I am motivated to teach for several reasons. As an interaction designer, I recognize that interaction designers should learn the technologies that underlie the system being designed. Without having exposure to the technology, an interaction designer’s role in the whole system development process will be limited due to communication issues with others, especially developers.

I have training in industrial design and visual communication design from a traditional design school, and have worked as an interaction designer for a software development company. I well recognize what interaction designers need to learn in school from these experiences. As a researcher in human-computer interaction and interaction design, I also recognize that learning design research methods and processes are not just for design students, but also for engineering students who want to build a system that interacts with humans, since design is a process of finding an optimal solution from a large problem space instead of focusing on how engineers usually solve a single problem. Lastly, as a student myself, I also recognize that my accomplishments are the direct result of the excellent teachers and advisors who have influenced me over the years. I would like to give the same experience to my future students.

My teaching philosophy and process mirrors my overall research, teaching and working experiences. I believe the following approaches are important:

**Provide design methods and experience**

In most interdisciplinary courses in the Interaction design or HCI area, we often see students with different backgrounds working together. I believe having students with various backgrounds in a class project is a great opportunity for students to gain different perspectives of the system to be designed.

The interdisciplinary courses often try to teach the process of solving systematic problems; these classes don’t just teach and improve student’s skill set or provide a solution to a single problem. The methods used in the interaction design and HCI area are very useful for understanding the whole system and the user of the system. For example, if a new music player has to be designed, it would be important to understand and imagine how this new product will be used, instead of focusing only on improving a navigational button. Using ethnographic design research methods, concept development, prototyping and iterative design, students learn how to initiate a design project, frame a problem using design re-
search and create a design solution through iterative design. Understanding of this process is critical to both design students and students lacking a background in design.

At Carnegie Mellon, I was a TA for the “Graduate Design Seminar II (51-702)” course. It is an interdisciplinary course that students from design, computer science, engineering, and cognitive science backgrounds attended. I observed that students benefited from collaboration with other students from different backgrounds. Students learn how to develop concepts through discussion, and how to divide tasks based on the background of each student during the project. They learn time and resource management skills and also communication skills.

**Provide understanding of technologies**

I believe students with non-technical background should understand how the system they are designing is being implemented. Understanding technology and developing implementation skills will not only extend student’s ability when designing a system, but will also open up communication with the engineers. Thus, I hope my students succeed in developing a level of implementation skills in my class. For this, I will introduce various prototyping tools and techniques to the class and let the students familiarize themselves with the underlying architecture and technology of the system they will design. For example, while a doctoral student at Carnegie Mellon, I taught “Introduction to Computing in Design (51-741),” a graduate level design course. In this course, I taught designers how to program user interfaces using Java programming language.

To teach design students a programming language, I believe a different approach than teaching traditional computer science is needed. Unlike computer science students, design students are more interested in building visual forms than dealing with text outputs in the console window. For example, in my class, when teaching “loops” or “conditional statements,” which are basic concepts in programming, students were assigned to create small graphic applications that generate an array of primitive visual objects on a canvas instead of printing out text lines.

In addition to programming basics, I also taught students how to implement various forms of user interface widgets such as buttons, panels, labels, and scroll bars, intended to help them understand how a modern user interface can be implemented. For the last project, students created a very simple functional drawing program – drawing lines, changing the width and color of the lines, erasing the lines on the screen and so on. From this exercise, students were able to learn the basic ideas underlying complex graphic programs such as Photoshop.

I believe these classes were valuable experiences for the design students in terms of understanding the underlying mechanisms of the user interface they use when they design.

From these perspectives, I’m interested in teaching several courses:

First, since I have trained as a designer, I am equipped to teach fundamental design courses. Also, I am interested in teaching courses that prepare students for the real-world practices of human-computer interaction and interaction design. Students in this course would receive an introduction to collecting user requirements, user research through ethnographic research, iterative design, prototyping a design, and testing the prototype with users. Another type of course could introduce various prototyping tech-
niques, with a focus on user interface programming and web development. During this class, students will develop skills to implement their design ideas.

At a higher course level, I would be most interested in teaching courses on situationally appropriate user interaction, and the method and process of human-computer interaction. In addition to this, I would be interested in teaching an advanced user interface prototyping course that covers user interface development for pervasive computing. Through these courses, students will learn research, design and implementation of advanced user interface development.