MODULE **DESCRIPTOR**

MODULE TITLE	Distributed Enterprise Systems		
MODULE CODE	CO3409 (L6)	CREDIT VALUE	20 credits / 10 ECTS
SCHOOL	SCHOOL OF SCIENCE		

MODULE **AIMS**

1. To integrate the students' software development skills through the construction of distributed enterprise-level systems
2. To convey the fundamental principles of designing distributed enterprise systems considering scalability, security, heterogeneity and concurrency.
3. To analyse technologies and patterns for enterprise architectures
4. To demonstrate how distributed enterprise systems can support business processes
5. To emphasise the need to deliver business value rather than technical perfection

MODULE **CONTENT**

Enterprise applications are complex, distributed systems composed of distinct interacting components. They provide integrated support for key business processes, which allows an organisation to be more effective and responsive to changing needs than multiple standalone applications.

The emphasis of this module are the technical issues relating to development of distributed software to support enterprise-level goals (design methods, technologies, standards and trends). However, to support communication and decision making during development and deployment, the commercial motivation and system management issues are considered.

Syllabus Content

Business Context

COMMERCIAL ISSUES

Business processes and enterprise applications, e.g. workflow systems, E-commerce.

Management issues: project selection, costs and benefits of an enterprise architecture approach, cash flow and priorities. Economics of cloud-based approaches

DISTRIBUTED SYSTEM ISSUES

A subset of the following topics will be considered: Security, integration, persistence, performance, scalability, heterogeneity, reliability, availability, synchronisation, asynchrony and coordination and agreement algorithms.

Distributed Enterprise System Design and Implementation

ARCHITECTURE

Container and component, persistence, (distributed) transactions, concurrency, events ordering, naming and directory services, security, deployment of software components in an application server, remote method invocation (e.g. Java RMI, Web services). Examination of an enterprise component model (e.g. Enterprise Java Beans)

Architectural design: Service-Oriented Architectures, implementation using Web Services (RESTful approaches, WADL)

Architecture Frameworks (e.g. Zachman and TOGAF)

IMPLEMENTATION

Architectural design and development of server-side software for distributed systems using industry-standard application frameworks (e.g. JEE). Development using a modern environment.

Deployment and integration on the Web using standard frameworks and tools (e.g. JEE)

INTENDED LEARNING OUTCOMES

On successful completion of this module a student will be able to:

1. Critically evaluate the role of patterns, technologies and frameworks in the design of a distributed enterprise system.
2. Compare potential technologies for the development of a distributed enterprise system
3. Implement a distributed application using appropriate technology and frameworks
4. Critically evaluate the relationship between business goals and software development

TEACHING METHODS

A combination of lectures, seminar and practical work will be used. Lectures will consider the commercial context, supporting technology (patterns, frameworks and standards) and discuss illustrative applications.

Lectures introduce concepts, problems and solutions (e.g. persistence, maintaining consistency, transactions) and identify common themes (e.g. container-control vs. component control, convention over configuration, Don't Repeat Yourself). Where appropriate, lectures will build on the students' programming skills by illustrating concepts with source code examples.

Seminars will be used to compare technologies, including discussing the results of students' investigations, and to prepare for practical work, particularly for programming exercises. Seminars will include reviews of sample code and design exercises. Internet materials will also be used.

Practical work will develop programming and software construction skills. The purpose is to examine and evaluate a variety of technologies rather than to develop a high level of expertise in a specific technology. However, to avoid superficiality, as far as possible one programming language and development will be used (e.g. Java and NetBeans can support JEE and cloud-based work using Google Application Engine)

Practical exercises will use a modern IDE to implement examples illustrating enterprise concepts and help the students to apply them to solving problems. Examples provided to demonstrate the implementation of concepts using an IDE will be augmented by questions linking the practical work to theory to ensure the students understand the concepts rather than simply follow the recipe.

The examination will expect students to explain, compare, evaluate and apply to simple situations, concepts, tools, languages and techniques.
The coursework will allow students to develop a prototype, but realistic application and to compare relevant technologies.

The coursework allows the student to demonstrate practical skills and to integrate techniques explored in the practical. Where appropriate, students will use the facilities of an IDE to avoid having to generate boilerplate code so they can focus on high-level integration not low-level coding. They will analyse the concepts, tools and techniques in a short evaluative report.

ASSESSMENT METHODS

This module is assessed through an assignment that will involve the development of a distributed application and an evaluation of the technology used in its construction (50%) and a Written examination (50%).