introductory CS courses and other programming courses on campus. With the ACE initiative, degree programs in bioinformatics, and initiatives in digital arts and humanities, there is a gathering momentum on the UNL campus providing opportunities that the Renaissance Computing will be able to seize to affect the overall academic planning and revolution on UNL in the near future. However, most of the recent initiatives (except for ACE) have been to expand the interdisciplinary research or graduate programs. Therefore, it is imperative that this Renaissance Computing project to take a catalyst and leadership role at this juncture to help guide the transformation of the overall undergraduate education at UNL, particularly, the revitalization of undergraduate computing education at UNL.

10. Broader Impacts

The broader impacts of the proposed project are threefold. First, as a result of the concept development and planning, we expect that through our workshop, more academic units on campus will be inspired and join our effort, giving the Renaissance Computing framework a stronger foothold at UNL. We expect that active role will affect how UNL's ACE program is to be implemented, how we recruit students to major and minor in CS, how we promote interdisciplinary research programs at the graduate levels, and how we extend our curricula to include more contacts with the industry. We also expect computer-supported collaborative learning systems such as I-MINDS to open new doors to new and creative collaborative activities because of the technology's ability to produce high-resolution accounting of student participation, support teamwork, and manage classrooms. Second, the results of our Renaissance Computing such as completed programming assignments (e.g., software that are visualizations, graphical user interfaces, and simulations) and learning objects will be disseminated to middle and high schools to recruit students into CS and STEM areas, as well as digital arts and humanities. As these courseware products are modular and standalone, they can be conveniently incorporated as 1-day lessons or online activities by school teachers. Third, the percentages of female students in sciences, arts, and humanities are significantly higher than those in engineering, particularly CS. With Renaissance Computing, we will expose computational thinking to reach a large portion of female students, and will increase female participation in undergraduate computing.