E0 239 ELECTRONIC COMMERCE  3:1
JANUARY–APRIL 2007

Y. NARAHARI

The vision of the course is provide a sound understanding of the following topics:

1. Software Architecture and Design in E-Commerce
2. Combinatorial Optimization Algorithms in E-Commerce
3. Game Theory and Mechanism Design in E-Commerce

Apart from three tests and a final exam, the course will involve a term paper and a programming assignment. Each student is required to write a term paper on a state-of-the-art research topic in E-Commerce and give a presentation on the same. Also, each student is required to implement a non-trivial market algorithm from the current literature and explore game theoretic mechanisms around that algorithm.

*******************************************************************************

TIME TABLE: Wednesday, Friday, 10-11.30 AM, L7CL
First Meeting: Wednesday, January 3, 2007, 10 AM at L7CL

Assessment:

3 Tests:
Quiz 1: Wednesday, January 24, 2007  5 Marks
Test 2: Monday, February 26, 2007  10-11.30  10 Marks
Test 3: Monday, March 26, 2007  10-11.30  10 Marks

Final Exam
Monday, April 23, 9AM-12Noon  25 Marks

Term Paper  10 Marks
Presentation on the Term Paper  10 Marks

Programming Assignment  30 Marks

*******************************************************************************

CONTENTS OF THE COURSE:
The course will be in three parts.

Part I: Software Architecture and Design for E-Commerce
(Roughly about 20 percent of the course)
This part of the course focuses on best practices in software analysis, architecture, and design, with specific emphasis on e-commerce. The following are some representative topics.

* Software Process: Rational Unified Process
* Object Oriented Modeling and UML (Unified Modeling Language)
* E-Commerce Architectural Patterns
* Design Patterns for E-Commerce
* Case Study: An Electronic Marketplace

Part II: Combinatorial Optimization Algorithms in E-Commerce
(Roughly about 30 percent of the course)

The focus in this part will be on winner determination algorithms in electronic markets. These are mostly based on mathematical programming models. Topics include:

* Classical Auctions (single unit, single item auctions) and Revenue Equivalence Theorem
* Multi-Unit auctions: Volume Discount Auctions
* Multi-Item Auctions: Combinatorial Auctions
* Multi-Attribute Auctions
* Double Auctions and Exchanges
* Approximate Algorithms for Winner Determination

Part III: Game Theory and Mechanism Design in E-Commerce
(Roughly about 50 percent of the course)

In this part of the course, the focus is on understanding the science of designing auctions and other e-commerce mechanisms using game theory and mechanism design theory. Topics include:

* Strategic Form Games and Examples
* Dominant Strategy Equilibria and Nash Equilibria
* Bayesian Games
* Foundations of Mechanism Design
* Properties of Mechanisms: Ex-Post Efficiency, Incentive Compatibility, Individual rationality, budget balance, etc
* Vickrey-Clarke-Groves Mechanisms
* Optimal Auctions
* Case Study: Procurement Auctions
* Case Study: Sponsored Search Auctions on the Web

PROGRAMMING ASSIGNMENT
Each student is expected to carry out a substantive programming assignment of implementing an e-commerce system, using best practices in software design. The assignment will have three "releases"; each release will build upon the previous one and will be reviewed independently.

Release Deadlines:

Release 1: February 1, 2007 (emphasis on OO modeling, design patterns, and software architecture)
Release 2: March 1, 2007 (emphasis on a combinatorial optimization algorithm for winner determination)
Release 3: April 1, 2007 (emphasis on game theoretic properties)

# TERM-PAPER AND PRESENTATION

Each student is required to do an independent study of the current literature focusing on an emerging research topic in E-Commerce and prepare a term paper. It is proposed to publish all the term papers on the course website after a thorough review. Every student shall make a 20 minute presentation of the term paper towards the end of the course.

# PREREQUISITES

It is desirable the students have solid programming skills in C and a sound exposure to data structures and algorithms. Familiarity with C++ or Java is helpful but not mandatory. Similarly, familiarity with linear programming and basic optimization would be useful but not mandatory.