SWEN40004 Modelling Complex Software Systems

Credit Points:	12.5
Level:	4 (Undergraduate)
Dates & Locations:	2015, Parkville This subject commences in the following study period/s: Semester 1, Parkville - Taught on campus.
Time Commitment:	Contact Hours: 36 hours, 3 hours per week. Total Time Commitment: 170 hours
Prerequisites:	**One of the following:** **Subject** / **Study Period Commencement:** / **Credit Points:** COMP20004 Discrete Structures — Not offered 2015 — 12.50 COMP30026 Models of Computation — Semester 2 — 12.50 COMP30025 Theory of Computation — Not offered 2015 — 12.50 AND **One of the following:** **Subject** / **Study Period Commencement:** / **Credit Points:** COMP20003 Algorithms and Data Structures — Semester 2 — 12.50 COMP90038 Algorithms and Complexity — Semester 1, Semester 2 — 12.50 OR 433-253 Algorithms & Data Structures
Corequisites:	None
Recommended Background Knowledge:	None
Non Allowed Subjects:	Students cannot enrol in and gain credit for this subject and: 433-441 Systems Modelling and Analysis 433-641 Systems Modelling and Analysis
Core Participation Requirements:	For the purposes of considering request for Reasonable Adjustments under the Disability Standards for Education (Cwth 2005), and Student Support and Engagement Policy, academic requirements for this subject are articulated in the Subject Overview, Learning Outcomes, Assessment and Generic Skills sections of this entry. It is University policy to take all reasonable steps to minimise the impact of disability upon academic study, and reasonable adjustments will be made to enhance a student's participation in the University's programs. Students who feel their disability may impact on meeting the requirements of this subject are encouraged to discuss this matter with a Faculty Student Adviser and Student Equity and Disability Support: http://services.unimelb.edu.au/disability
Coordinator:	Assoc Prof Harald Sondergaard
Contact:	email: **harald@unimelb.edu.au (mailto:harald@unimelb.edu.au)**

Subject Overview:	**AIMS** Mathematical modelling is important for understanding and engineering many facets of digital complex systems. The aim of this subject is for students to understand the range and use of mathematical theories and notations in the analysis of discrete systems, how to abstract the key aspects of a problem into a model to handle complexity, and how models can be employed to verify large-scale complex software systems. **INDICATIVE CONTENT** Topics covered will be selected from: concurrent systems modelling, analysis and implementation; process algebra; temporal logic and model checking; probability and stochastic modelling; dynamical systems; cellular automata; agent-based modelling; simulation and analysis of complex systems.
Learning Outcomes:	**INTENDED LEARNING OUTCOMES (ILO)** On completion of this subject the student is expected to: 1 Select from a range of techniques with which to model discrete systems 2 Select analysis techniques and perform analysis on discrete systems 3 Have the theoretical basis for understanding modern analytical techniques and the skill to solve problems using those techniques 4 Abstract the key aspects of a complex system into a mathematical model for analysis and verification
Assessment:	Project work during semester comprising: One assignment (in two parts) requiring approximately 25- 30 hours of work (25%). The assignment is due in weeks 4 (first part) and 7 (second part) One research project executed in pairs, consisting of a report of no more than 1500 words, requiring approximately 27 - 32 hours of work (25%). The project is due in week 11 One two-hour end-of-semester written examination (50%). Hurdle requirement: To pass the subject, student must obtain: at least 50% overall; at least 50% (25/50) in project work; and at least 50% (25/50) in the written examination. Intended Learning Outcomes (ILOs) 1 to 4 are addressed in the examination ILOs 2, 3, ad 4 are addressed in the assignments, and the pair research project Generic skills are addressed by all assessment items
Prescribed Texts:	None
Breadth Options:	This subject is not available as a breadth subject.
Fees Information:	Subject EFTSL, Level, Discipline & Census Date, http://enrolment.unimelb.edu.au/fees
Generic Skills:	On completion of this subject, students should have the following skills. # Ability to undertake problem identification, formulation and solution # Ability to utilise a systems approach to analysing software properties # Capacity for independent critical analysis of models, and self-directed research for mathematical modelling approaches # Intellectual curiosity and creativity, including understanding of the philosophical and methodological ideas behind research in software systems analysis # Openness to new ideas and unconventional critiques of received wisdom.
Notes:	**LEARNING AND TEACHING METHODS** The subject will be delivered through a combination of lectures, hands-on workshops, individual assignments, and a pair-based project in which students use modelling and simulation to study a complex system. **INDICATIVE KEY LEARNING RESOURCES** A package of notes will be made available to the students at the start of the course. An addition reference is: Kramer, Jeff, and Jeff Magee: Concurrency: State Models and Java Programs, *John Wiley and Sons, 2nd edition* (2006). **CAREERS / INDUSTRY LINKS**

	The ability for software engineers and computer scientists to abstract and analyse complex problems is key to their profession. As software systems continue to be deployed in increasingly complex and critical environments, such as transport control, manufacturing, and healthcare, the tools and methods for analysing complex systems will become more important.
Related Course(s):	Master of Information Technology Master of Information Technology
Related Majors/Minors/ Specialisations:	B-ENG Software Engineering stream MIT Computing Specialisation Master of Engineering (Software with Business) Master of Engineering (Software)