UNIVERSIDAD POLITÉCNICA DE MADRID
FACULTAD DE INFORMÁTICA
Campus de Montegancedo
Boadilla del Monte. 28660 Madrid

Agent-Based Software Engineering

Learning Guide – Information for Students

1. Description

Grade	Máster Universitario en Ingeniería de Software - European Master on Software Engineering
Module	Advanced Software Engineering Aspects
Area	
Subject	Agent-Based Software Engineering
Type	Elective
ECTS credits	6
Responsible department	Computer Languages and Systems and Software Engineering
Major/Section/	

Academic year	2012/2013
Term	2nd term
Language	English
Web site	

2. Faculty

NAME and SURNAME	OFFICE	email
Ricardo Imbert Paredes (Coord.)	5112	rimbert@fi.upm.es

3. Prior knowledge required to take the subject

Passed subjects	•
Other required learning outcomes	•

4. Learning goals

SUBJECT-SPECIFIC COMPETENCES AND PROFICIENCY LEVEL		
Code	**Competence**	**Level**
SC13	To have a vision of the different specific and emergent aspects of the Software Engineering, and to go further in some of them.	S
SC14	To understand what nowadays software engineering procedures can and cannot reach, their limitations and their possible future evolution.	S

Proficiency level: knowledge (K), comprehension (C), application (A), and analysis and synthesis (S)

SUBJECT LEARNING OUTCOMES			
Code	**Learning outcome**	**Related competences**	**Proficiency level**
LR1	Within an application field of Software Engineering, uses and designs the appropriate solution to solve some of its problems, describing the technical difficulties and the application limits	SC13, SC14	A
LR2	Facing a real problem, chooses an appropriate Software Engineering solution, analyzing its viability, what can and cannot be achieved from the current state of development of the selected solution, and what is expected to advance in the future	SC13, SC14	A
LR3	Explains which are the Software Engineering limits and frontiers, and the base of new tendencies and developments and advanced topics and their possible application	SC13, SC14	A

5. Subject assessment system

ACHIEVEMENT INDICATORS		
Ref	**Indicator**	**Related to LR**
I1	Comprehend the agent-based paradigm, its restrictions and strengths, considering it as a software engineering abstraction tool	LR1, LR3
I2	Evaluate the applicability of a agent-based solution regarding the nature of the system	LR1
I3	Design a multiagent system architecture considering software quality criteria	LR1, LR2
I4	Design and implement a software agent structure, responsibilities, permissions and behaviors	LR2
I5	Design and build communication protocols allowing the information interchange among agents in a multiagent system, together to the ontology in which are supported	LR 2
I6	Apply software engineering principles and strategies to the development of a complex, highly uncertain, distributed and concurrent system, which, in addition, involves a number of stakeholders.	LR2, LR3
I7	Apply and adapt an agent-oriented development methodology to build agents and multiagent systems	LR1, LR3
I8	Use an agent-based development framework and language to implement agents and multiagent systems, understanding empirically the particularities of the paradigm	LR2
I9	Combine new technologies and development resources together to well-known practices and artifacts from other software engineering paradigms to solve development challenges	LR1, LR2, LR3

CONTINUOUS ASSESSMENT

Brief description of assessable activities	Time	Place	Weight in grade
Group exercise about risks on agent-based software engineering	Week 2	Classroom	4%
Individual theoretical work about applications of agents (first practical assignment)	Weeks 2-4	Individual work. Presented in the classroom on week 4	15%
Group exercise about agent systems topology	Week 5	Classroom	4%
Individual exercise about agent communication protocols	Week 6	Classroom	3%
Group work about development of a multiagent system (second practical assignment)	Weeks 7-16	Group work with weekly meetings in the classroom. Presented in the classroom on week 16	60%
Group exercise about agent based development	Week 16	Classroom	4%
Student implication and participation	Weeks 1-16	Classroom and Moodle	10%
			Total: 100%

GRADING CRITERIA

The subject is marked following continuous assessment.

The student passes the subject only if 5 or more points on 10 are obtained at the end of the course, regarding the following criteria:

FINAL GRADE = 3% Individual exercises in the classroom + 12% Group exercises in the classroom + 15% First practical assignment + 60% Second practical assignment + 10% Student participation

The final grade will be obtained from five components: (1) individual exercises and (2) group exercises performed in the classroom; (3) a first practical assignment consisting in a brief document and a classroom presentation about applications of agents (proposed by the professor); (4) a second practical assignment about a group development of a multiagent system, with weekly classroom meetings and weekly software integration group activities, also in the classroom; and (5) participation and implication of the student in the subject.

The maximum grade for each of these components and the minimum mark needed to compensate non-passed parts are indicated in the following table.

	MAXIMUM GRADE (and correspondence over the final grade)	**MINIMUM GRADE TO COMPENSATE NON-PASSED PARTS** (and correspondence over the final grade)
Individual exercises in the classroom (3%)	10 (0,3)	-
Group exercises in the classroom (12%)	10 (1,2)	-
First practical assignment (15%)	10 (1,5)	3 (0,45)
Second practical assignment (60%)	10 (6)	4 (2,4)
Student participation (10%)	10 (1)	-

When failed, first and second practical assignment could be repeated in the extra exam period, using the new marks together to the ones obtained in individual and group exercises in the classroom and student participation in the previous period to calculate the final grade of the subject.

6. Contents and learning activities

SPECIFIC CONTENTS		
Unit / Topic / Chapter	**Section**	**Related indicators**
Chapter 1: Introduction to Agents	1.1 General Concepts	I1
	1.2 Agent Architectures	I1
	1.3 Social Nature of Agents	I1
Chapter 2: Agent Oriented Software Engineering	2.1 Pitfalls of Agent Oriented Development	I1, I2, I9
	2.2 Standards	I1
	2.3 Agent Communication Languages	I1
	2.4 Development Frameworks	I1
	2.5 Methodologies	I1
	2.6 Development Notations	I1
Chapter 3: Agent Oriented Analysis	3.1 Particularities of Agent Oriented Methodologies	I1, I2, I7
	3.2 Identification of Sub-organizations	I7
	3.3 Environment Modeling	I7
	3.4 Role Modeling	I4, I7
	3.5 Interaction Modeling	I5, I7
	3.6 Organization Modeling	I7
	3.7 Extensions and Notations for the Analysis	I6, I7, I9
Chapter 4: Agent Oriented Architectural Design	4.1 Organizational Structure	I3, I7
	4.2 Role Refining	I3, I4, I7
	4.3 Interaction Refining	I3, I5, I7
	4.4 Extensions for the Architectural Design	I3, I6, I7, I9
Chapter 5: Agent Oriented Detailed	5.1 Agent Design	I4, I7
	5.2 Services Specification	I4, I7

Design (Part I)	5.3 Activities, Actions and Reactions	I4, I7, I9
	5.4 Detailed Protocols	I5, I7, I9
	5.5 Ontology Design	I5, I7, I9
Chapter 6: Agent Implementation	6.1 Introduction to an Agent Oriented Development Framework	I8
	6.2 Administrative Tools	I8
	6.3 Execution of an Agent	I8
	6.4 Agent Behaviors	I8
	6.5 Agent Messaging	I8
Chapter 7: Agent Oriented Detailed Design (Part II)	7.1 Ontology Construction	I5, I8
	7.2 Specification of ACL Messages	I5, I7, I9
	7.3 Specification of Behaviors	I4, I7, I9
	7.4 Detailing the Agent Design	I4, I5, I7, I9
	7.5 Extensions for the Detailed Design	I6, I7, I9

7. Brief description of organizational modalities and teaching methods

TEACHING ORGANIZATION		
Scenario	**Organizational Modality**	**Purpose**
	Theory Classes	***Talk to students***
	Seminars/Workshops	***Construct knowledge through student interaction and activity***
	Practical Classes	***Show students what to do***
	Placements	***Round out student training in a professional setting***
	Personal Tutoring	***Give students personalized attention***
	Group Work	***Get students to learn from each other***
	Independent Work	***Develop self-learning ability***

TEACHING METHODS			
	Method	**Purpose**	
	Explanation/Lecture	***Transfer information and activate student cognitive processes***	Known as explanation, this teaching method involves the "*presentation of a logically structured topic with the aim of providing information organized according to criteria suited for the purpose".* This methodology, also known as *lecture*, mainly focuses on the verbal exposition by the teacher of contents on the subject under study. The term *master class* is often used to refer to a special type of lecture taught by a professor on special occasions
	Case Studies	***Learning by analyzing real or simulated case studies***	Intensive and exhaustive analysis of a real fact, problem or event for the purpose of understanding, interpreting or solving the problem, generating hypotheses, comparing data, thinking, learning or diagnosis and, sometimes, training in possible alternative problem-solving procedures.
	Exercises and Problem Solving	***Exercise, test and practice prior knowledge***	Situations where students are asked to develop the suitable or correct solutions by exercising routines, applying formulae or running algorithms, applying information processing procedures and interpreting the results. It is often used to supplement lectures.
	Problem-Based Learning (PBL)	***Develop active learning through problem solving***	Teaching and learning method whose starting point is a problem, designed by the teacher, that the student has to solve to develop a number of previously defined competences.
	Project-Oriented Learning (POL)	***Complete a problem-solving project applying acquired skills and knowledge***	Teaching and learning method where have a set time to develop a project to solve a problem or perform a task by planning, designing and completing a series of activities. The whole thing is based on developing and applying what they have learned and making effective use of resources.
	Cooperative Learning	***Develop active and meaningful learning through cooperation***	Interactive approach to the organization of classroom work where students are responsible for their own and their peers' learning as part of a co-responsibility strategy for achieving group goals and incentives. This is both one of a number of methods for use and an overall teaching approach, or philosophy.
	Learning Contract	***Develop independent learning***	An agreement between the teacher and student on the achievement of learning outcomes through an independent work proposal, supervised by the teacher, and to be accomplished within a set period. The essential points of a learning contract are that it is a written agreement, stating required work and reward, requiring personal involvement and having a time frame for accomplishment.

BRIEF DESCRIPTION OF THE ORGANIZATIONAL MODALITIES AND TEACHING METHODS	
THEORY CLASSES	Some theoretical lectures will be arranged during the course to present basic concepts and key aspects, always supported by audiovisual resources and innovative techniques to enhance student comprehension
PROBLEM-SOLVING CLASSES	Professor and students will solve problems in the classroom to apply and fix the knowledge acquired during the theory classes
PRACTICAL CLASSES	Group work is complemented with practical classes in which students have to deal with problems and challenges as close as possible to real life developments
INDIVIDUAL WORK	Students will have to do two individual works, one in the classroom and another one to be done out of classroom time. The professor will provide the instructions to complete them
GROUP WORK	Two kinds of group works are planned during the course: for fixing knowledge, in short sessions in the classroom; to practice and gain skill, during great part of the end of the course. Instructions will be provided in both cases to students by the professor
PERSONAL TUTORING	Students will be able to attend personal tutoring, following the procedure established at the School

8. Teaching resources

TEACHING RESOURCES	
RECOMMENDED READING	de Antonio, A. and Imbert, R. (2005) Combining Requirements Engineering and Agents. In A. Silva and J. L. Maté (eds.) Requirements Engineering for Sociotechnical Systems, pp. 68–83. Idea Group Publishing, Hersey, PA, USA.
	Bellifemine, F., Caire, G. and Greenwood, D. (2007) Developing Multi-Agent Systems with JADE. John Wiley & Sons Ltd, England.
	Bratman, M. E., Israel, D. and Pollack, M. (1988) Plans and Resource-Bounded Practical Reasoning. Computational Intelligence, 4(4): pp. 349–355.
	Brooks, R. A. (1991) Intelligence without Representation. Artificial Intelligence, 47: p. 139–159.
	Franklin, S. and Graesser, A. (1996) Is It an Agent, or Just a Program?: A Taxonomy for Autonomous Agents. In Intelligent Agents III. Agent Theories, Architectures and Languages (ATAL'96), vol. 1193. Springer-Verlag, Berlin, Germany.
	Genesereth, M. R. and Ketchpel, S. P. (1997) Software Agents. Communications of the ACM, 37(7).
	Iglesias, C. A., Garijo, M. and González J. C. (1999) A Survey of Agent-Oriented Methodologies. In J. Müller, M. P. Singh and A. S. Rao (eds.), Proceedings of the 5th International Workshop on Intelligent Agents V: Agent Theories, Architectures, and Languages (ATAL-98), 1555, pp. 317–330. Springer-Verlag, Heidelberg, Germany.
	Jennings, N. R., Sycara, K. and Wooldridge, M. (1998) A Roadmap of Agent Research and Development. Journal of Autonomous Agents and Multi-Agent Systems, 1(1): pp. 7–38.
	Luck, M., Griffiths, N. and d'Inverno, M. (1997) From Agent Theory to Agent Construction: A Case Study. In J. P. Müller, M. J. Wooldridge and N. R. Jennings (eds.), Proceedings of the ECAI'96 Workshop on Agent Theories, Architectures, and Languages: Intelligent Agents III, 1193, pp. 49–64. Springer-Verlag, Heidelberg, Germany.

Maes, P. (1994) Modeling Adaptive Autonomous Agents. Artificial Life, I , 1&2(9): pp. 135–162.

Müller, H. J. (1997) Towards Agent Systems Engineering. Data & Knowledge Engineering, 23: pp. 217–245.

Nwana, H. S. (1996) Software Agents: An Overview. Knowledge Engineering Review, 11(2): pp. 205–244.

Padgham, L. and Winikoff, M. (2004) Developing Intelligent Agent Systems. John Wiley & Sons Ltd, England.

Petrie, C. (2000) Agent-Based Software Engineering. In J. Bradshaw and G. Arnold (eds.), Proceedings of the 5th International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM 2000). The Practical Application Company Ltd., Manchester, United Kingdom.

Rao, A. S. and Georgeff, M. P. (1995) BDI Agents: From Theory to Practice. In V. Lesser (ed.), Proceedings of the First International Conference on Multi-Agent Systems, ICMAS-95, pp. 312–319. MIT Press, San Francisco.

Russell, S. J. and Norvig, P. (2004) Inteligencia Artificial. Un Enfoque Moderno. Pearson Education, Spain.

Shoham, Y. (1990) Agent-Oriented Programming. Tech. Rep.. STAN-CS-90-1335, Computer Science Department, Stanford University.

Shoham, Y. and Leyton-Brown, K. (2009) Multiagent Systems. Algoritmic, Game-Theoretic, and Logical Foundations. Cambridge University Press, USA.

Sterling, L.S. and Taveter, K. (2009) The Art of Agent-Oriented Modeling.The MIT Press, Cambridge, Massachusetts, USA.

Sycara, K. (1998) Multiagent Systems. AI Magazine, 19(2): pp. 79–92.

Wooldridge, M. (1997) Agent-Based Software Engineering. IEE Proceedings Software Engineering, 144(1): pp. 26–37.

Wooldridge, M. (2002) An Introduction to MultiAgent Systems. John Wiley & Sons Ltd. Chichester, England.

	Wooldridge, M. and Ciancarini, P (2000) Agent-Oriented Software Engineering: The State of the Art. In P. Ciancarini and M. Wooldridge (eds.), First Int. Workshop on Agent-Oriented Software Engineering, 1957, pp. 1–28. Springer-Verlag, Berlin, Germany.
	Wooldridge, M. and Jennings, N. (1994) Agent Theories, Architectures, and Languages: A Survey. In M. Wooldridge and N. R. Jennings (eds.), Intelligent Agents - Theories, Architectures, and Languages, Proceedings of ECAI'94 Workshop on Agent Theories, Architectures & Languages, vol. 890, pp. 1–32. Lecture Notes in Artificial Intelligence, Springer-Verlag, Amsterdam.
	Wooldridge, M., Jennings, N. R., Kinny, D. (2000) The Gaia Methodology For Agent-Oriented Analysis And Design. Autonomous Agents and Multi-Agent Systems, 3(3), pp. 285–312. Kluwer Academic publishers.
	Zambonelli, F., Jennings, N. R. and Wooldridge, M. (2003) Developing Multiagent Systems: The Gaia Methodology. ACM Transactions on Software Engineering and Methodology, 12(3): pp. 317–370.
WEB RESOURCES	Subject web site (http://is.ls.fi.upm.es/docencia/abs/)
	Subject Moodle site (http://moodle.upm.es/titulaciones/oficiales/course/view.php?id=1054)
EQUIPMENT	Room 6202
	Group work room 6202

9. Subject schedule

Week	Classroom activities	Lab activities	Individual work	Group work	Assessment activities	Others
Week 1 (5 hours)	• 1.1 – 1.2 Introduction to agents (3 hours)		• Subject study (2 hours)			
Week 2 (10 hours)	• 1.3 Introduction to agents (1 hour) • 2. Agent Oriented Software Engineering (1,5 hours)		• Subject study (3 hours) • First practical assignment elaboration (4 hours)	• Exercise about risks on agent-based software engineering (0,5 hours)	• Evaluation of the group work	
Week 3 (9,5 hours)	• 3.1 – 3.4 Agent Oriented Analysis (3 hours)		• Subject study (2 hours) • First practical assignment elaboration (4,5 hours)			
Week 4 (6 hours)	• 3.5 – 3.7 Agent Oriented Analysis (1 hour)		• Subject study (0,5 hours) • First practical assignment elaboration (2,5 hours)		• Presentation of the first practical assignment in the classroom (2 hours)	
Week 5 (6 hours)	• 4. Agent Oriented Architectural Design (1,5 hour) • 5.1 – 5.4 Agent Oriented Detailed Design Part I (1 hour)		• Subject study (3 hours)	• Exercise about systems topology (0,5 hours)	• Evaluation of the group work	

Week 6 (6 hours)	• 5.4 – 5.5 Agent Oriented Detailed Design Part I (1,5 hours) • 6.1 – 6.2 Agent Implementation (1 hour)		• Exercise about agent communication protocols (0,5 hours) • Subject study (3 hours)		• Evaluation of the individual work	
Week 7 (10 hours)	• 6.3 – 6.5 Agent implementation (2 hours)		• Subject study (5 hours)	• Second practical assignment elaboration (2 hours) • Meeting in the classroom for the second practical assignment (1 hour)		
Week 8 (11 hours)	• 7.1 Agent Oriented Detailed Design Part II (1 hour)		• Subject study (4 hours)	• Second practical assignment elaboration (4 hours) • Meeting in the classroom for the second practical assignment (2 hours)		

Week 9 (10 hours)	• 7.2 Agent Oriented Detailed Design Part II (0,5 hours)		• Subject study (0,5 hours)	• Second practical assignment elaboration (7 hours) • Meeting in the classroom for the second practical assignment (2 hours)	• Evaluation of the group work	
Week 10 (11 hours)	• 7.3 – 7.5 Agent Oriented Detailed Design Part II (1 hour)		• Subject study (0,5 hours)	• Second practical assignment elaboration (8 hours) • Meeting in the classroom for the second practical assignment (2 hours)		

Week 11 (11 hours)	• Weekly meetings for the second practical assignment			• Second practical assignment elaboration (8 hours) • Integration test Meeting in the classroom (2 hours) • Meeting in the classroom for the second practical assignment (1 hours)		
Week 12 (11 hours)	• Weekly meetings for the second practical assignment			• Second practical assignment elaboration (8 hours) • Integration test Meeting in the classroom (2 hours) • Meeting in the classroom for the second practical assignment (1 hours)		

Week 13 (11 hours)	• Weekly meetings for the second practical assignment			• Second practical assignment elaboration (8 hours) • Integration test Meeting in the classroom (2 hours) • Meeting in the classroom for the second practical assignment (1 hours)		
Week 14 (11 hours)	• Weekly meetings for the second practical assignment			• Second practical assignment elaboration (8 hours) • Integration test Meeting in the classroom (2 hours) • Meeting in the classroom for the second practical assignment (1 hours)		

Week 15 (11 hours)	• Weekly meetings for the second practical assignment			• Second practical assignment elaboration (8 hours) • Integration test Meeting in the classroom (2 hours) • Meeting in the classroom for the second practical assignment (1 hours)		
Week 16 (11,5 hours)	• Presentation of the second practical assignment			• Second practical assignment elaboration (8 hours) • Exercise about agent based development (0,5 hours)	• Presentation of the second practical assignment in the classroom (3 hours)	

Note: Student workload specified for each activity in hours