



# Secure Coursework Submission System

Massimo Felici

Room 1402, JCMB, KB

0131 650 5899

[mfelici@inf.ed.ac.uk](mailto:mfelici@inf.ed.ac.uk)

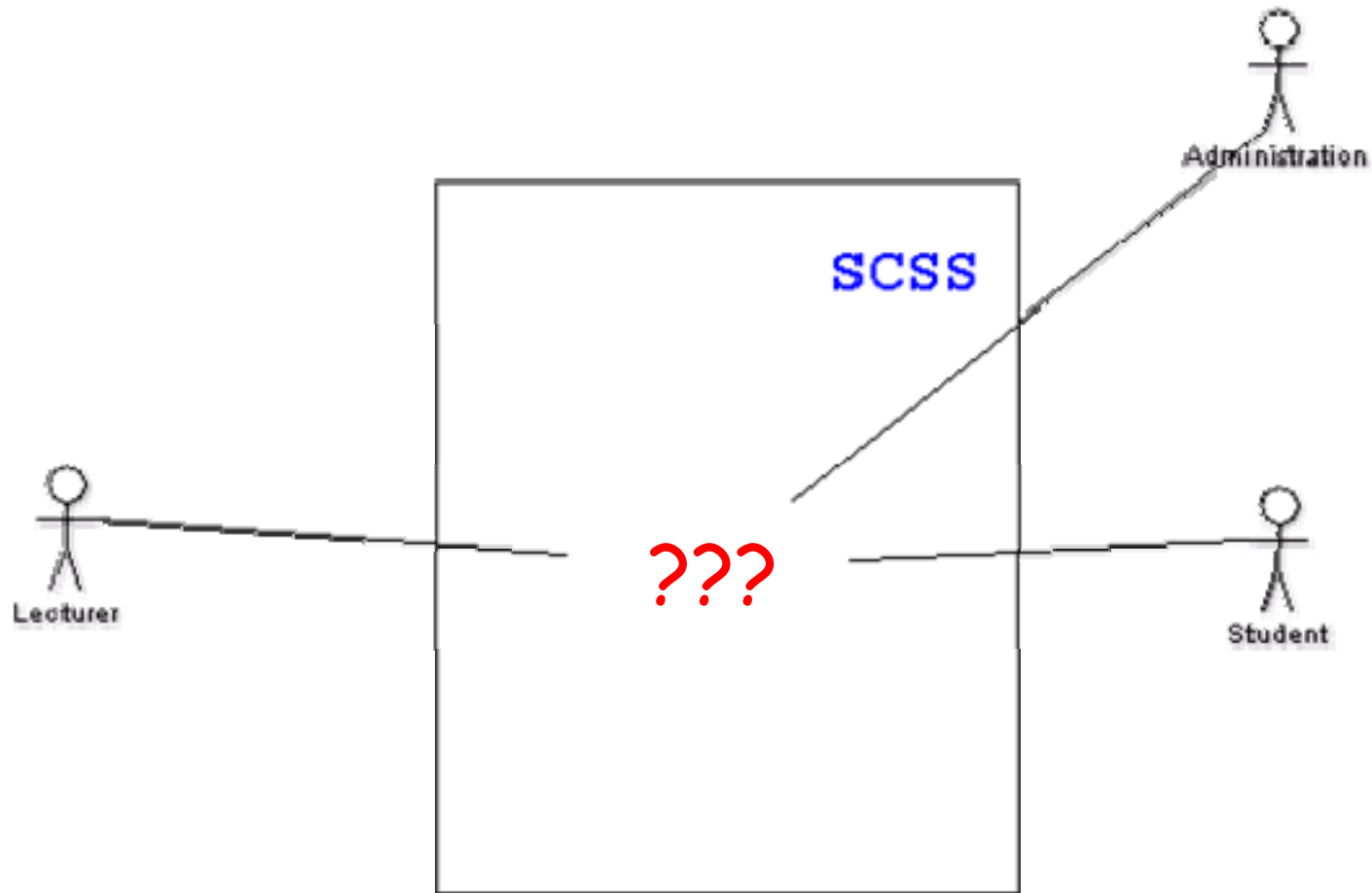
# Main Sources of Requirements

- The references in the practical web page provide the main information for your system requirements

[http://www.inf.ed.ac.uk/teaching/courses/seoc/2006\\_2007/practicals/](http://www.inf.ed.ac.uk/teaching/courses/seoc/2006_2007/practicals/)

1. M. Luck, M. Joy. A Secure On-line Submission System.
  2. M. Joy, P.-S. Chan, M. Luck. Networked Submission and Assessment.
  3. Informatics Web Submission System, Project Proposal, University of Edinburgh.
- They represent a basis for eliciting and gathering your system requirements

# Secure Coursework Submission System



# Requirements Viewpoints

- What is a viewpoint?
  - Encapsulation of partial information about system requirements from a particular perspective
- Requirements Viewpoints:
  - **Student viewpoint (S)**
  - **Lecturer viewpoint (T)**
  - **Admin viewpoint (A)**
- Why viewpoints?
  - Requirements completeness
  - Partial specifications
  - Viewpoints highlight requirements association, hence Traceability
- What are your system requirements?
  - You are looking at the same system from different viewpoints
  - Requirements viewpoints highlight system requirements (i.e., your perspective understanding of the system)
- Identify
  - high level requirements
  - the System Stakeholders
  - The scope of the system
- Capture requirements from your viewpoint (i.e., S, T or A)
- **Main Issues:** Difficult to Identify a stable set of requirements; Contradicting viewpoints

# Document your Requirements

## 1. Document your Requirements

- Use a Requirements Specification template

- o Overall Objectives - "The system has to support teaching and administrative activities related to the courses' assessments. The system should support the secure submission of course exams and solutions."
- o Functional Requirements
  1. "The system should be integrated with DICE."
  2. "The system should allow the submission of practicals and exams."
  3. "The system supports logins from different locations."
- o ...
- o Non-Functional Requirements
  1. "The System should guarantee secure submissions"
- o ...
- o Others
  1. Open Issues

# UML Modeling

## 2. Use Cases

- Capture your “functional requirements” graphically
- Define the system boundaries
- Generalizing and structuring let you simplify use cases
- Use a Template to describe each use case

## 3. Class Diagrams

- Some class may have missing information, because you don't yet know all the “responsibilities” and the “collaborators”



# Validate Your (Requirements) Modeling

## 4. Validation by CRC Cards

- Run few use cases (point 2) to validate your class diagrams (and use cases too)
- Select significant use cases
- Select also use cases involving different Actors (e.g., S and T, S and A, T and A).

# Deliverable 1 Assessment

## ■ Completeness

**Deliverable 1** consists of 1-4 and 5 (Deliverable 1 Assessment and marking)

## ■ Coverage (Requirements Completeness)

- The "basic functionalities" should be covered

## ■ Quality (of the design)

- UML and OO Design Proficiency
- Software Engineering practice (e.g., use of templates)

