Teaching Statement
Nicolas Schiper

Who dares to teach must never cease to learn.

John Cotton Dana

I have always felt passion for solving problems and devising elegant as well as simple-to-understand solutions. I vividly remember my first programming project at EPFL: we were given a set of letters of the Scrabble® game and we were asked to write a program that would find a word with the maximum number of points as quickly as possible. I was excited to design precomputations on the English dictionary to make my program run faster. Later on, I felt the same passion when discovering the inherent difficulty of software-based data replication. From there, my passion to understand and build distributed systems only grew.

Teaching should not only transmit knowledge and explain how to use this knowledge but it should also communicate the passion we have for the subject. I believe this is one of the best ways to teach. I clearly remember the graduate distributed computing course I taught in the fall semester of 2009 at the University of Lugano. At the beginning of the first lecture, around twenty percent of the class showed little interest in the topic (fortunately for me at that time, a similar percentage seemed interested). After a few weeks of teaching, I told the students that I understood their difficulties and admitted that some of the topics taught were complex. I was surprised and happy to see that one of the skeptical student started asking questions. This definitely constituted an experience that I will not forget: engaging students in the class helped establish more interactivity, I could assess better which topics were well-understood and which needed clarification, and I hope, this made students integrate concepts better. The course was evaluated at the end of the semester and it received a score of more than 8 out of 10.

I recently came across Peer Instruction, a new methodology proposed by Professor Mazur of Harvard University, that aims at increasing the level of understanding of the lecture’s material. The methodology proposes to ask students to read the subject to be covered in class beforehand. The lecture is composed of several concept tests, or multiple choice questions, that students first try to answer individually. Students are then given the opportunity to confront their opinions. Interestingly, Professor Mazur mentions that students are better equipped to explain the solution to other students since they are closer to each other’s level of understanding than the professor. The teacher finally explains what the correct answer is and why. Given my previous experience on the benefits of interactivity, I would be interested in implementing this methodology in an introductory computer science course. Such an experience would certainly benefit both the students and the teacher.

During my doctoral studies at the University of Lugano, I performed the usual duties of a teaching assistant. I assisted classes on data structures and algorithms, operating systems, and distributed systems. I was also co-responsible for an operating systems laboratory where students were asked to build a database. These teaching duties enabled me to gain experience in designing assignments, exams, and projects.
Given my research background, I would be most interested in teaching a class on distributed systems. I believe that such a class should not only present fundamental concepts to understand and build distributed applications, but it should also expose students to modern datacenter technologies and foster their interest in the challenges presented by the cloud. My experience at Google could be beneficial to this course. In the spirit of Mazur’s methodology where students learn by doing, the course would include exercises testing their understanding of concepts, and projects asking students to build simple versions of a distributed protocol taught in class. Naturally, I would also be capable of teaching introductory computer science courses.