1 Vision, Goals, Objectives, and Outcomes

The goal of this project is to build a consensus across disciplines toward incorporating computational thinking in DePaul University's Liberal Studies curriculum. The work done in the project will enable similar changes in general education programs at academic institutions nationally.

Expanding computing across the higher education curriculum and the training of a computing-savvy workforce has been recognized by the NSF as crucial to the nation's continued prosperity and security. In this project we focus on using liberal studies courses – part of the education of the vast majority of undergraduates – as a vehicle for the teaching of computational thinking. In order to integrate computational thinking across the liberal studies curriculum, a broad consensus among faculty from diverse areas must be reached. In this project we will work toward achieving this consensus by developing a framework that faculty without formal training in information technology can use to understand and integrate computational thinking into their liberal studies courses. We intend, after the completion of this two-year project, to use the framework to integrate computational thinking into DePaul's Liberal Studies program and to disseminate the framework so that other institutions can begin the same process.

To integrate computational thinking into liberal studies courses across several disciplines, we propose in this project to:

- Make computational thinking explicit in a selection of liberal studies courses
- Develop assessment tools that evaluate the learning of computational thinking
- Use this information to develop a framework that can be used by instructors outside of information technology to understand and integrate computational thinking into their own courses, at DePaul University and elsewhere

The framework we will develop will will include a set of principles for identifying computational thinking, examples of computational thinking in various disciplines, and best practices for teaching computational thinking in liberal arts courses. This framework will demonstrate how computational thinking can become a part of the general education curriculum at any college or university. This framework will be peer reviewed, tested, and assessed.

2 Intellectual basis/related work

Expanding computing across the higher education curriculum has been recognized by many organizations as crucial to the nation's continued prosperity and security. Several studies, including one by the Computer Science and Telecommunications Board (CSTB) of the National Research Council (NRC) [1] and one by a focus group of the Computing Curricula 2001 Computer Science (CC 2001 CS) Task Force [2], have developed guidelines on how to improve information technology (IT) skills and fluency. One example of the implementation of these guidelines in higher education is DePaul University's two-quarter course sequence, "Math and Technology Literacy I&II", which is an integral part of DePaul University's Liberal Studies program and is required of all non-computing majors.

Neither the NRC report nor the CC 2001 CS volume, however, explicitly recognize that computer science offers certain analytical concepts and tools that go beyond IT fluency. In a CACM article[3] Wing argues that computer science has developed a set of "computational thinking" skills that have direct application beyond computer science. She argues that the ideas of abstraction, layering of abstractions, and automation – to name a few – are fundamental computer science concepts that have already yielded new insights in the natural sciences and hard social sciences such as economics. Wing gives specific examples of computational thinking in those fields. She argues that computational thinking is an emerging basic skill that should become an integral part of education together with reading, writing, critical thinking, and problem solving.

This project will start implementing Wing's vision by integrating computational thinking into courses in various areas of the liberal arts curriculum. We will make computational thinking explicit in a number of Liberal Studies courses taught by technically-oriented faculty, develop tools that evaluate the learning of computational thinking, and create a framework that can be used by non-technical instructors to integrate computational thinking in their own courses, at DePaul University and elsewhere.

DePaul University and its College of Computing and Digital Media (CDM) are in a unique position to take on this project. Several years ago, the CDM faculty decided that our teaching responsibilities should not be limited to the traditional technical IT areas or the teaching of IT fluency. As members of the DePaul faculty, we understood that we must participate in the teaching of core skills of reading, writing, problem solving, and critical thinking in the context of the Liberal Studies program. As a liberal arts university, DePaul places a strong emphasis on general education, and students are required to take approximately half of the credits required for graduation in the Liberal Studies Program. The Liberal Studies Program has a common core of classes that are required of (nearly) all DePaul students as well as courses in specific domains, in particular, Arts and Literature, Philosophical Inquiry, Religious Dimensions, Scientific Inquiry, Self, Society, and the Modern World, and Understanding the Past. A large number of DePaul students who do not major in computing take computing courses as a part of their general education.

The De Paul CDM faculty also understood that the broader field of computing is rich in content for courses in diverse liberal arts areas. The CDM faculty has, over the last 5 years,

[1] *Being Fluent in Information Technology*, Computer Science and Telecommunications Board, National Research Council, National Academy Press, 1999

[2] *Computing Curricula 2001: Computer Science*, IEEE Computer Society and the ACM

[3] *Computational Thinking*, Communications of the ACM, Vol. 49, No. 3, March 2006

developed a wide range of liberal arts courses that go beyond the traditional courses in fluency in IT, computer science and programming for non-majors, or specialized courses for certain majors. These courses were developed in partnership with many academic units and administrative bodies outside of CDM, and the content and the goals of the courses were thoroughly examined and vetted by the faculty who traditionally teach those courses. We list a sample of these courses, categorized by Liberal Studies domain area:

- Arts and Literature

  GAM 224: Introduction to Game Design

  ANI 101: Animation for Non-Majors

- Scientific Inquiry

  CSC 233: Codes and Ciphers

  CSC 235: Problem Solving

  CSC 239: Personal Computing

  ECT 250: Internet, Commerce, and Society

  HCI 201: Multimedia and the World Wide Web

  IT 130: The Internet and the Web

- Self, Society, and the Modern World

  DC 105: Digital Media Literacies

  CSC 223: The Impact of Computing Technology On Our Lives

- Understanding the Past: Intercontinental/Comparative

  GAM 206: History of Games

  GPH 205: Historical Foundations of Visual Technology

- Philosophical Inquiry

  CSC 208: The Computer and Social Responsibility

  IT 228: Ethics in Computer Games and Cinema

While primarily developed to teach liberal arts concepts and skills, these and other courses developed by CDM faculty always included some elements of computational thinking. The starting point for our project will be to study the skills taught in these courses and make the computational thinking explicit.

In order to categorize the computational thinking skills we uncover, we will use a framework developed by Denning in his "Great Principles of Computing" project[4]. The project's goal is

[4] *Great Principles of Computing*, http://cs.gmu.edu/cne/pjd/GP/GP-site/welcome.html

to articulate the fundamental principles of computing. Of particular interest to us is one of Denning's motivations: "Establish a new relationship with people from other fields by offering computing principles in a language that shows them how to map the principles into their own fields." He claims that the "principles of computing can be organized into seven categories, each emphasizing a unique perspective on computation." They are: 1. Computation, 2. Communication, 3. Coordination, 4. Recollection, 5. Automation, 6. Evaluation, and 7. Design. These seven "principles" form a foundation which can be useful to recognize and categorize instances of computational thinking and build a framework that can translate computational thinking to contexts outside of computer science.

One problem that Denning does not address is the wide gap between the computer science community and other communities. Given this gap, it is not clear how the framework can be populated with instances and best practices of computational thinking in areas outside of computer science. Our proposed project will help bridge this gap and thus be a part of Denning's goal of "mapping computing principles to other fields". We will do this in the context of Liberal Studies courses including courses in the "softer" social sciences, the humanities, and the arts. We will exploit the unique opportunity available at DePaul in which professionals familiar with computational thinking are teaching Liberal Studies courses in areas outside of traditional information technology.

An additional opportunity was created as CDM faculty have become involved in the administration of liberal studies courses and came to recognize that a number of courses taught by non-CDM faculty implicitly include elements of computational thinking. The proposed project will include active engagement with these instructors and include their courses in the project. However, the concepts of computational thinking and computing principles are foreign to most academics outside of computer science. It is therefore understandable that this community may be skeptical of the usefulness of these concepts, especially without examples of computational thinking in areas outside of traditional information technology. We plan to use the intermediate results of this project, that is examples of computational thinking in CDM Liberal Studies courses across the curriculum, to explain computational thinking and demonstrate the usefulness of the concept to the Liberal Studies courses that non-CDM faculty teach.

2.1 Computational thinking outside the sciences

One of the unique contributions of this project will be the identification of computational thinking in courses outside of both computer science and areas that have commonly be associated with computation such as the hard sciences. In order to illustrate the potential that we see in courses beyond these traditional areas, we discuss some of the computational thinking content of the course GAM 224: Introduction to Game Design. The course content is rich in computational thinking and that the ideas in the course can be used as examples to uncover computational thinking instances in other courses in the humanities, arts, and social sciences.

There is no better way to understand the design of computer games than to think about multiple layers of abstraction. While GAM 224 was not created with the idea of computational thinking in mind, the course has multiple topics and activities that encourage it. The reason for this is clear: computer games rely on the complex interaction of multiple formal systems to produce emergent behaviors in a dynamic environment. Computing is the modern tool for designing and understanding such a system.

GAM 224 is a required course in the Computer Games Development major, recommended as one of the first courses taken by freshman. It has no prerequisites and does no computer game development. Instead it considers the study of computer games from three angles: as examples of media that can be analyzed and critiqued for their thematic elements, formal structure, plot and interactive appreciation, as complex software artifacts subject to technological constraints and the product of a labor-intensive design and implementation process; and as a cultural artifact with behaviors and associations comparable in import to other popular art forms.

GAM 224 is a course in the Arts and Literature domain of the Liberal Studies Program, and it satisfies the same requirement as courses in the departments of Art, English, Communication, Modern Languages, and Music. The Arts and Literature Domain of which GAM 224 is a part requires students to expand their knowledge of the arts while developing critical and reflective abilities.[5] This emphasis on critical and reflective techniques provides many opportunities for computational thinking.

The course content is based on the required textbook for the course[6]. The approach used to describe game design is defined as iterative design. In iterative design, design decisions are made based on the experience of playing a game in development and involves prototyping, playtesting, evaluation, and refinement.[7]. While the textbook and the course involve no programming, iterative design is clearly a technique taken from computer science. This not surprising given that the authors of the text have worked in the computer gaming business for many years. The course considers not just computer games but card, board, and dice games, and other older forms of gaming. Iterative design of a card or board game is something that is widely discussed in the course. Iterative design is an example of the design principle of computing. In the context of software engineering, iterative design is also known as agile software development, a framework that has been proposed, extensively studied, and enhanced. It has developed a set of principles such as "simplicity", "self-organizing teams", etc.[8] that are supposed to enhance the software development process. It would be interesting to investigate which of those principles have been or could be applied to the "iterative design" of games.

The basic definitions used in the course are infused with multiple layers of abstraction, which is again a design principle. Early in the book, three types of rules are defined: constituative, operational, and implicit. Operational rules are the guidelines players require in order to play, such as the rules printed on the box of a board game. Constitutative rules are the underlying

[5]DePaul University, Liberal Studies Program, Arts and Literature Domain Approved Courses, http://las.depaul.edu/lsp/public_html/approved/student/al.htm

[6]*Rules of Play: Game Design Fundamentals*, Katie Salen and Erin Zimmerman, The MIT Press, 2004

[7]Ibid, p. 11

[8]The Agile Manifesto, http://www.agilemanifesto.org/principles.html

logical and mathematical structures in the game. Implicit rules are the "unwritten" rules of the game, such as rules about decorum.[9] Two games are the same if there is a 1-1 relationship between the constituative rules of the two games, so that if you can find a winning strategy in one game you can use the mapping to find a winning strategy in the other game. At the same time, the operational rules for two structurally identical games can vary significantly. While the operational rules are what makes a game enjoyable to play, the constituative rules are the ones that more experienced players are using when they find winning strategies. An activity done early in the course asks the students to play two games that have different operational rules but have the same constituative structure. Students are asked to play tic-tac-toe for a few minutes before a class discussion of the strategy of blocking your opponent from obtaining three X's or O's in a row. Many students are familiar with the game, and frankly a bit bored by it. Next they are asked to play the game 3-to-15, where two players choose numbers between 1 and 9. Each player must choose a number the other player has not yet chosen and the first to obtain 3 numbers that sum to 15 wins. As stated, 3-to-15 is game that involves a lot of mental arithmetic, and students are stunned to learn that with the use of a "magic square", 3-to-15 is exactly the same as tic-tac-toe. With one exercise, the students appreciate the significance of an isomorphism without knowing the mathematical meaning of the term. Through this exercise, students appreciate the concepts of abstraction, layering and modeling, all examples of the design principle.

The course places a strong emphasis on games as systems, and one of the ways of seeing a game is as a cybernetic system. Examples of positive and negative feedback loops can be found in games. A positive feedback loop is described as creating exponential growth or decline, and a negative feedback loop is characterized as maintaining equilibrium.[10] For example, the rule that says that players must win the children's game "Chutes and Ladders" by an exact roll is given as a classic example of a negative feedback loop, maintaining a more level playing field.[11] Another class activity enforces students' understanding of cybernetic systems. They are given various situations in a variety of games, including board and computer games, and must classify those situations as a negative feedback loop, a positive feedback loop, or neither. If they choose negative feedback loop they must indicate what direction the game state is moving in and how the rule aids the movement of the state of the system. If the students indicate a situation is a negative feedback loop, they must indicate what state the rule is trying to preserve and how the mechanism keeps the game in that state. If they choose neither, they must be ready to indicate why neither mechanism is in place. This activity is particularly important because feedback loops are complex and students struggle to understand the difference between the interaction between the game state, the mechanisms detecting the game state, and the mechanisms altering the game state. Through this activity the students learn computational thinking instances categorized in the coordination principle.

[9] Ibid, p. 130

[10] Ibid, pp. 215-216

[11] Ibid, p. 223

3 Implementation Plan

The goal of our project is to build a community of DePaul faculty across all fields that will reach a consensus about how to integrate computational thinking into DePaul's Liberal Studies Program. As mentioned above, the non-computing faculty may not be familiar with the concept of computational thinking. Without an extensive set of examples of computational thinking across diverse areas, they may be skeptical about the concept of computational thinking.

In our project, we will pursue a two-phase approach in order to generate the examples that we need. We intentionally focus on Liberal Studies courses closer to computing first – i.e. courses taught by CDM faculty members – so that the computational thinking skills taught in the course are explicit and well understood. The categorization of these skills is clearer for courses that are more directly related to computer science. As the framework becomes populated with examples, we will move to courses further from computing and taught by non-CDM faculty. We will use the examples already in the framework to uncover computational thinking skills implicitly taught in these courses or discover how computational thinking can be applied in the courses in novel and effective ways.

The project we propose to undertake includes the following stages:

3.1 Data gathering by CDM faculty members

We will document and make explicit instances of computational thinking in CDM Liberal Studies courses. We will categorize and populate the framework (based initially on Denning's "Great Principles" categories but adjusted as needed) with examples of and best practices for teaching computational thinking. We will work with Prof. Kanj on CSC 235: Problem Solving, with Prof. Schaefer on CSC 233: Codes and Ciphers, with Prof. Fang on ECT 250: Internet, Commerce, and Society, with Prof. Furst on CSC 239: Personal Computing, with Prof. Miller on HCI 201: Multimedia and the World Wide Web, with Prof. Roberts on ANI 101: Animation for Non-Majors, with Prof. Irvine on DC 105: Digital Media Literacies, and with Prof. Jones on ANI 230: 3d Modeling for Animation and Gaming. Prof. Settle will cover GAM 224: Introduction to Game Design and Prof. Perkovic will cover IT 130: The Internet and the Web.

3.2 Develop explicit computational thinking learning goals

In partnerships with the participating CDM faculty, we will develop computational thinking learning goals for the above courses. These goals will be clear, achievable, and measurable. We will also develop assessment tools that we will use to evaluate the learning of these computational goals.

3.3 Peer review

In this first phase of the project, the principal investigators will consult with Prof. Denning about the categorization of the computational thinking instances. We will use Prof. Denning's expertise to make sure we are recognizing computational thinking instances correctly and categorizing them into appropriate principles. We will consult with him about the adequacy of the computational thinking learning goals as well as of the assessment tools we will develop.

3.4 Selection of non-CDM participants

In the second phase of the project, we will begin working with non-CDM participants. We will invite DePaul faculty to apply to join the project and we will make a selection of faculty across disciplines. We plan to select those faculty who are teaching courses that already have, or that can benefit from, elements of computational thinking. The framework created in the first phase of the project will be used to illustrate to non-CDM faculty members what is meant by computational thinking.

3.5 Data gathering with CDM and non-CDM faculty members

We will document and make explicit instances of computational thinking in selected non-CDM Liberal Studies courses. We will do this in partnership with the CDM faculty participants. We will also develop explicit computational thinking learning goals for non-CDM courses.

3.6 Dissemination of our results and our framework

We will write a document that describes the framework that we used to incorporate computational thinking in liberal arts courses. We will publicize our framework and our assessment results in appropriate conferences and journals and on a project web site. More details are provided in the section discussing the Dissemination Plan for the project.

4 Management Plan

We summarize the project activities:

Project milestones	**Expected outcomes**	**Responsible parties**
April 30, 2009		
Final selection of CDM Liberal Studies courses		All CDM participants

Project milestones	**Expected outcomes**	**Responsible parties**
July 31, 2009		
CDM liberal courses analyzed	An analysis of the computational thinking content in the selected courses, their categorization into the principles categories, development of computational thinking learning goals and assessment tools.	All CDM participants and Denning
December 1, 2009		
First Assessment	All selected CDM courses taught in the Fall quarter will be assessed.	Co-PIs
April 1, 2010		
Preliminary Framework	A preliminary framework document will be developed and shared with interested non-CDM DePaul faculty.	Co-PIs and Denning
Second Assessment	All selected CDM courses taught in the Winter quarter will be assessed.	Co-PIs
April 30, 2010		
Selection of non-CDM faculty and courses		Co-PIs
July 31, 2010		
Selected non-CDM Liberal Studies courses analyzed	The selected non-CDM courses will be analyzed. The CDM participants will work with the Co-PIs to educate the non-CDM faculty about instance of computational thinking across disciplines. The computational thinking content in the selected courses will be categorized, and computational thinking learning goals and assessment tools will be developed.	All participants (CDM and non-CDM)
Third Assessment	All selected CDM courses taught in the Spring quarter will be assessed.	Co-PIs

Project milestones	Expected outcomes	Responsible parties
December 1, 2010		
Fourth Assessment	All selected CDM and non-CDM courses taught in the Fall quarter will be assessed.	Co-PIs
Framework Finalized	A final framework document will be developed.	Co-PIs with Denning

5 Project Expertise

DePaul CDM and its faculty are in excellent position to provide the leadership in this project. CDM is one of the largest and most innovative information technology institutions in the country. Over 2,000 graduate students and nearly 1,000 undergraduates are enrolled in 14 Bachelors and 15 Masters degree programs, ranging from traditional programs like computer science to degrees focused on the digital arts such as digital cinema and animation. CDM also serves all DePaul students through the Liberal Studies Program, offering 40 courses in 5 different domains. Because of its size and diversity, CDM is an excellent setting for identifying challenges and also for expanding computing education to under-represented demographic groups. Because of its leadership in course and new program development, it is the ideal testbed for proposing strategies for revitalizing computing and for developing and disseminating new models for computing curricula. And because of its aggressive initiatives to include computing-themed courses in all areas of Liberal Studies, it is a model for other universities to integrate computing into the broader undergraduate curriculum.

Amber Settle has been a member of the Undergraduate Common Core Committee since its inception and chaired the committee for 2 years. The Undergraduate Common Core Committee is charged with overseeing undergraduate courses that either have large enrollments or are shared between a number of degree programs. She also served as the original co-chair of the Teaching and Learning Excellence Committee, which is overseeing CDM efforts at improving teaching and learning across the curriculum. She has also served on the Advising and Undergraduate Committees and the Program Committee for the gaming degrees at CDM. She is the founding head of the Educational Research group at CDM and organizes a bi-quarterly seminar series for the group. At the university level, she served for over four years as a member of the Liberal Studies Council which is the faculty body that oversees the general education program at DePaul. She also served for three years on the Scientific Inquiry Domain committee which reviews all proposals for courses in that domain of the Liberal Studies Program.

Ljubomir Perković has served on the Program Committee for the computer science degrees since its inception and currently serves as chair of the committee. He has been instrumental in the revision of the M.S. in CS in 2004 and this year. He also worked on the revision of the B.S. in CS and has been active in promoting the program and serving as an advisor for students entering the new program. He has also served on the Advising and Assessment Committees at CDM. He has an active research agenda in distributed algorithms.

Xiaowen Fang has been at CDM for 9 years has been active nationally in the area of human-computer interaction. He has held high profile positions both with the Association for Information Systems and the International Journal of Human-Computer Interaction. A faculty member at CDM since 1998, Jacob Furst has an active research program in visual computing and image processing and has been a co-PI on a funded NSF Research Experience for Undergraduates project. He has served on both the Liberal Studies Council and the Scientific Inquiry Domain Committee of the Liberal Studies Program. Matthew Irvine has been at DePaul University since 1997 and currently serves as the Director of the Communication & Digital Cinema Center. He has developed much of the digital cinema curriculum at CDM while remaining active as a director, editor, and writer of films. An Assistant Professor in CDM, Joshua Jones has taken an large role in the development of the animation program, led the effort to revamp course evaluations as a member of the Teaching and Learning Excellence Committee, and currently serves as one of the CDM representatives on the Liberal Studies Council. Iyad Kanj is an expert in the area of parameterized and exact computation. At DePaul University he has worked extensively on improving students' critical thinking skills including founding a Problem Solving Club and coaching a DePaul ACM Programming Team to the world finals in 2005-2006. Craig Miller has an active research program in cognitive science, human- computer interaction, and information technology education. He has served in leadership roles in several program committees at CDM as well as on the Teaching and Learning Excellence Committee, and he has received funding from the NSF for multiple educational projects. As a member of the CDM faculty since 2005, Scott Roberts has developed curriculum in the areas of animation, computer game development, and interactive media, working on over half a dozen program proposals. He has also had a number of solo and group exhibitions that have been well reviewed by national newspapers. Marcus Schaefer has a strong research program in the area of theoretical computer science, focusing much of his recent work on graphs. He has developed over half a dozen new courses in both computer science and mathematics and is active on the national level with the Conference on Computation Complexity.

Peter J. Denning is a Distinguished Professor at the Naval Postgraduate School in Monterey, California. He chairs the Computer Science Department and directs the Cebrowski Institute, an interdisciplinary research center for innovation and information superiority. In the 1990s he was at George Mason University, where he was vice provost, associate dean, CS department chair, and founder of the Center for the New Engineer. In the 1980s, he was the founding director of the Research Institute for Advanced Computer Science at NASA-Ames, and was co-founder of CSNET. He received a Ph.D. from MIT and BEE from Manhattan College. He was president of the Association for Computing Machinery (ACM) 1980-82. As chair of the ACM publications board 1992-98, he was project leader for the ACM digital library, now the ACM's crown jewel. He leads the Great Principles of Computing Project, which is identifying the scientific theories of computing and applying them to curriculum innovation. He also co-leads an Innovation Project that has identified and teaches the seven foundational practices of innovation. He has published 7 books and 315 articles on computers, networks, and their operating systems. He is working on two more books, one on the foundational practices of innovation and the other on the great principles of computing. In 2002, he was named one of the top 5 best teachers at George Mason University and the best teacher in the School of Information Technology and Engineering. In 2003, he received one of Virginia's 10 outstanding faculty awards. He holds three honorary degrees, three professional society fellowships, two best-paper awards, three distinguished service awards, the ACM Outstanding Contribution Award, the ACM SIGCSE Outstanding CS Educator Award, and the prestigious ACM Karl Karlstrom Outstanding Educator Award. In 2007 ACM gave him a special award for 40 years of continuous volunteer service and the NSF gave him one of two Distinguished Education Fellow awards.

6 Community Support

Our project directly addresses a need recognized by the larger intellectual community including the NSF. Computational thinking has the potential to revolutionize the American educational system, and many institutions have taken an interest in developing this potential. Few institutions, however, are in as strong a position as DePaul CDM to lead the expansion of computational thinking into areas such as the humanities, the arts, and the social sciences. Our proposal has attracted the attention of Peter Denning, a leader and innovator in computing education for the past 30 years. He believes that this proposal is particularly interesting because of the potential to build bridges between different educational communities. Cross-disciplinary communication shows great potential for revitalizing computing education. A letter of support from Prof. Denning is included with this proposal.

There is also strong support for this project within DePaul CDM. The Dean of CDM is particularly enthusiastic about the proposal and a support letter from Dean Miller has been included with this proposal. The proposal has also attracted the active participation of 8 faculty within CDM, faculty whose interests and research areas include computer science, human-computer interaction, animation, and digital cinema. Five of the participating faculty have Ph.D.s and three of the participating faculty have MFAs, demonstrating that CDM has the kind of working environment that can produce a framework that speaks to a broad range of faculty at DePaul and beyond.

7 Evaluation Plan

In order to ensure that the framework produced by this project is appropriately rigorous, evaluation and assessment of this work will be conducted at every stage of the project. In the initial stage of this project, several documents will be produced for each CDM course covered by the project:

- A classification of the concepts covered in the course, indicating how those concepts employ computational thinking

- A set of learning goals for the course with respect to computational thinking

Further, the project participants will develop a set of questions that can be used to assess the computational thinking learning goals produced for the initial set of courses. As needed, these assessment questions may be global for all CDM courses or may be specialized by subject area. Each of these documents will be evaluated by Prof. Denning to ensure that the concepts have been categorized correctly and that the assessments are appropriate for the identified learning goals. Once any changes he suggests are incorporated, the assessment questions will be used to evaluate the courses as taught beginning in the Fall 2009 and repeated each subsequent quarter. Data gathered from these assessments will be evaluated on quarterly basis to gauge the appropriateness both of the course and of the categorization of the course concepts. An article discussing the results of the first stage and including the data from the initial assessments will be produced by the Summer 2010.

All of the documents and data gathered during the first stage of the project will be used to create a larger framing document to be used during the second stage of the project when non-CDM courses are included. The framing document will organize the concepts, learning goals, and assessment questions in several ways, including by subject area and by computational thinking approach. The second stage of the project will include the same steps taken in the first stage, including the categorization of concepts, identification of learning goals, and development of assessment materials, but this time for non-CDM courses. Again the documents produced will be reviewed by Prof. Denning before the assessment of the non-CDM courses commences in the Fall 2010. By the end of the Fall 2010 a final framing document incorporating both CDM and non-CDM courses will be produced. A second article discussing the results of the second stage and including data from all assessments will be produced by the Spring 2011.

8 Dissemination Plan

There are several ways that the work in this project will be disseminated. By the nature of the proposal, part of the dissemination will need to be local to DePaul University. The Educational Research Group at CDM will hold a quarterly seminar beginning in the Spring 2009 to discuss the ongoing status of the project. In order to engage faculty outside of CDM as soon as possible, the members of this project will submit an article to the yearly DePaul Faculty Teaching and Learning Conference to be held in the Spring 2010.

However, for this project to realize its potential, the results must be disseminated widely in the information technology education community and beyond. At a minimum, the articles produced at the end of each stage of the project will be submitted to a national technology education conference. The Technical Symposium on Computer Science Education (SIGCSE) is the natural venue for this work. It is possible that parts of this project will have international interest and should be submitted to the Conference on Innovation and Technology in Computer Science Education (ITiCSE). Some of the work may be appropriate for a more general conference such as the Science of Learning Conference. More comprehensive treatments of the work, particularly the analysis of the assessment data gathered during the project, might be best submitted to a national education journal such as Computer Science Education or the Journal of Information Technology Education.