Teaching Portfolio

Fabrizio Montesi
Department of Mathematics and Computer Science (IMADA)
Computer Science
Postal address:
Campusvej 55
5230
Odense M
Denmark
Email: fmontesi@imada.sdu.dk
Phone: 65507171
Web address: http://www.fabriziomontesi.com/, http://www.fabriziomontesi.com/, http://www.fabriziomontesi.com/, http://www.fabriziomontesi.com/

Formal educational training

• 2015: Lecturer Training Programme (Pædagogikum), University of Southern Denmark, Denmark.
This is the standard training programme for lecturers at Danish universities (in the version offered at the University of Southern Denmark).
• 2011: Pedagogical Teaching Development (Ph.D. course), IT University of Copenhagen, Denmark.
An optional course that I followed during my Ph.D. studies at the IT University of Copenhagen, which involved studying pedagogical theories and teaching reflection.
• 2008: Team Building Course, I-TECH-OFF, Italy.
An obligatory course for the grant winners of the I-TECH-OFF business grant in 2008. In my industrial experience from 2008 to 2010, I was responsible for team building and employee training at italianaSoftware s.r.l., Italy.

Combining experiences in academia and industry widened my perspective on teaching, since I became an expert in the relationship between abstract models and practical problems. I pay attention to teaching students how to choose between different models and tools (or even create new ones), in particular by discussing the implications of different choices on concrete project requirements.

Administrative tasks related to education

• 2018: “Concurrency and Logic” Specialisation for the M.Sc. degree in Computer Science.
• 2017–present: Head of the SDU Talent Fellowship in Computer Science.
• 2017–2018: Reform of the B.Sc. Degree in Computer Science.
Together with my colleagues, I participated in the redesign of the B.Sc. study programme for Computer Science. My main administrative contribution was taking care of the reform of the course in Concurrent Programming.
• 2017: Organiser of the team from the University of Southern Denmark for the Deloitte Hackathon in Copenhagen.
• 2016: Organiser of the team from the University of Southern Denmark for the Deloitte Hackathon in Copenhagen.
• 2016: Judge for the SDU Supercomputer Challenge at the University of Southern Denmark.

Teaching experience

Selected courses from the last 6 years:

• Concurrency Theory, M.Sc. course, Course Responsible, 10 ECTS. University of Southern Denmark, 2019.
• Concurrent Programming, B.Sc. course, Course Responsible, 10 ECTS. University of Southern Denmark, 2019.
• Concurrency Theory, M.Sc. course, Course Responsible, 10 ECTS. University of Southern Denmark, 2018.
• Concurrent Programming, B.Sc. course, Course Responsible, 5 ECTS. University of Southern Denmark, 2018.
• Concurrency Theory, M.Sc. course, Course Responsible, 10 ECTS. University of Southern Denmark, 2017.
• Concurrent Programming, B.Sc. course, Course Responsible, 5 ECTS. University of Southern Denmark, 2017.
• Microservice Programming, M.Sc. course, Course Responsible, 10 ECTS. University of Southern Denmark, 2016.
• Concurrent Programming, B.Sc. course, Course Responsible, 5 ECTS. University of Southern Denmark, 2016.
• Integrated Mobile and Distributed Systems, M.Sc. course, Course Responsible, 10 ECTS. University of Southern Denmark, 2014.

Methods, materials, and tools

Assessment methods

I have experience with all standard examination forms: written exams, oral exams, projects, and reports. I also used obligatory assignments during courses. For example, I used written assignments in the courses Introduction to Computer Science and Concurrency Theory, while in Microservice Programming, each student has to read a scientific or technology article and then prepare and give a formal presentation in class about it during the course. These presentations are held in

a conference style, sometimes with explicit student groups that are given defending and attacking positions.

Original material production

During my career I developed a series of original material. Most of it is available through my teaching webpage (https://fabriziomontesi.com/teaching.html). Some interesting items:
• The first (to the best of my knowledge) academic course material on microservices (rubrics, slides, and notes) and the first course material on the Jolie programming language (updated version at https://fabriziomontesi.com/teaching/mp-2016/index.html). This material (or parts of it) has been adopted by other lecturers at other universities. For example, it was used in 2016 in the Operating Systems course at the University of Bologna (where Jolie is used to teach some aspects of concurrency). Examples of other universities that used this material are: the University of Padova, the IT University of Copenhagen, and the University of Udine.
• The first lecture notes designed for M.Sc. courses on how to formally reason about concurrent communication protocols using choreographies (“Introduction to Choreographies”).
• A rubric for helping students in evaluating presentations and preparing their own.
• A series of original examples and exercises for learning concurrent programming and microservice programming (https://github.com/mp2016).
• Guideline documents for writing final reports.

Original tool production

I make ample use of live coding in my lectures, where I demonstrate and discuss with students various concrete implementations. To support these lectures, and also for helping students in their exercises, I use two tools that I created for giving online feedback (meaning that the feedback is given immediately, while the students write the code):
• A plugin for developing Jolie code in the Atom code editor.
• An extension of Eclipse (a widespread IDE) for developing choreographies (first of its kind). This IDE is used by students to write descriptions of communicating systems. It supports syntax highlighting, notification of errors in protocol implementations and their composition, and a tool for generating executable code that lets students try their designs.

Other tools

These are some standard tools that I use in my teaching:
• Student response systems, like Socrative and Poll Everywhere. For example, I use them for one-minute-papers at the end of my lectures and also for quizzes in class.
• GitHub. I use this system to share some of the material among students. This enables the students themselves to contribute to the material and the exercises (a sort of Wiki technology for code, if you like), making them active and giving me precise feedback on the changes that they propose. Students use this system also for online discussions on the course; GitHub gives them the ability of referring to precise pieces of the material (e.g., code lines), and then provides a web interface to visualise such activities.
• Wikis.

Teaching philosophy and practice

My teaching philosophy is rooted on two main determinants:• The recognition that the environment in which students carry out their learning is a fundamental factor for their development.• The combination of my past experiences in academia and industry.These two elements drove my teaching philosophy towards a combination of a) a careful tuning of practice towards the establishment of an environment that fosters learning and b) research-based teaching. This is all tied by my interest in group dynamics.EnvironmentWhat is a good learning environment, and how much does it influence our students’ ability to learn?To investigate this question, we must first cast off the idea that the intelligence of an individual can be measured in absolute and independent terms. On the contrary, an increasing body of research backs the hypothesis that intelligence is always situational. Our ability to think changes depending on the context that we are put in and, more specifically, on our emotional state. Some may find the importance of this idea difficult to accept. After all, academics are grown in a tradition of intellectual excellence and individualism, a message that also appears repeatedly in Western culture–where we evaluate the individual and the individual’s ability to control herself, much more often than we evaluatepeople in groups. This tradition can easily bring us to believe that our intellect can perform independently of our emotions (or unless such emotions are intense, such as grief), detaching us from the notion that intelligence is actually embodied all the time and it can change noticeably even during the course of a class [Berg and Seeber, 2016]. A manifesto from 2004 by experts at MIT Media Lab recognises that “affective functions and cognitive ones are inextricably integrated with one another”. They comment that even a slight positive mood induces a different kind of thinking, improving our creativity and flexibility in problem solving. This connects nicely to Frederickson’s observations on group dynamics and the “broaden and build” theory of positive emotions. For example, joy broadens by urging us to play, explore, push our limits, be creative, take in new information, and ultimately expand the self. We can then conclude that we, as teachers, should invest considerable effort in thinking about the emotional state of students in their learning. As the aforementioned manifest by the MIT Media group experts states, we should devote at least as much time to this goal as

we do for the cognitive and informational goals. Thus, we should pay particular attention to the emotional environment that we build. This may appear in contrast with the academic rigor that we expect in higher education, but it is not. If anything, we can expect students to get more out of their time inacademia if we succeed in making them more creative and investigative through emotion.Where and when do we start building such emotional environment? In live lectures. Live lectures are key to the transmission of emotion, which is essential to this process. There are many positive emotions that our students may benefit from.An exampleI give an example (a case study, if you like) of how my methodology can be concretely applied to a course.Motivation. I first describe the motivation underlying my example.My field of interest is that of distributed software running, e.g., on the Internet or other large networks. With the extreme rise of the interest in this field in the last two decades, there is a tendency of periodically creating new technologies to address different problems or to address the same problem in a better way. New problems are themselves discovered very often as the field develops, every year or more. As a consequence, companies which want to be successful have to develop ways of keeping up with the continuous technological changes. The fast pace at which these changes are made and commercialised elicits the importance of a particular set of skills: that of analysing, choosing, and creating new technologies. This is in contrast with more established fields where for many problems addressed by the industry there are solid foundations standing on many years of research. For example, the technologies of Web Services, something upon which we all depend on every day as a society, are still very much in flux and it is not uncommon to see big projects carried out with entire technologies developed just a year before. Even more, sometimes such technologies do not exist and have to be developed purposefully for a given problem.Application with research-based teaching. Observe that the professional skills discussed in the motivations are akin to that of a researcher in Computer Science. In particular, it is not sufficient for students to develop skills that go deep in the details of existing models, but also to develop skills that enable them to assimilate, understand, and discuss new ideas very quickly. Towards this aim, my teaching philosophy borrows much from the methodology of research-based teaching. In the following, I exemplify how I apply this into practice.Microservice Programming (MP) is a new elective course that I designed for master students in Computer Science at the University of Southern Denmark. The course is about advanced topics on distributed systems and software integration. In this setting, I see my main role as teacher to be a guide towards reaching independence and leadership in the world of research and its application. The course focuses on explaining what goes on behind the scenes in the R&D of new technologies in the field, with the objective of enabling the students to recreate this process later on in their work after the degree (which is what the leading companies need).I use my experience as the creator of a technology to show, in class with live demonstration, the crucial steps of how that technology was created and why. I periodically make the students discuss (among them and with me) the different choices made. The students are asked to propose different paths that may have been followed in the development of the technology, or alternative ones, and I show them the consequences (comparing them with what the students predicted). Discussions can either be open or mixed with Think-Pair-Share sessions. One key development here is to show the errors and the uncertainties that lie in the path towards the creation of a technology. These canlead to negative emotions, so if the students are not guided in dealing with them they may become less responsive and creative than we wish for. Hence, I make classes personal. I discuss the problems that I know researchers (including myself) had to deal with in reaching their results, and I encourage students to discuss these steps together.I purposefully insert some anecdotes about researchers I know, perhaps even give them a face, to make them appear more familiar and less like a distant figure. Sometimes I report on discussions I had with my colleagues before reaching a solution. These sessions are conducted in a relaxed and informal way, to establish a learning environment where students can feel free to share wrong ideas. Building on this methodology, I introduce a twist to kickstart positive emotions. I perform live demonstrations of the consequences of the ideas discussed in class. As students get progressively nearer to the solution—perhaps with some help of the teacher sometimes, using other techniques for active learning not discussed here—this starts causing a shared sentiment of excitement. A key objective for me here is that students start enjoying the ideas that they discuss and that they start feeling a sense of belonging to the course. Enjoyment is thus an activator towards the end of reaching affection, one of the cornerstones of active learning. When affection is reached, there is no need to maintain a status of “hype” in the students. We have established the environment that we want, and we can proceed in the rest of the course building on that to tackle difficult learning objectives.The live lectures described above are intended to make students take the first step in a journey of discovery, guided by the teacher. I then progressively make my guidance more and more subtle, with the aim of letting students developing their independence (recall the motivations behind the course). For example, I give lectures on how to study R&D papers, and how to present their motivations and results. This is in preparation to reverse the roles and enable students to understand and explain the course content by themselves. The students then have to choose a research paper and prepare a presentation for the class, enabling discussion of many different technologies. A main task givento the presenter is to prepare the presentation such that the audience can understand it without having studied the paper, but only read it, but without losing the crucial technical aspects of the material. Some students may get stressed by this kind of tasks, or feel peer pressure negatively. Consequently, this activity starts only after the emotional environment reached a certain level of openness and gives me the opportunity to discuss how to deal with stress in presentations.These activities are fundamental to remove the psychological barrier that a student may have towards complex material from advanced recent R&D, i.e., not feeling prepared or “smart” enough to quickly understand the main aspects of difficult issues. I find the theories of cognitive dissonance and mental self-government particularly useful in dealing with this barrier, which is in my opinion one of the main obstacles in educating a student into becoming a potential leader in the field (achieving professional excellence). In my industrial experience, I observed that empowered and confident professionals often dare trying more revolutionary approaches and ultimately achieve more easily results that benefit the entire workgroup they participate in.In the last part of the course, the students have to do their own R&D, both in groups and individually, using bleeding-edge research results from recent years. Here, my role has now completely switched to a support role in which I help students in discussing their ideas, but they have completely taken the lead in what they do. A key technique here is the use of online sharing platforms for software source code, which the students use to publish their projects. Together with the student

presentations, which are also publicly shared, this enables a sense of ownership that motivates them towards a deep exploration of their project ideas.Exemplary practiceThis example comes from a lecture in the course Microservice Programming (2016), a 10 ECTS elective course in Computer Science at IMADA (SDU), dated 5 May 2016. The external supervisor of my Lecturer Training Programme and a peer in my learning group for the same programme were present.The assessment form for the course is a software project, accompanied by a written report. A distinctive point of this course is that the problem for the project is not given by the teacher. Rather, students must come up with the problems that they respectively want to address in their projects to pass the exam. Therefore, students participate to the assessment (self-assessment). To do that effectively, they need to cross over to the teacher role and acquire skills such as the design and evaluation of the assessment process. To develop the necessary critical thinking and activate creativity, a key activity in the course is student presentations. In this practice in particular, a student was presenting the problem and results of a research paper in the field of the course. The idea is that each student should present a paper or a technology, such that then the entire group can have a discussion and develop critical thinking on ideas (those of the paper), which is what they need to judge if their own project ideas for passing the course are appropriate. (This is not the only activity towards that end, since it would not be sufficient, but discussing the other activities goes out of the scope of this text.)In this particular lecture, a student was presenting and the presentation was good since the beginning, but the final part positively exceeded my expectations. The paper was a study that identified security issues in web services. In the last part of the presentation, the student had a slide showing a table with two columns. On the left, the problems identified in the paper, which she discussed previously one by one; on the right, an empty column. The student intended to fill the empty column cell by cell, by stating the issue at hand (from the cell on the left) and then revealing the solution proposed by the paper authors on the right.At this point I stepped in (reflection in action) and asked the presenting student to wait. Instead of revealing the solution herself for each issue, why not asking the audience (the other students) for it? The presentation was perfectly set up to perform this exercise together, which in my mind would have been a far more engaging activity than just revealing the solutions directly. The audience managed to discuss and propose the solutions to the first three problems. Time was up then, and I announced that we would proceed with this exercise in the next lecture.Students were really activated by this exercise: engagement was very high. It is an exercise that came to my mind only when I saw the last slide by the presenting student, so I had to react quickly.A comment that I received from the observers (external supervisor and group peer) was that the presenting student may have been surprised by my intervention and this may cause unpredictable emotions. Also, they commented that I could consider giving students even more time for thinking about their answers. I agreed. For this reason, I had a meeting with the student presenter where I explained my reasons for interrupting her presentation and congratulated her for the good job in preparing the presentation. I then tasked her to conduct the first part of the following lecture,which I organised with a special structure, by finishing her presentation but continuing with the exercise I started in the previous lecture. That class went really well. I organised it as follows. For each security issue:• The student presenter recaps the security issue.• The teacher writes a software that can be attacked by the students exploiting that security issue.• While the teacher writes the software to be attacked, the students (in groups) write programs to attack the softwarewritten by the teacher, trying to exploit the security issue.• The teacher executes the attacks and shows the results to everybody using a projector.• All the code developed by the teacher and the students was shared in the class.This was the first time that a student inspired me to apply such a big change to what I wanted to do in a lecture. I was excited and dived right into it. The students were very happy to do it. I think that they felt that the course is collaborative and that it proceeds as a discussion where they can share their ideas and focus on their interests. The feedback I got in the course confirms this.

Student evaluations

Students typically leave excellent course evaluations and free-form comments, which can be provided on request.