ANX-PR/CL/001-01
LEARNING GUIDE

SUBJECT

103000538 - Agent-based Software Development

DEGREE PROGRAMME

10AM - Master Universitario En Ingenieria Del Software

ACADEMIC YEAR & SEMESTER

2023/24 - Semester 2

Index

Learning guide

1. Description

1.1. Subject details

Name of the subject	103000538 - Agent-Based Software Development
No of credits	6 ECTS
Type	Optional
Academic year ot the programme	First year
Semester of tuition	Semester 2
Tuition period	February-June
Tuition languages	English
Degree programme	10AM - Master Universitario en Ingenieria del Software
Centre	10 - Escuela Tecnica Superior De Ingenieros Informaticos
Academic year	2023-24

2. Faculty

2.1. Faculty members with subject teaching role

Name and surname	Office/Room	Email	Tutoring hours *
Jose Maria Barambones Ramirez	D-5106	j.barambones@upm.es	M - 10:00 - 12:00 Tu - 10:00 - 12:00 W - 10:00 - 12:00
Ricardo Imbert Paredes (Subject coordinator)	D-5112	ricardo.imbert@upm.es	Tu - 15:00 - 18:00 W - 15:00 - 18:00

* The tutoring schedule is indicative and subject to possible changes. Please check tutoring times with the faculty member in charge.

3. Prior knowledge recommended to take the subject

3.1. Recommended (passed) subjects

The subject - recommended (passed), are not defined.

3.2. Other recommended learning outcomes

- Strong requirement: skills in programming (Java)

- Algorithms and data structure

4. Skills and learning outcomes *

4.1. Skills to be learned

CE13 - Tener una visión de los distintos aspectos específicos y emergentes de la ingeniería del software, y profundizar en algunos de ellos

CE14 - Comprender lo que pueden y no pueden conseguir las prácticas actuales de ingeniería del software, y sus limitaciones y su posible futura evolución.

4.2. Learning outcomes

RA1 - 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

RA2 - 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

RA3 - Explains which are the Software Engineering limits and frontiers, and the base of new tendencies and developments and advanced topics and their possible application

* The Learning Guides should reflect the Skills and Learning Outcomes in the same way as indicated in the Degree Verification Memory. For this reason, they have not been translated into English and appear in Spanish.

5. Brief description of the subject and syllabus

5.1. Brief description of the subject

The continuous search for more powerful and of a higher level new abstraction mechanisms has lead nowadays towards a new development paradigm, based on software agents. This approach, which has been so many times referred as the final solution for all the previously unaffordable problems, far from being a "silver bullet", must be considered as another software development paradigm and, as such, subject to the Software Engineering discipline.

This subject will introduce the students into this new paradigm, settling the basic concepts of the technology, offering them a wide perspective of the current Software Engineering efforts in this area, always from a practical and applied perspective.

5.2. Syllabus

1. Introduction to agents

   1.1. General concepts

   1.2. Agent architectures

   1.3. Social nature of agents

2. Agent oriented software engineering

   2.1. Pitfalls of agent oriented development

   2.2. Standards

   2.3. Agent communication languages

   2.4. Development frameworks

   2.5. Methodologies

   2.6. Development notations

3. Agent oriented analysis

   3.1. Concepts for building agents

   3.2. Analysis according to different methodologies

   3.3. Goal identification

   3.4. Role modelling

3.5. Interface description

4. Agent oriented architectural design

4.1. Architectural design according to different methodologies

4.2. Agent type decision

4.3. System architecture

4.4. Interaction model

5. Agent oriented detailed design (part I)

5.1. Detailed design according to different methodologies

5.2. Detailed protocols

5.3. Process specifications

5.4. Ontology design

6. Agent implementation

6.1. Introduction to an agent oriented development framework

6.2. Administrative tools

6.3. Execution of an agent

6.4. Agent behaviors

6.5. Agent messaging

7. Agent oriented detailed design (part II)

7.1. Ontology construction

7.2. Specification of ACL messages

7.3. Packaging protocols

7.4. Agent detailed desing

8. Development process

8.1. Development scenario

8.2. Development strategy

8.3. Development team roles

8.4. Project startup stage

8.5. Project iteration stage

6. Schedule

6.1. Subject schedule*

Week	Classroom activities	Laboratory activities	Distant / On-line	Assessment activities
1	**1. Introduction to agents** Duration: 02:00 Lecture **2 Agent oriented software engineering** Duration: 00:15 Lecture			**Exercise about risks on agent based software engineering** Group work Continuous assessment and final examination Presential Duration: 00:45
2	**3. Agent oriented analysis** Duration: 01:30 Lecture **Second assignment. Analysis of the system** Duration: 01:30 Problem-solving class			
3	**4 Agent oriented architectural design** Duration: 01:30 Lecture **Second assignment. Architectural design of the system** Duration: 01:30 Problem-solving class			**Exercise about systems topology** Group work Continuous assessment and final examination Presential Duration: 01:00
4	**5. Agent oriented detailed design (part I)** Duration: 01:00 Lecture			**Exercise about agent communication protocols** Individual work Continuous assessment and final examination Not Presential Duration: 02:00 **First assignment. Presentation in the classroom** Individual presentation Continuous assessment Presential Duration: 00:15 **First assignment. Delivery of the assignment** Individual work Continuous assessment and final examination Not Presential Duration: 09:45

5	**8. Development process** Duration: 00:20 Lecture **Second assignment. Workshop of user stories - sprint #1** Duration: 00:40 Cooperative activities **6. Agent implementation** Duration: 02:00 Problem-solving class			
6	**6. Agent implementation** Duration: 02:00 Problem-solving class **7. Agent oriented detailed design (part II)** Duration: 01:00 Problem-solving class			**Exercise about agent implementation** Individual work Continuous assessment Not Presential Duration: 05:00
7	**Second assignment. Roadmap. Sprint planning #1** Duration: 03:00 Cooperative activities			
8	**Second assignment. Workshop of user stories - sprint #2** Duration: 03:00 Cooperative activities			
9	**Second assignment. Integration sprint #1** Duration: 02:00 Cooperative activities **Second assignment. Retrospective - sprint #1. Sprint planning #2** Duration: 01:00 Cooperative activities			
10	**Second assignment. Workshop of user stories - sprint #3** Duration: 03:00 Cooperative activities			
11	**Second assignment. Integration sprint #2** Duration: 02:00 Cooperative activities **Second assignment. Retrospective - sprint #2. Sprint planning #3** Duration: 01:00 Cooperative activities			
12	**Second assignment. Workshop of user stories - sprint #4** Duration: 03:00 Cooperative activities			
13	**Second assignment. Integration sprint #3** Duration: 02:00 Cooperative activities **Second assignment. Retrospective - sprint #3. Sprint planning #4** Duration: 01:00 Cooperative activities			

14				
15	**Second assignment. Integration sprint #4** Duration: 02:00 Cooperative activities **Second assignment. Retrospective - sprint #4** Duration: 01:00 Cooperative activities			
16				
17				**Second assignment. Presentation in the classroom** Group presentation Continuous assessment Presential Duration: 04:00 **Student implication and participation** Other assessment Continuous assessment Presential Duration: 00:00 **Second assignment. Delivery of the inform and code** Group work Continuous assessment Not Presential Duration: 100:00

Depending on the programme study plan, total values will be calculated according to the ECTS credit unit as 26/27 hours of student face-to-face contact and independent study time.

* The schedule is based on an a priori planning of the subject; it might be modified during the academic year, especially considering the COVID19 evolution.

7. Activities and assessment criteria

7.1. Assessment activities

7.1.1. Assessment

Week	Description	Modality	Type	Duration	Weight	Minimum grade	Evaluated skills
1	Exercise about risks on agent based software engineering	Group work	Face-to-face	00:45	6%	0 / 10	CE14
3	Exercise about systems topology	Group work	Face-to-face	01:00	6%	0 / 10	CE13
4	Exercise about agent communication protocols	Individual work	No Presential	02:00	3%	0 / 10	CE13
4	First assignment. Presentation in the classroom	Individual presentation	Face-to-face	00:15	1.5%	0 / 10	CE13 CE14
4	First assignment. Delivery of the assignment	Individual work	No Presential	09:45	13.5%	3 / 10	CE14 CE13
6	Exercise about agent implementation	Individual work	No Presential	05:00	3%	0 / 10	CE14
17	Second assignment. Presentation in the classroom	Group presentation	Face-to-face	04:00	12%	0 / 10	CE14 CE13
17	Student implication and participation	Other assessment	Face-to-face	00:00	7%	0 / 10	CE13
17	Second assignment. Delivery of the inform and code	Group work	No Presential	100:00	48%	4 / 10	CE14 CE13

7.1.2. Global examination

Week	Description	Modality	Type	Duration	Weight	Minimum grade	Evaluated skills
1	Exercise about risks on agent based software engineering	Group work	Face-to-face	00:45	6%	0 / 10	CE14
3	Exercise about systems topology	Group work	Face-to-face	01:00	6%	0 / 10	CE13
4	Exercise about agent communication protocols	Individual work	No Presential	02:00	3%	0 / 10	CE13
4	First assignment. Delivery of the assignment	Individual work	No Presential	09:45	13.5%	3 / 10	CE14 CE13

7.1.3. Referred (re-sit) examination

Description	Modality	Type	Duration	Weight	Minimum grade	Evaluated skills
Exercise about risks on agent based software engineering	Individual work	Face-to-face	03:00	6%	0 / 10	CE14
First assignment	Individual work	Face-to-face	03:00	13.5%	3 / 10	CE13 CE14
Exercise about systems topology	Individual work	Face-to-face	03:00	6%	0 / 10	CE13
Exercise about agent communication protocols	Individual work	Face-to-face	03:00	3%	0 / 10	CE13
Second practical assignment	Group work	Face-to-face	50:00	48%	4 / 10	CE13 CE14

7.2. Assessment criteria

Progressive evaluation

The subject is graded following a continuous assessment.

The subject's progressive evaluation consist of:

- *Individual assignments (18% of the final grade):* the student must complete several individual assignments related to some specific aspects of the subject. Some of these activities are performed during class time. These activities, except the exercise about agent implementation, are recoverable in the global and extraordinary evaluation, always that they had been failed (grade below 5).
- *First assignment (15% of the final grade):* the first assignment gives the student a width perspective of the agent paradigm and technology. The student delivers a document with his work and prepares a presentation for a specific live session. Since there will not be enough time in that session for all the students to present their findings, the document weight will be for them of 15% of the final grade, while those presenting will divide their grade between the document (13,5%) and the presentation (1,5%). Only the document part is recoverable in the global and extraordinary evaluation when it has been failed (grade below 3). To pass the subject the student must achieve a minimum grade of 3 between grade and presentation.
- *Second assignment (60% of the final grade):* the second assignment allows the student to learn by doing concepts related to agent analysis, design, software project, agile methodologies, software integration, quality assurance and software configuration management, among others. Students participate in the group development of a big system and, at the end of the course, deliver their code and a document with their work. They also prepare a presentation for a specific classroom session. Only the code and document part is recoverable in the global and extraordinary evaluation when it has been failed (grade below 4). To pass the subject the student must achieve a minimum grade of 4 between grade and presentation.

- *Student participation (7% of the final grade)*: a critical mindset and the analysis skills from the student are valued. These activities are not recoverable, either in the global evaluation or in the final evaluation, since it is the student participation during the classes what it is evaluated.

The student passes the subject only if 5 or more points on 10 are obtained at the end of the course.

Global evaluation

When failed during the progressive evaluation, the student may have a new opportunity to pass the subject repeating the recoverable failed parts. Since these parts cover a 28,5% of the final grade this will be the margin for passing the subject during the global evaluation, except when the second assignment has been failed (grade under 4). The second assignment is not recoverable for the global evaluation since it is delivered the day before the subject's exam date.

Extraordinary evaluation

For the extraordinary term evaluation the student can repeat the following activities, only if they were graded under 5:

- The individual assignments (15% of the final grade).
- The first assignment (13,5% of the final grade).
- The correction of the second assignment (48% of the final grade).

Zero tolerance against fraud

If fraudulent acts are detected during the development of evaluation tests, the provisions of article 13 of the UPM Evaluation Regulations approved by the Governing Council on May 26, 2022 will apply.

8. Teaching resources

8.1. Teaching resources for the subject

Name	Type	Notes
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.	Bibliography	Agent oriented analysis<br />
Bellifemine, F., Caire, G. and Greenwood, D. (2007) Developing Multi-Agent Systems with JADE. John Wiley & Sons Ltd, England.	Bibliography	Agent implementation
Bratman, M. E., Israel, D. and Pollack, M. (1988) Plans and Resource-Bounded Practical Reasoning. Computational Intelligence, 4(4): pp. 349-355.	Bibliography	Introduction to agents: concepts
Brooks, R. A. (1991) Intelligence without Representation. Artificial Intelligence, 47: p. 139-159.	Bibliography	Introduction to agents: reactive architectures
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.	Bibliography	Introduction to agents: definition

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.	Bibliography	Introduction to agents: general view
Müller, H. J. (1997) Towards Agent Systems Engineering. Data & Knowledge Engineering, 23: pp. 217?245.	Bibliography	Architecture conceptualization and design
Padgham, L. and Winikoff, M. (2004) Developing Intelligent Agent Systems. John Wiley & Sons Ltd, England.	Bibliography	Agent oriented development
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.	Bibliography	Introduction to agents: BDI
Shoham, Y. and Leyton-Brown, K. (2009) Multiagent Systems. Algoritmic, Game-Theoretic, and Logical Foundations. Cambridge University Press, USA.	Bibliography	Design of multiagent systems
Sterling, L.S. and Taveter, K. (2009) The Art of Agent-Oriented Modeling.The MIT Press, Cambridge, Massachusetts, USA.	Bibliography	Modeling of multiagent systems
Sycara, K. (1998) Multiagent Systems. AI Magazine, 19(2): pp. 79-92.	Bibliography	Social nature of agents
Wooldridge, M. (2002) An Introduction to MultiAgent Systems. John Wiley & Sons Ltd. Chichester, England.	Bibliography	Introduction to agents: general view

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.	Bibliography	Agent oriented methodology: Gaia
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.	Bibliography	Agent oriented methodology: Gaia
http://moodle.upm.es/titulaciones/oficiales/course/view.php?id=1054	Web resource	Subject Moodle site
Room assigned by the school for the classes	Equipment	Lecture and group work room