

Teaching guide

IDENTIFICATION DETAILS

Degree:	Expert in Robotics
---------	--------------------

Faculty/School:	Senior Polytechnic School
-----------------	---------------------------

Course:	ROBOT COMPONENT PROGRAMMING
---------	-----------------------------

Type:	Compulsory Internal
-------	---------------------

ECTS credits:	3
---------------	---

Year:	3
-------	---

Code:	36217
-------	-------

Teaching period:	Sixth semester
------------------	----------------

Teaching type:	Classroom-based
----------------	-----------------

Language:	English
-----------	---------

Total number of student study hours:	75
--------------------------------------	----

Teaching staff	E-mail
Javier Vázquez Pereda	j.vazquez.prof@ufv.es

SUBJECT DESCRIPTION

La asignatura de Programación de Componentes de Robots pertenece al conjunto de asignaturas de Robótica del Título Propio. La asignatura se centra en el diseño de arquitecturas software para el desarrollo de programas de control en robots, las cuales sean lo más robustas y escalables posibles. Dado que los robots deberán procesar y combinar un gran número de señales sensorias con las que deliberar las acciones, las cuestiones de diseño de una buena arquitectura son críticas a medida que la complejidad del robot debe crecer.

La asignatura se articula en una parte teórica y otra de práctica. En la teórica se impartirán conocimientos de aproximación a esta materia. La práctica se basará en el estudio y reflexión sobre casos prácticos de proyectos de investigación robótica.

GOAL

La impartición de la asignatura toma como base el siguiente objetivo general:

- Introducir el concepto de arquitectura software y relevancia de esta durante la fase de diseño técnico del programa que se desarrolla.
- Conocer distintas clases de arquitecturas software y comprender los escenarios donde serán más adecuadas unas u otras.

The specific aims of the subject are:

- Capacidad para justificar los beneficios de las arquitecturas escalables y modulares
- Definir protocolos de interconexión entre componentes
- Conocimiento de plataformas middleware para la programación de robots que sirven como base para el desarrollo de los algoritmos de control del robot, simplificando así la complejidad total del problema
- Capacidad para introducir elementos de redundancia y tolerancia a fallos durante el diseño de la arquitectura software del robot
- Programación concurrente de la lógica que ejecuta el robot
- Conocimiento sobre el diseño arquitecturas software durante el desarrollo de programas para un computador.

PRIOR KNOWLEDGE

El alumno no es necesario que tenga conocimientos previos para esta asignatura.

COURSE SYLLABUS

TEMA 1.- Introducción a las arquitecturas software

- Introducción de arquitectura
- Clasificación y tipos de arquitecturas
- Diseño basado en componentes
- Práctica.

TEMA 2.- Diseño modular del software del robot

- Identificación de componentes funcionales
- Diseño de una arquitectura técnica adecuada a las necesidades del procesamiento
- Práctica

TEMA 3.- Uso de middleware para el desarrollo de robots

- Introducción a plataformas de desarrollo de componentes
- Descripción de plataformas concretas
- Desarrollo de componentes sobre dichas plataformas
- Práctica

EDUCATION ACTIVITIES

El proceso de enseñanza-aprendizaje de la asignatura se caracteriza por una importante orientación a la

aplicación de los conocimientos, tanto en proyectos desarrollados en laboratorios, en grupo o individualmente, y con la supervisión del profesor, como en proyectos abordados por los alumnos de manera personal y autónoma. Respecto a las actividades de carácter presencial, predominan las lecciones expositivas, las clases prácticas y los laboratorios que serán descritos posteriormente.

El trabajo presencial se completará con una importante carga de trabajo autónomo por parte del alumno, en muchos casos desarrollado en grupo, de manera que se fomente el aprendizaje colaborativo y cooperativo. Las actividades de carácter no presencial previstas incluyen el estudio y trabajo individual, que permitirá trabajar en la fijación de los conceptos abordados en las clases expositivas, así como de la aplicación que de los mismos se realiza en las clases prácticas y laboratorios.

El estudio o trabajo individual, toda vez que sea de índole práctica, estará muy centrado en el estudio basado en problemas con el fin de fijar las metodologías y técnicas de resolución de problemas.

El estudio o trabajo en grupo adquirirá especial relevancia en este modulo para fomentar el trabajo en equipo utilizando, entre otros, el método de aprendizaje cooperativo, así como el método colaborativo mediante el cual cohesionar el trabajo de los alumnos y el profesor, y potenciar la responsabilidad compartida en relación a los resultados del equipo, las habilidades de planificación, liderazgo y la investigación.

Todo el estudio y trabajo realizado por el alumno será supervisado y guiado por el profesor mediante tutorías, individuales o en grupo. En algún tema, el alumno tendrá que realizar en clase la exposición de las principales conclusiones de su estudio o trabajo, lo que permitirá el intercambio de conocimientos y experiencias entre alumnos que fomentan la necesidad de comunicación efectiva y la capacidad de síntesis.

Finalmente, con el fin de facilitar al alumno el acceso a los materiales y la planificación de su trabajo, así como la comunicación con el profesor y el resto de alumnos, se empleará el Aula Virtual, que es una plataforma de aprendizaje on-line que ofrece diferentes recursos electrónicos para complementar, de forma muy positiva, el aprendizaje del alumno.

DISTRIBUTION OF WORK TIME

CLASSROOM-BASED ACTIVITY	INDEPENDENT STUDY/OUT-OF-CLASSROOM ACTIVITY
34 hours	41 hours
Lección Expositiva 10h Clase práctica 12h Tutorías 4h Examen 4h Presentación de Trabajos 4h	Trabajo Individual 20h Trabajo en grupo 21h

SKILLS

Conocimiento sobre el diseño arquitecturas software durante el desarrollo de programas para un computador.

Capacidad para justificar los beneficios de las arquitecturas escalables y modulares

Definir protocolos de interconexión entre componentes

Conocimiento de plataformas middleware para la programación de robots que sirven como base para el desarrollo de los algoritmos de control del robot, simplificando así la complejidad total del problema

Capacidad para introducir elementos de redundancia y tolerancia a fallos durante el diseño de la arquitectura software del robot

LEARNING RESULTS

Capacidad para realizar un diseño arquitectónico, tanto lógico como físico, para el software que gobierna el robot

Capacidad para tomar las decisiones de diseño arquitectónico más adecuadas, según las necesidades específicas del robot

Proponer los protocolos de comunicación robustos y adecuados para la correcta comunicación de los varios procesos software que integran la lógica del robot

Capacidad para desarrollar la lógica de un robot, sobre la base de un middleware robótico disponible en el mercado, ya sea abierto o propietario.

Capacidad para mejorar la robustez de la lógica del robot, previniendo fallos en el comportamiento o, en caso de ser inevitables, incluyendo operativas automáticas de detección y recuperación.

Capacidad para segregar la lógica completa del robot, en hilos de ejecución paralelos, que cooperan y se sincronizan adecuadamente, mediante señales de evento y traspaso de información necesaria.

LEARNING APPRAISAL SYSTEM

La nota final del alumno tendrá en cuenta los siguientes factores:

1) Prácticas realizadas en el laboratorio durante el curso. Todas estas prácticas son de obligada entrega, cada una de ellas se evaluará de 0 a 10, no entregar una práctica supone recibir una calificación de 0 puntos en la misma. Para que una práctica sea dada por válida deberá obtener una calificación igual o superior a 5, siendo necesario repetir la práctica si esto no se consigue. No se conservarán calificaciones entre convocatorias. El promedio de todas estas calificaciones prácticas reflejará el 50% de la calificación final.

2) Examen test: de carácter teórico que se realizará a la finalización del temario con el fin de evaluar la asimilación de conocimientos que ha realizado el alumnado de los contenidos de la asignatura. Se puntuará de 0 a 10 y reflejará el 40% de la calificación final. Es necesario obtener en este examen una calificación mínima de 5 para superar la asignatura. No se conservarán calificaciones entre convocatorias.

3) Participación en clase. Esta nota reflejará, principalmente, la participación del alumno en las clases prácticas de la asignatura. Reflejará un 10% de la nota final.

-Aquellos alumnos que estén exentos de la obligación de asistir a clase, bien por segunda matrícula en la asignatura o sucesivas, bien por contar con autorización expresa de la Dirección del Grado, serán evaluados por el mismo tipo de pruebas. El 10% de la participación en clase podrán obtenerlo asistiendo al menos a tres tutorías con el profesor responsable de la asignatura.

-A efecto de cómputo de convocatorias en una asignatura, solamente se contabilizarán como consumidas aquellas en las que el alumno se haya presentado a todas las pruebas de evaluación, o a una parte de las mismas, siempre que su peso en la nota final supere el 50%, aunque no se presente al examen final. Se entenderá que un alumno se ha presentado a una prueba aunque la abandone una vez comenzada la misma. La condición de No Presentado en la convocatoria extraordinaria estará ligada a la no asistencia o entrega de ninguna prueba, práctica o trabajo que esté pendiente.

-El sistema de evaluación será el mismo entre la convocatoria ordinaria y extraordinaria. En esta última no se podrá recuperar el 10% de la participación.

- Cualquier tipo de fraude o plagio por parte del alumno en una actividad evaluable será sancionado e implicará un 0 en la calificación de esa parte de la asignatura, anulando la convocatoria en curso. La situación, además, será comunicada a la Dirección de la Carrera, que a su vez comunicará a Secretaría General, siguiendo el protocolo establecido en la universidad.

BIBLIOGRAPHY AND OTHER RESOURCES

Basic

Documentación de la asignatura disponible en el Aula Virtual.

Wilson, Scott F. Analyzing Requirements and Defining Solution Architectures. Redmond: Microsoft Press, 1999.

Fernández Aramayo, David Ricardo. Arquitectura de Software. Universidad Tecmilenio, ITESM

Additional

Booch, Grady. Object-Oriented Analysis and Design. Second Edition. Benjamin/Cummings, Redwood: 1994

Jacobson, Ivar, Grady Booch, and James Rumbaugh. El Proceso Unificado de Desarrollo de Software. México: Addison-Wesley, 1999

Kruchten, Philippe. "Architectural Blueprints--The 4+1 View Model of Software Architecture". IEEE Software, Institute of Electrical and Electronics Engineers. November 1995, pp. 42-50