

Undervisning CV

Personal details

- First name and surname: Ulrik P. Schultz
- Address: Campusvej 55, 5230 Odense M (work); Svendborgvej 12A, 5762 Vester Skerninge (home)
- Email: ups@mmmi.sdu.dk; Phone: +45 65 50 35 70

Education

- December 2000. Ph.D. in Computer Science from University of Rennes I, France.
- June 1997. Master of Science in Computer Science from University of Aarhus, Denmark.

Positions held

- From-To (Position): 01/09/2005— Associate Professor, Maersk Institute, University of Southern Denmark. Main teaching areas: software engineering, embedded systems, programming languages.
- From-To (Position): 01/09/2000—01/09/2005 Assistant Professor, Computer Science, University of Aarhus. Main teaching areas: object-oriented programming, pervasive computing, software engineering, programming languages.

1 Teaching philosophy

My approach to and motivation for teaching is captured by the following quote:

Always be a first-rate version of yourself, instead of a second-rate version of somebody else. —Judy Garland

I always strive to communicate the subject being taught in a personal and lively way, based on my own interpretation of the subject, in combination with appropriate material grounded in relevant scientific literature. I carefully motivate why each topic is included, and I have high expectations for how much the students learn. Although I strive for friendly interactions with students, I generally adhere to an underlying principle of “tough love” where students are given difficult challenges and praise must be earned. Across all of the courses that I have taught (see part 2.1), I have consistently and without exceptions received positive oral and/or written evaluations from a significant majority of the students. I believe the key to good teaching is in effective two-way communication. As a teacher I must adapt my communication to the specific topic, the specific students, and the specific interaction we are having at a given point in time. Interactivity is key to this communication, to lift students I must meet them where they are, and this is only possible if I communicate effectively with them.

I embrace the use of new technologies, new techniques, and the inclusion of new subject matters in the courses that I teach. New technologies often provide opportunities for enhancing various aspects of the way I teach, although they never change the fundamental principles. New techniques provide opportunities for new forms of interaction and for new ways of structuring the learning process, which can serve to complement the basic communication with the students. I enjoy discussing teaching issues with my colleagues, both issues specific to selected courses and general pedagogical issues. I see good teaching as a continuous evolution over the years, both in terms of content and the form.

2 Teaching experience

2.1 Courses taught

I have taught one software bachelor course and 18 master courses in computer science, robotics and software engineering. I have often defined the course contents and learning goals. Without any exceptions, I have always defined the specific contents and the curriculum, and I have always created my own teaching materials (e.g., slides).

- Model-Driven Software Development Project (MDSD-P), taught spring 2019, Software Engineering MSc course at MMMI, University of Southern Denmark. Material covered: metamodeling, domain-specific languages, automatic program generation, parsing, model-driven software development, language design and implementation.
- Model-Driven Software Development (MDSD), taught spring 2017, 2018, Software Engineering MSc course at MMMI, University of Southern Denmark. Material covered: metamodeling, domain-specific languages, automatic program generation, parsing, model-driven software development.
- Distribution and Integration Technologies (TEC), taught spring 2016, Software Engineering MSc course at MMMI, University of Southern Denmark. Material covered: web-based systems, scalability, virtualization.
- Software Customization (SWC), taught spring 2014, 2015, 2016, Software Engineering MSc course at MMMI, University of Southern Denmark. Material covered: domain-specific languages, automatic program generation, parsing, model-driven software development.
- Operating Systems and Networks (OPN), taught fall 2013, 2014, 2015, 2016, 2017, Software Engineering BSc course at MMMI, University of Southern Denmark. Material covered: operating systems, virtual machines, networks, socket- and object-based communication.
- Programmeringssprog, beskrivelse og udførelse (SSE02), taught spring 2012, 2013, Robotics MSc course at MMMI, University of Southern Denmark. Material covered: general programming languages principles, domain-specific languages, automatic program generation, agile software development.
- Embedded Software Design (ESD), taught spring 2011, Mechatronics and Robotics MSc course remote in parallel at MCI and MMMI, University of Southern Denmark. Material covered: hierarchical state machines, automatic code generation, domain-specific languages.
- Scientific Methods (SCM), taught fall 2009, 2010, Robotics MSc course at MMMI, University of Southern Denmark.

Material covered: empirical science, scientific methods, statistics, reading and writing academic papers.

- Software Aspects of Embedded Systems (EMB4) / Distributed Embedded Systems (DES), co-taught fall 2010, 2011, taught fall 2012, Mechatronics and MSc course remote in parallel at MCI and MIMI, University of Southern Denmark. Material covered: operating systems, device drivers, real-time programming, distributed embedded systems, automatic program generation.
 - Advanced Topics in Software Engineering (SSE05), taught spring 2008, Robotics MSc course at MIMI, University of Southern Denmark. Material covered: advanced topics in programming languages and frameworks, empirical methods.
 - Model-based Processes and Systems (SSE04), taught spring 2006, 2007, 2008, 2009 (fourth quarter), Computer Systems Engineering MSc course at MIMI, University of Southern Denmark. Material covered: model-driven software development, agile development processes, domain-specific languages, code generation.
 - Introduction to Software Systems Engineering (SSE01), taught fall 2005, 2006, 2007, 2008 (first quarter), Computer Systems Engineering MSc course at MIMI, University of Southern Denmark. Material covered: object-oriented programming, unified process, unified modelling language, mobile agents.
 - Aspects of Object-Oriented Programming (dAOP), co-taught fall 2001, 2004 (second quarter), Computer Science MSc course at CS, University of Aarhus, Denmark. Material covered: virtual machines, static and dynamic optimizations.
 - Software Patterns, taught fall 2004 (first quarter), Computer Science MSc course at CS, University of Aarhus, Denmark. Material covered: design patterns, architectural patterns, frameworks, antipatterns, and various other kinds of patterns. (At the time the largest elective graduate course to date at the Computer Science Department, with over 100 students.)
 - Partial Evaluation for Imperative and Object-Oriented Languages, taught spring 2004 (fourth quarter), Computer Science MSc course at CS, University of Aarhus, Denmark. Material covered: generative programming, two-level languages, binding-time analysis, applications of partial evaluation.
 - Pervasive Computing (PvC), co-taught fall semester 2003, Computer Science MSc course at CS, University of Aarhus, Denmark. Material covered: technologies and issues in pervasive computing, context awareness, software mobility, embedded systems.
 - Patterns and Software Architecture (PASA), taught spring 2001, co-taught spring 2002, 2003, Computer Science MSc course at CS, University of Aarhus, Denmark. Material covered: general introduction to software patterns, design patterns, frameworks, and software components.
 - Distributed Objects and Pervasive Computing (DOPC), co-taught fall 2002, Computer Science MSc course at CS, University of Aarhus, Denmark. Material covered: Java RMI, CORBA, J2EE, overview of topics in pervasive computing.
 - Design and Implementation of Object-Oriented Virtual Machines (OO-VM), co-taught spring 2002, Computer Science MSc course at CS, University of Aarhus, Denmark. Material covered: virtual machines, advanced dynamic optimizations.
- In addition to the practical experience listed above, I have followed the two-day course in university pedagogy for assistant professors (adjunktprædagogisk kursus) offered by the University of Aarhus, summer 2001.

2.2 Student supervision (not PhD)

- I have supervised 58 MSc students in computer science, computers systems engineering and robotics. In addition, I have supervised 4 bachelor students in robotics.
- Since 2005: supervisor for 42 MSc students at MIMI, 4 in collaboration with industrial partners, as well as 4 bachelor students.
- 2000-2005: supervisor for 14 MSc students at Computer Science, University of Aarhus, 1 in collaboration with an industrial partner, 3 of which resulted in joint publications at international conferences.
- 1998-2000: supervisor for 2 Maitrise (fourth year) and a DEA (PhD preparatory) student at IRISA, Université de Rennes I, France.

2.3 Pedagogical accomplishments

I have made significant contributions to the development of education programs in software engineering and robotics, and I have been key in the technological and pedagogical development of a remote teaching system.

- Key person in the design of the new software engineering education, bachelor and master (2013- . . .): significant contributions to all aspects of the bachelor education, key contributions to all aspects of the master education.
- Key person in the overall design of the robotics master education (2009-2013): significant contributions to the overall structure of the education and the content of specific courses.
- Key person in designing the requirements for and pioneering the use of a parallel remote-teaching system between Odense and Sønderborg: developed pedagogical approach that enabled successful simultaneous teaching of courses for students across Robotics and Mechatronics education programs.

I remain a member of the education advisory committees for both the robotics and software educations, and thus I continuously provide input to the continued evolution of these two educations.

As member (previously Chair) of IFIP Working Group 2.11 on Program Generation (Chair 2013–2019, member since 2008) I regularly meet with world-leading researchers in program generation. Teaching is regularly discussed at our meetings, providing a unique opportunity for both improving and communicating the way I teach advanced topics in software engineering.