Università degli Studi di L'Aquila - Dipartimento di Ingegneria e Scienze dell'Informazione e Matematica

Course catalogue

Page compiled on 13/08/2020

Programme of Integrated course "Autonomous Software & Service Engineering"		
This course is composed of 2 Modules: 1) Service - Oriented Software Engineering, 2) Software Engineering for Autonomous Systems		
Programme of Module "Service - Oriented Software Engineering"		
• Code: DT0203 • Type of course unit: Elective (Master Degree in Computer Science curriculum GSEEM), Elective (Master Degree in Computer Science curriculum NEDAS), Compulsory (Master Degree in Computer Science curriculum SEAS), Elective (Master Degree in Computer Science curriculum UBIDIS) • Level of course unit: Postgraduate Degrees • Semester: 1		
Number of ects credits: (Master Degree in Computer Science) 6 (workload 150 hours)		
Teachers: Marco Autili (marco.autili@univaq.it)		
1	**Course objectives**	The objective of this course is to introduce Service-oriented Architecture (SOA) as a way of building distributed applications using Web Services (WS). The course aims at deepening the understanding of key aspects and principles of SOA and WS technologies, as well as related Software Engineering methodologies. At the end of the course the students will: (i) understand the notions of WSs, WS standards, and SOA; (ii) understand the service engineering development processes that can be followed to realize reusable and flexible WSs; (iii) understand how business process models and service-oriented programming models can be used as a basis for the design and the implementation of service-oriented systems; (iv) be familiar with a number of frameworks that support the development and deployment of service-oriented applications, both WS clients and WS providers; (v) have been introduced to the notion of service composition as a means for developing complex service-oriented applications.
2	**Course content and learning outcomes (dublin descriptors)**	Topics of the module include: • Reuse-based System Development • Component-based System Development • Service Oriented Architecture (SOA) • Service-oriented Distributed System Development (WEB Services, REST Services, Microservices, Data as a Service (DaaS), Load Balancer) • SOA-enabling Technologies (XML, SOAP, WSDL, WADL, etc) • SOA-supporting Frameworks (ANT, MAVEN, SPRING WS, JAX WS, APACHE AXIS, APACHE CXF, etc.) • Service Composition • Numerous Practical Sessions in Classroom On successful completion of this module, the student should : • have profound knowledge of the notions of Web Services, Web Service standards, and Service-oriented Architecture (SOA); • have knowledge and understanding of the service engineering development processes that can be followed to realize reusable and flexible Web Services; • analyse and discuss the Service-oriented Architecture principles and the related engineering development processes; • be familiar with a number of frameworks that support the development and deployment of service-oriented applications, both WS clients and WS providers; • explain and illustrate how business process models and service-oriented programming models can be used as a basis for the design and the implementation of service-oriented systems;

		• explain and illustrate the notion of service composition as a means for developing complex service-oriented applications; • demonstrate skill in problem-solving, demonstrate ability to use (subset of) the programming framework to realize service-oriented applications; • demonstrate capacity of abstraction and modularity when designing service-oriented applications.
3	**Course prerequisites**	Object-oriented programming, Java language, and XML basics.
4	**Teaching methodsand language**	Lectures and practical exercises **Language**: English **Reference textbooks** • Ian Sommerville, ***Software Engineering***. (vol. 10th Edition) 2016. • Michael P. Papazoglou, ***Web Services & SOA, Principles and Technology***. (vol. 2nd Edition) 2012.
5	**Assessment methods**	MID-TERM TEST (OPTIONAL - to be decided during the course): either students will be assigned a homework concerning the course topics (tutorial preparation, lecture simulation, small project, ...), or students will be asked questions concerning the Software Engineering aspects learned during the first lessons including all topics of PART I and SOA Principles of PART II. FINAL TEST: Students will be given the specification of a Service-oriented System to be implemented by applying the learned service-oriented engineering principles and methods, and by using the SOA enabling technologies and the Java frameworks taught and demonstrated during the classroom activities. Students will present the system and discuss the way it has been realized. Contextually, if the mid-term test was not given or was not passed, students will be asked questions concerning the Software Engineering aspects learned during the first lessons including all topics of PART I and SOA Principles of PART II.

Programme of Module "Software Engineering for Autonomous Systems"

- Code: DT0227
- Type of course unit: Elective (Master Degree in Computer Science curriculum GSEEM), Elective (Master Degree in Computer Science curriculum NEDAS), Elective (Master Degree in Computer Science curriculum SEAS), Elective (Master Degree in Computer Science curriculum UBIDIS)
- Level of course unit: Postgraduate Degrees
- Semester: 1

Number of ects credits: (Master Degree in Computer Science) 6 (workload 150 hours)

Teachers: Davide Di Ruscio (Davide.Diruscio@univaq.it)

1	**Course objectives**	Systems that can change their behaviour in response to unanticipated events during operation are called “autonomous”. Unlike more traditional systems that have predefined purposes, autonomous systems are able to tailor their behaviour and operations in accordance with the circumstances they find. Engineering autonomous systems is a challenging task involving several theoretical foundations and application fields (e.g., self-adaptiveness, machine learning, sensor networks, control engineering, and artificial intelligence). This course aims at introducing the fundamental concepts related to the development of autonomous systems from a software engineering perspective. Various methods and techniques currently applied in the design of autonomous systems are shown. Self-* attributes of autonomous systems, architectures, models, and languages are presented in order to show the technical viability of systems that can dynamically adapt their behaviour to varying operating conditions, delivering the appropriate application level response under these different conditions. Concrete examples of autonomous systems in the domains of Internet of Things, Cyber-Physical Systems, and unmanned vehicles are given.
2	**Course content and learning outcomes (dublin**	Topics of the module include: • Autonomous systems • Self-adaptive systems

	descriptors)	• Software Engineering • Model-Driven Engineering • Models@runtime • Internet of Things • Cyber-Physical Systems • Unmanned Vehicles On successful completion of this module, the student should : • On successful completion of this module, the student should: • Be able to use modeling languages to model autonomous systems. • Be able to adopt model-driven techniques and tools to engineer autonomous systems. • Be able to apply knowledge of computing and mathematics appropriate to the discipline. • Be able to analyze a problem, identify and define the computing requirements appropriate to its solution. • Be able to function effectively on teams to accomplish a common goal. • Be able to communicate effectively with a range of audiences. • Be able to analyze the impact of computing on individuals, organizations and society, including ethical, legal, security and global policy issues. • Be able to use current techniques, skills, and tools necessary for computing practice. • Have developed practical skills that can be transferred into a real-world environment.
3	**Course prerequisites**	
4	**Teaching methodsand language**	Lectures, exercises - Scientific papers and other references provided during the course. **Language**: English
5	**Assessment methods**	- Pre-Assessment: There is no formal pre-assessment, but Course pre-requisites are clearly stated on the Module website. Fulfilment of such pre-requisites is verified by formative assessment. - Formative Assessment: The formative assessment is performed via interactive interaction between teacher and students during lectures. Students are aware since the beginning of the Course that they will be involved (in turns) in: - Questioning and discussion, by means of open oral questions to the class or to single students. - Summative Assessment: Group project followed by an optional oral exam. The group project is aimed at: (1) verification of theoretical competences, and in particular of knowledge and comprehension of Course contents; (2) verification of skills in understanding and solving significant problems, and in explaining the proposed solutions; (3) capability of collaborative work. This in order to verify the ability of application of techniques learnt during the Course, of analysis of problems and synthesis of suitable solutions, and of evaluation of alternative solutions. Criteria of evaluation will be: the level of knowledge and practical ability; the property of use of the technical language; the clarity and completeness of explanations. The oral exam will occur within one week of the project delivery and will typically cover the areas of the project that need clarification plus additional subjects proposed by the teacher. - Assessment breakdown: 100% end-of-semester summative assessment.