Strategies to Integrate Writing in Problem-Solving Courses: Promoting Learning Transfer in an Interdisciplinary Context

Dr. Reneta Davina Lansiquot, New York City College of Technology

Reneta D. Lansiquot is an Associate Professor and Program Director, Bachelor of Science in Professional and Technical Writing, as well as the Assistant Director of the Honors Scholars Program at New York City College of Technology. Dr. Lansiquot earned an A.A.S. in Computer Information Systems, a B.Tech in Computer Systems at the New York City College of Technology, City University of New York, a M.S. in Integrated Digital Media at Polytechnic University (now NYU Polytechnic School of Engineering), and her Ph.D. in Educational Communication and Technology at New York University. Her mixed methodology research, focusing on interdisciplinary studies, has been presented at numerous national and international conferences and published in peer-reviewed book chapters and journal articles on varied topics such as technical writing, the future of science education, game design, virtual reality, and problem solving. Her book is entitled Cases on Interdisciplinary Research Trends in Science, Technology, Engineering, and Mathematics: Studies on Urban Classrooms (Information Science Reference, 2013).

Dr. Candido Cabo, New York City College of Technology

Candido Cabo is a Professor in the Department of Computer Systems Technology at New York City College of Technology, City University of New York. He earned the degree of Ingeniero Superior de Telecomunicacion from the Universidad Politecnica de Madrid in 1982, and a Ph.D. in Biomedical Engineering from Duke University in 1992. He was a post-doctoral fellow at Upstate Medical Center, State University of New York, and a research scientist in the Department of Pharmacology at the College of Physicians and Surgeons of Columbia University. Since 2005, he has been a member of the doctoral faculty at the CUNY Graduate Center. His research interests include computer science and engineering education and the use of computational models to understand and solve problems in biology. He has published a number of peer-reviewed journal articles on computer modeling of biological processes, particularly relating to cardiac electrophysiology.

Strategies to Integrate Writing in Problem-Solving Courses: Promoting Learning Transfer in an Interdisciplinary Context

Abstract

Reflective writings, the contextualization of learning experiences, and the application of learning to real life all facilitate the transfer of interdisciplinary learning. Such strategies include making explicit to students the need for such a transfer, advising them to follow the appropriate course sequence, emphasizing material they need to transfer to other courses, practicing transfer by inviting guest lecturers, developing metacognitive skills, and reinforcing concepts by using them in different contexts. As the transfer of learning does not occur automatically, curricular and course design should intentionally emphasize the connection between courses.

Problem Solving with Computer Programming (PS) is a required course for first-year computer systems majors, offering an ideal opportunity to establish a transfer structure. To make students aware of the connections between PS and English Composition (EG1), another required course, and to facilitate the transfer of skills, we developed a learning community (LC) linking these courses. This innovative approach to teaching computing and writing to first-year computer systems majors at a college of technology uses programming narratives as its theme. Students write and implement narratives using computer programming to develop a narrative-driven video game prototype. They use *Alice*, a three-dimensional animation software (www.alice.org).[9,10] The LC emphasizes the importance of connecting courses in the major and those in general education. The LC builds on our previous research, which found that introducing narrative elements into problem-solving courses improves overall student performance and computer programming-related problem-solving skills in particular.

This study presents best practices and lessons learned from our LC, and we present three novel strategies to integrate writing in PS courses for majors and non-majors. First, since implementation of LCs is not always feasible, to infuse narrative elements into problem-solving we developed a narrative module to help students develop narrative and writing skills that can be incorporated in all sections of the PS course. Second, we developed a series of student-assessed case studies that can be integrated in all sections of the PS course for computer systems majors. Cases studies provide a narrative context in which students learn basic constructs of computer programming such as sequencing, selection and repetition structures. Third, we created a general education interdisciplinary course, Programming Narratives: Computer Animated Storytelling, open to non-computer majors, which emphasize creative writing and computational thinking. In this interdisciplinary course, students learn the structure of narrative, concepts of problem solving, and the logic of computer programming languages as they develop a narrative-driven video game prototype. This process helps students achieve the college-wide learning goal of making meaningful and multiple connections among the liberal arts as well as between the liberal arts and the areas of study leading to a major or profession.

1. The Challenge: Interdisciplinary Learning in Undergraduate Education

For approximately two-thirds of their course credits, students majoring in computing and engineering at our institution must take a combination of courses in their discipline. General education courses in the liberal arts and sciences comprise the remaining one-third to meet graduation requirements. Each degree program has well-defined learning outcomes mapped to the learning outcomes of each course within the program. Learning outcomes usually expand across individual courses. Students must consider the different courses required for graduation as part of a single whole, the degree program. They then can establish synergistic connections between the different courses of the curriculum.

In reality, students often do not relate courses in their major to general education courses. Consequently, they cannot transfer skills between them. This lack of transfer is not a problem unique to our institution. Abundant evidence suggests transfer of skills between courses is relatively rare.[2,4,13] The problem occurs not only between general education and the major, but also between courses in the major. Many factors hinder this transfer. Students may have forgotten some of the material learned in a previous course, they may not perceive the connections, they may see the connections but cannot use the material in meaningful ways in a different context, or the instructor's pedagogical approach may not foster transfer.[4]

Approaches used to facilitate transfer of learning include the use of reflective writings, contextualization of learning experiences, and application of learning to real life.[4] To make transfer of learning explicit to students, instructors advise students to take courses in the appropriate sequence, emphasize in each course the material transferable to other courses, model transfer by inviting guest lecturers, develop students' metacognitive skills, and reinforce concepts by using them often and in different contexts. However, regardless of the strategies used, transfer of learning does not occur automatically. Curriculum and course design should emphasize the connection between courses to stimulate transfer.

The primary strategy used at our institution to connect courses from different disciplines in order to provide our students with an interdisciplinary learning environment is the first-year LC model. A first-year LC is a group of students who enroll in two or more courses, generally in different disciplines that are linked together by a common theme in an academic semester. We have implemented LCs at our institution for over 10 years, and the academic performance of students participating in LCs reflects national trends.[1, 14] We have been running the LC linking the PS and EG1 courses once per year for the last six years. The best practices and lessons learned in that LC are described in section 2. Those lessons inspired the strategies to integrate writing in PS courses for majors and non-majors that are described in section 3. To develop further an interdisciplinary culture, our institution now requires students to complete an interdisciplinary general education course. This course links two or more disciplines and more than one faculty member from two or more departments as team-teachers. The lessons learned from the PS/EG1 LC guided the development of one of those general education interdisciplinary courses, which is also described in section 3.3.

We believe that this intentional emphasis of connections between courses and disciplines develops interdisciplinary skills and perspectives important for the completion of a degree.

Perhaps more importantly, it lays the groundwork for interdisciplinary thinking in the workplace upon graduation.

2. Programming Narratives: Synergies Between Writing and Computer Programming

Our institution offers a PS course to prepare students in computing and engineering majors for a rigorous first programming course (CS1). This course introduces students to problem-solving concepts using constructs of logic inherent in computer programming, including procedural programming and object-oriented programming. Students learn the nature of problems, common solution approaches, and analysis techniques. During the first 3 weeks of a 15-week semester, we emphasize solving problems in a context known to the students, like mazes or tic-tac-toe. The course introduces computer programming constructs such as sequencing, selection structures, and repetition loops to solve different problems. To learn these constructs, students also use a flowchart interpreter (Visual Logic, www.visuallogic.org). Flowchart interpreters allow students to conceptualize and solve their problems without the complicated syntax of a full-fledged programming language. In the following 10 weeks of the semester, students use *Alice* to create interactive animations, develop and reinforce their problem-solving skills, and familiarize themselves with object-oriented programming concepts. In the final two weeks of the semester, we expose students to an Integrated Development Environment (IDE), such as *NetBeans* or *Eclipse*, which helps programmers to write, compile, and test computer programs. The students also learn basic Java programming to facilitate the transition from the characters and the game world they have created and manipulated in *Alice* to the more abstract objects found in a user interface window such as buttons, textboxes, and labels in more advanced programming courses.

Preliminary PS courses are effective in preparing at-risk students with minimal previous programming experience and weak mathematical background for a CS1 course in C++ or Java. For many years, we used a programming language (Visual Basic [VB]) in our PS curriculum as the context for teaching and learning computer programming concepts and skills. This practice resulted in low passing rates. Following the successful experience reported by others,[20,21,24] and our own experience,[16] we introduced *Alice* to all sections of the PS course in spring 2010. *Alice* improved student performance, retention, attitudes towards computer science.[20,21,24] *Alice* provides the ability to implement narratives using computer programming that VB does not. Expanding earlier pedagogies used in the teaching and learning of computer programming, we investigated the effect of using storytelling as the context and interdisciplinary learning communities as the pedagogical approach.[6,7,16-19]

A large study involving over 1,500 students over a period of five years showed the introduction of *Alice* in the curriculum of the PS course increases the passing rate in the course by about 8% (from 70% to 78%) in comparison to the VB-based PS course.[7] Lower rates of withdrawal or getting an F in the course account for the improvement. More importantly, the higher passing rate in the *Alice*-based PS course does not mean students are less prepared for the subsequent CS1 course: students taking the *Alice*-based PS course pass the CS1 at the same or better rate than students who took the VB-based PS course.[7]

The PS course is not the only gateway course our students take in their first year; EG1, required of all students at the college, has also challenged our students. These gateway courses lay a general foundation for the other academic courses in general education as well as for more advanced work in computer programming, database, and networking courses. To improve student success, in fall 2009 we decided to link these courses in an interdisciplinary LC. The LC is one of the ten high-impact educational practices recognized nationally to improve student persistence and retention.[1,14] The LC makes more obvious to students the connections between these courses, thus facilitating the transfer of skills between these courses and beyond. This LC is an innovative approach to the teaching of computing and writing to first-year computer systems majors at a college of technology. The theme of the LC is programming narratives—that is, having students write narratives they will then implement, using computer programming, as a video game prototype. The LC approach incorporates and builds on many of the suggestions in the literature on how to facilitate transfer. Moreover, it makes a statement early in the students' academic careers about the importance of connecting courses throughout the curriculum to facilitate transfer. Other studies provide a detailed description of the LC, including selection criteria and an evaluation of its effectiveness.[19]

The LC builds on previous research showing that introducing narrative elements into PS courses improves student performance in general as well as in computer programming-related problem-solving skills.[6,7,16-19] Our experience shows teaching the PS course in the interdisciplinary environment of a LC further increases student retention and performance. An interdisciplinary LC with strong narrative components linking writing stories with writing code improves students' ability to transfer programming concepts to practical programming skills.

Students in the computer system major often perform better in introductory computer courses than in EG1. This may be because students do not relate computer courses to general education courses and consider computer courses more important for their degree than general education courses. The interdisciplinary environment of a LC not only boosts student performance in the PS course but also improves student performance in the EG1 course to the same level as in the computer courses.[19] It is possible the intentional interdisciplinary learning occurring in the LC helps students connect writing and computer courses, facilitating transfer of concepts and skills with the consequent improvement in academic performance.

3. Strategies to Integrate Narrative Elements in Computer Programming Courses

Despite the success of the programming narratives in the LC, it is not always feasible to implement LCs linking every section of the PS with EG1 courses. Currently, we run (and will keep running) one LC linking one section (24 students) of the PS with an EG1 course per academic year. Since we run about twenty-five sections of the PS course per year, it means that only 4% of students experience the interdisciplinary environment of a LC. Scaling up the LC model is not simple given scheduling issues, the social and academic support provided to LC students, and the need for faculty training to implement cooperative learning, alternative assessment in the classroom, cross-disciplinary writing assignments, and critical thinking activities. Therefore, we have developed three different strategies to integrate writing that incorporates narrative elements into PS courses for majors and non-majors. The strategies described in sections 3.1 and 3.2 will make it

possible to incorporate programming narratives in all sections of the PS course. The strategy described in section 3.3 will allow students in non-computer systems majors to experience the synergies between creative writing and computational thinking. These strategies help students make the connections between courses in the major and general education courses to facilitate learning transfer.

The first strategy is a module to infuse narrative elements into PS courses. This helps students develop narrative and writing skills they can incorporate into all sections of the PS course. The second strategy is a series of student-assessed case studies they can integrate into all sections of the PS course for computer systems majors. Case studies provide a narrative context in which students learn basic constructs of computer programming such as sequencing, selection, and repetition structures. Finally, recognizing that computational thinking and skills are important for every student at our institution, we created a general education interdisciplinary course, Programming Narratives: Computer Animated Storytelling. This course is open to non-computer majors and emphasizes creative writing and computational thinking. In this interdisciplinary course, students learn the structure of narrative, concepts of problem solving, and the logic of computer programming languages as they develop a narrative-driven video game prototype. The course helps students achieve the college-wide learning goal of making meaningful and multiple connections among the liberal arts, between the liberal arts and their majors, and on into their professions.

3.1 Developing Writing Skills for Computer Programming Courses

An important component of the PS course, whether part of a LC or not, is a group project in which students create a story that they later implement as a video game prototype using *Alice*, developing their computer programming concepts and skills along the way. We believe that one of the reasons why students perform better in sections of the PS course linked to a LC is because the narrative skills learned in the EG1 course allows them to create more engaging stories which they then implement as a computer program using *Alice*. Students taking a PS course not linked to a LC may not be taking EG1 in the same semester, they may have forgotten about the narrative and writing skills learned in EG1, or the EG1 instructor teaching the course may not emphasize the narrative skills that could be applied readily to develop a video game prototype. We have observed that the video game narratives created by students in the LC are more engaging and interesting than the ones developed by students not in the LC. We believe that students' engagement with their narratives is reflected in more engagement and better performance in the PS course.

Therefore, we have created a module to help students develop narrative and writing skills that can be incorporated is all sections of the PS course. The module has the structure of a case study and prepares students to create an original background story for a video game. The goal of the case study is to engage student writing and understanding of video game narratives. After completing the case study, students should be able to demonstrate an understanding of the purpose of game narratives, the structure of game stories and related narrative structures, including the hero's journey.[3,25] Moreover, the addition of this module to PS courses intentionally links a course in the major (the PS course) to general education skills (reading and writing) learned in

other courses (i.e., EG1), thus reinforcing those skills and helping students making connections between their general education and major courses.

The module consist of two components: understanding the general elements of plot and drama structure through classical works, and guidance for the development of characters and video game narratives. Before starting this narrative module, students are asked to read Aristotle's *Poetics* and Sophocles' *Oedipus the King*. Aristotle's *Poetics* introduces students to the elements of a plot (i.e., the cause-effect arrangement of incidents in the story), and the differences between plot and story. In particular, students learn the concept of "unity of action," Aristotle's idea of a plot with a beginning (exposition and rising action), middle (the climax) and end (falling action and resolution). Students then apply these concepts to the analysis of *Oedipus the King*. Students also learn the other elements of drama as outlined by Aristotle: theme, character, language, rhythm, and spectacle.

Aristotle's ideas on character development also aid student learning. Aristotle explains that a character must be good; that is, appropriate to the person in whom it is depicted, true to life, and consistent—or, if the character is inconsistent, at least he or she should be consistently inconsistent. Stories should contain a protagonist and an antagonist. On the other hand, an antagonist is simply someone in the way of the protagonist. In different activities, students are asked to write down examples of different kinds of protagonists and antagonists, conceptualizing their traits, and to share with other students the characteristics of their favorite video game character. This activity is important because, while our students may find it difficult to connect with classical stories and drama, they are gamers, and they can relate to video games and their characters.

In addition to having characters, a good video game needs a story, a narrative.[3,25] Video game narratives connect the events of the game, providing a structure and a justification, a reason, for the game play encounters. They pull the player forward through the experience, creating the desire to achieve the hero's goals and, more importantly, to see what happens next. Narratives also achieve this goal through immersion, reward, and identification. Immersion is related to the psychological state of "flow"[11] and suspension of disbelief. When players are immersed in a game, the real world ceases to exist, and the game world becomes their reality. Narrative can serve as a reward; for example, in *God of War*, the back-story is revealed gradually—after a chapter is cleared, we find out how the protagonist, Kratos, came to be in such dire straits in the first place. For the player to be able to identify directly with a game protagonist, the narrative should lay everything out for the player. The player wants to know what's what, who's who, and the state of the world around him/her. Narrative also helps players form the desire to become the central character.

During a story, the main character experiences interpersonal conflict, which is conflict with one or more other characters, nature or society, or an internal conflict, with him- or herself. The importance of conflict, and the personal growth that comes about through conflicts, has been described by the noted anthropologist and expert in comparative mythology, Joseph Campbell.[8] He analyzed stories from around the world to identify a common pattern or theme that was intrinsic in many cultures, even among cultures that had no contact with each other throughout their histories. This resulted in the visual model for a hero's journey that looks like a circle,

which begins and ends at the top of the circle with the Ordinary World, moving counter-clockwise through a series of stages—Call to Adventure; the Refusal of the Call; Meeting the Mentor; Crossing the Threshold; Tests, Allies, and Enemies; Approach the Inmost Cave; the Ordeal; Reward; the Road Back; Resurrection; and Return with the Elixir. The hero's journey thus includes a departure, an initiation, and a return.

In a video game, the protagonist is the character who goes on the journey, and it is also the character the player assumes. We see the game world through this character. From a story perspective, the protagonist is the character with whom the audience is asked to identify. If the players do not empathize with the character, then they will have a harder time engaging with the game. Therefore, students are asked to think and complete the steps of the hero's journey of their character.

Using the departure section of the hero's journey plot structure, students write an original background story for a video game, including the protagonist and antagonist. Students should explain in a three-page narrative what the central character wants and what the expertise of this hero is. Why is this character on this important quest? Students should provide sufficient descriptions so that it is clear what the player needs to do to "win" the game. Their videogame background story should end where game play begins. Students should also write a separate page that includes a well-developed paragraph describing the game play addressing the following: Who is the target audience? Why should the player care about the protagonist? Why is the story socially relevant or engaging to the proposed target audience?

The video game development project in this module is a combination of individual and group work. Students develop individually their video game plot and present it to the class. The best ideas will be chosen and developed further in groups of four or five students. Overall, this module should not take more than 10-15% of class time in the PS course. We are still in the early stages of implementation and have piloted the module in spring 2015 in one PS course that had a maximum enrollment of 24 students.

3.2 Student Stories as Contexts for Computer Programming Courses

In the PS course, students learn basic procedural programming concepts, like input, sequencing, selection (if/else), repetition and output using flowchart interpreters like Visual Logic (see above). Later on in the course, students use *Alice* to create interactive animations, to further develop and reinforce their problem-solving and procedural programming skills as well as to introduce object-oriented programming concepts and skills. The goal of the narrative module described earlier is to help students develop engaging narratives that they can then implement as a computer program using *Alice*, and improve their success in the PS course.

An important difference between the flowcharting and the *Alice* components of the PS course is that, in the flowcharting component of the course, instructors, in general, pose the problems that the students will solve to develop their problem-solving skills. That is not the case in the *Alice* component of the course because students create their own stories (i.e., their own problems) that they then solve by implementing as a video game prototype. When trying to solve flowcharting problems, students have difficulty translating word problems into a computer algorithm.

Moreover, most problems proposed to students are closely related to mathematics and accounting, and our students are not well prepared in mathematics. Moreover, students are not interested or engaged by the problems proposed to them in the flowcharting component of the course. Researchers have shown that an understanding of (and an engagement with) the problem domain to be solved by implementing a computer program should be a prerequisite for writing the computer program itself.[5,12,22,23] Therefore, the students' inability to create a mental model[14] of a given problem domain hinders their ability to develop problem-solving skills and write computer programs.

Here, we describe our solution to create problem domains that students could understand, relate to and be engaged with, so they can be used as the contexts to develop problem-solving and procedural programming skills in the flowcharting component of the PS course. Our approach is based on the premise that students themselves know better which problems are relevant to them, which problems they can relate to and they can understand. In fall 2014, we selected a group of five students, majoring in computer systems, and who passed the PS course in the last three years, to develop stories that could be used as context to solve flowcharting problems. Those five students took a section of the PS course that was linked to a LC, so, in addition to understanding well the course in which the stories were going to be used, they had considerable narrative and writing skills. The students themselves suggested which flowcharting assignments could be asked from the stories. The stories were developed iteratively, following a combination of individual writing, group discussion, and faculty suggestions, to further improve the versions of the stories. Students were provided with a small stipend.

The five stories developed by the former LC PS students (3 male and 2 female) range in length from 1,200-2,600 words, and have different themes. The themes reveal what is important, interesting, and engaging to our students. One story is a fantasy (*Shell Sleuth*), another is science fiction (*Mind Games*), and the other three are New York City stories imbued with magic realism (*Belvedere*, *Movement in Green*, and *The Shadows of Invisibility*). Summaries appear below. In spring 2015, we piloted *Shell Sleuth*, *Mind Games*, and *Belvedere* with approximately 240 students in 10 PS course sections.

Shell Sleuth
This story takes place in a fantasy world during a war between Elves and Orcs. The protagonist is an Elven spy named Zerk who must calculate the size of the enemy army. To do this, he has disguised himself to sneak aboard one of the enemy creature ships. He is on his way to an outpost to which all the forces are heading for a celebration. Once there, he will be able to calculate the size of the enemy forces by counting the ships at the post.

Mind Games
What happens when you create a machine with artificial intelligence? A better future or total annihilation? Well, the genius Professor Harry Odum didn't think much of his invention ARIIA. He also did not think much of placing ARIIA at the helm of the United States nuclear missiles. When the professor's son, Michael, comes to visit the Arizona military base, ARIIA begins to act strangely. Michael initiates a game with ARIIA with all of the military's nuclear missiles as the prize for the winner and total destruction for the loser. Join Michael as he becomes entwined in a game of machine versus human with a machine that knows a little too much about Michael.

Belvedere
Isaiah Jones is experiencing the American dream, but not one filled with glamour, expensive cars, and a college education. Isaiah is experiencing the "other" American dream, days of repetitive and monotonous work, no college degree, and no money. "Having no luck" cannot even begin to describe Isaiah's life, but he is not a lost cause because he possesses an impressive witty brain. When hope seems lost and there seems to be no chance of a better life, James Hensen, a rich entrepreneur, offers an opportunity to the young man in the form of delivering packages for large sums of money. However, James does not seem to be the entrepreneur he presents himself to be; rather, Isaiah grows suspicious when James asks him to meet at Belvedere Castle in Central Park. Isaiah, presented with the quandary of delivering the packages or not, begins to weigh his options. What will he choose to do?

Movement in Green
Ivory Nyr, an ordinary college girl, discovers only three nights before New Year's Eve that a mysterious, invisible, omnipotent force named Magister has been watching her. After Ivory completes a Good Samaritan deed, Magister assumes human form to speak with her directly on an empty, speeding train that seems to never reach its destination. Here, the entity reveals Ivory has already participated in three challenges as part of his "game" and informs her that, with each challenge, a life is saved. Ivory has a mere 72 hours to complete the remaining challenges. If she fails to fulfill them before the New Year arrives, or if she declines the offer, the lives at stake will be lost—including her own.

The Shadow of Invisibility
Samantha, a first-year college student, starts as an ordinary girl who often feels unnoticed. Her day begins with its usual boring pattern, but something about this day feels strange. She senses some mysterious dark figures are following her. On her way home, several of them confront her and reveal themselves to be an underground family. They tell Samantha she has the power of invisibility and ask her to join them. Although she declines, her life is forever changed. She becomes extremely sneaky and devious towards the people around her. Ultimately, Samantha's life is forever changed.

To implement the case studies, we conducted a workshop with instructors of the PS course to share suggestions on how to incorporate those stories in the curriculum. The stories will be used as the context to create flowcharting assignments, some suggested by the students who wrote the case studies and others by the instructors, for in-class or outside-class assignments, with students preferably working in teams. Student and faculty surveys will further help us refine the case studies to make them more engaging to students and more relevant as pedagogic tools to improve student success in the PS course.

3.3 Interdisciplinary Creative Writing and Computational Thinking Course for Non-Majors

To further develop a culture of interdisciplinary thinking and practice, our institution requires students to complete an interdisciplinary general education course linking two or more disciplines. Although there are different definitions of interdisciplinary studies, we consider interdisciplinary a course that involves two or more academic subjects or fields of study

organized around synthesizing distinct perspectives, knowledge, and skills. Interdisciplinary courses focus on questions, problems, and topics too complex or too broad for a single discipline or field to encompass adequately; such studies thrive on drawing connections between seemingly exclusive domains. Usually theme-based, interdisciplinary courses intentionally address issues that require meaningful engagement of multiple disciplines.

Although many academic disciplines, such as Engineering, are inherently interdisciplinary, to be considered an interdisciplinary course at our institution the course must be team taught by faculty from two or more departments in the institution (exceptions are made for departments at our institution that provide a home for multiple disciplines, such as Humanities and Social Science). An interdisciplinary course, by definition, has an interdisciplinary theme as its nucleus. In its essence, such a course brings the analytic methods of two or more disciplines to bear on a specific problem or question. Pedagogical strategies focus on inquiry or problem-based learning. The application of different methods and concepts is the key to assessing whether a course is or is not interdisciplinary. The term *interdisciplinary* is occasionally used to identify individual projects or assignments, but these often fall short in the necessary scope for learning experiences that demand in-depth exposure to the methodologies of distinct intellectual disciplines, and the creative application of these methodologies to specific problems.

Students taking interdisciplinary courses should be able to:

- Purposefully connect and integrate across-discipline knowledge and skills to solve problems;
- Synthesize and transfer knowledge across disciplinary boundaries;
- Comprehend factors inherent in complex problems;
- Apply integrative thinking to problem-solving in ethically and socially responsible ways;
- Recognize varied perspectives;
- Gain comfort with complexity and uncertainty;
- Think critically, communicate effectively, and work collaboratively; and
- Become flexible thinkers.

In this institutional context, we have developed an interdisciplinary course (Programming Narratives: Computer Animated Storytelling [PN]) designed to help non-computer systems major students develop computational thinking skills through computer programming combined with English writing skills to satisfy the college requirement of an interdisciplinary course. There is general agreement that computational thinking and writing are skills that every college student should have upon graduation. Many non-computer systems majors take the PS course to develop computational thinking and computer programming skills. However, if the PS course is challenging to computer systems major, it has proven even more challenging for non-majors. By incorporating many of the lessons that learned in our programming narratives LC, we expect that this interdisciplinary course will help non-computer systems major students in the development of problem-solving, computational thinking and writing skills.

The PN course combines the perspectives and methodologies of two academic disciplines, English and computing, in pursuit of a common goal. The common goal is to create a narrative-driven videogame prototype so students can identify with that which is immersive, engaging and rewarding. To complete this videogame prototype, students need the perspectives and

methodologies in two distinct academic disciplines, English and computing. Students will rely on the perspectives and methodologies learned in the English component of the course to develop a story. The reading of various kinds of short narratives will be very valuable in helping the students make the kinds of connections necessary to recognize synergies between writing stories and writing programs. Students will read, annotate, and discuss short narratives of various kinds (e.g., short stories, myth, plays, fantasy, horror, science fiction, historical fiction, and quest narratives as well as open-ended stories) and apply appropriate narrative structure to the construction of their video game prototype. As students study the structure of narratives and learn problem-solving strategies for writing, they are introduced to concepts of problem solving using constructs of logic inherent in computer programming languages. Students then implement the story as a computer program with the perspectives and methodologies learned in the computing component. Such gained knowledge will facilitate creative writing and the application of solutions to computer programming problems. The distinct perspectives and methodologies in English and computing are presented by two faculty members who are experts in their fields and will be co-teaching the course.

In this interdisciplinary course, students comprehend factors inherent in complex problems. Implementing a narrative written in English into a machine-executable computer program is a complex task. Students are challenged to map the structure of their narrative, including character and setting development, into constructs of logic inherent in computer programming languages. We expect these challenges will give students more insight into both their creative writing processes as well as their computer programming writing processes. Throughout the semester, students are challenged to understand, think critically to solve writing and computing problems, analyze narrative structure, compare and contrast stories, and apply various narrative structures to their project. Students work collaboratively on this group project to create a video game prototype and an accompanying game design document. The document will describe the project and discuss elements of analysis and design. Moreover, students will prepare and revise an annotated bibliography to facilitate their ability to make connections across academic disciplines. This strategy requires students to write one paragraph summarizing and assessing narrative structure and reflect on assigned course readings as they relate to interactive storytelling. In spring 2015, we offered two PN sections with approximately 40 students total.

4. Conclusions

Building on the success of our programming narratives LC, we have developed three different strategies integrating writing that incorporates narrative elements into problem-solving courses for computer systems majors and non-majors: (a) a module to develop narrative and writing skills for computer programming courses; (b) student-developed stories serving as contexts for computer programming courses; and (c) interdisciplinary creative writing and computational thinking course for non-majors. Such intentional interdisciplinary approaches to problem solving will promote learning transfer, providing students with the skills to succeed in college and beyond.

5. References

[1] Assessment and Institutional Research. (2010). *CUNY Student Experience Survey*. New York City College of Technology, CUNY.
[2] Barnett, S. & Ceci, S (2002). When and where do we apply what we learn? A taxonomy for far transfer. *Psychological Bulletin*, *128*(4), 612-637.
[3] Bateman, C. (Ed.). (2007). *Game writing: Narrative skills for videogames*. Boston: Charles River Media.
[4] Benander, R., & Lightner, R. (2005). Promoting transfer of learning: Connecting general education courses. *The Journal of General Education*, *54* (3), 199-208.
[5] Brooks, R. E. (1977). Towards a theory of the cognitive processes in computer programming. *International Journal of Man-Machine Studies*, 9, 737-751.
[6] Cabo, C., & Lansiquot, R. D. (2013). Development of interdisciplinary problem-solving strategies through games and computer simulations. In R. D. Lansiquot (Ed.) *Cases on interdisciplinary research trends in science, technology, engineering, and mathematics: Studies on urban classrooms* (pp. 268-294). New York: Information Science Reference.
[7] Cabo, C., & Lansiquot, R. D. (2014). Synergies between writing stories and writing programs in problem-solving courses. In *Proceedings of the 2014 IEEE Frontiers in Education Conference* (pp. 888-896). New York: IEEE.
[8] Campbell, J. (1949). *The hero with a thousand faces*. Navato, CA: New World Library.
[9] Cooper, S., Dann, W., & Pausch, R. (2000). Alice: A 3-D tool for introductory programming concepts. *Journal of Computing Sciences in Colleges*, *15*(5), 107-116.
[10] Cooper, S., Dann, W., & Pausch, R. (2003). Using animated 3D graphics to prepare novices for CS1. *Computer Science Education*, *13*(1), 3-30.
[11] Csikszentmihalyi, M. (2008). *Flow: The psychology of optimal experience*. New York: HarperCollins.
[12] Davies, S. P. (1993). Models and theories of programming strategy. International Journal of Man-Machine Studies, 39(2), 237-267
[13] DeCorte, E. (2003). Transfer as the productive use of acquired knowledge, skills, and motivations. *Current Directions in Psychological Science*, *12*(4), 142-146.
[14] Kuh, G. D. (2008). High-impact educational practices. Washington, D.C.: AAC&U.
[15] Johnson-Laird, P. N. (1983). *Mental models*. Cambridge: Cambridge University Press.
[16] Lansiquot, R., & Cabo, C. (2010). The narrative of computing. In *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2010* (pp. 3655-3660). Chesapeake, VA: AACE.
[17] Lansiquot, R. D., & Cabo, C. (2011). Alice's adventures in programming narratives. In C. Wankel & R. Hinrichs (Eds.), *Transforming virtual learning: Cutting-edge technologies in higher education* (Vol. 4, pp. 311-331). Bingley, UK: Emerald.
[18] Lansiquot, R., & Cabo, C. (2014). Strengthening the narrative of computing with learning communities. In *Proceedings of World Conference on Educational Media and Technology 2014*. Chesapeake, VA: AACE.
[19] Lansiquot, R. D., Satyanarayana, A., & Cabo, C. (2014). Using interdisciplinary game-based learning to develop problem solving and writing skills. In *Proceedings of the 121st ASEE Annual Conference*. Washington, DC: ASEE.
[20] Moskal, B., Lurie, D., & Cooper, S. (2004). Evaluating the effectiveness of a new instructional approach. *ACM SIGCSE Bulletin*, *36*(1), 75-79.
[21] Mullins, P., Whitfield, D., & Conlon, M. (2009). Using Alice 2.0 as a first language. *Journal of Computing Sciences in Colleges*, *24*(3), 136-143.
[22] Rist, R. S. (1995). Program structure and design. *Cognitive Science*, 19, 507-562.
[23] Spohrer, J. C., Soloway, E., & Pope, E. (1989). A goal/plan analysis of buggy Pascal programs. In E. Soloway and J. C. Spohrer (Eds.), *Studying the novice programmer* (pp. 355-399). Hillsdale, NJ: Lawrence Erlbaum.
[24] Sykes, E. R. (2007). Determining the effectiveness of the 3D Alice programming environment at the computer science I level. *Journal of Educational Computing Research*, *36*(2), 223-244.
[25] Vogler, C. (2007). *The writer's journey: Mythic structures for writers* (3rd ed.). Studio City, CA: Michael Wiese.

Acknowledgements

We would like thank Solanlly Hernandez, Tykila McCray, Noel Melendez, Andy Persaud, and Walter Rada for developing the stories that serve as the context of the case studies. We also

recognize Perkins Major Effort Component project, which is supported by the New York State Education Department (NYSED) Grant Number 51980-05-24.