

Francisco Diaz Riollano

Prof. Marko Schütz

Course: INSO 4101

Due Date: August, 17, 2018

Homework #1  
Domain of study: What is a university?

Problem set:

**Exercise 1.1 Domain Entities:** For the fixed topic, selected by you, list some dozen or so domain entities. Give suitably short names for their types and describe these, whether simple or composite, and, if composite, describe their composition.

**type**

Student, Book, Id, Class, Institute, Professor, Department, Dean, Test, Grade, Homework, Office,  
Classroom

**value**

student:Student

id:Id

prof:Prof

book:Book

obs\_B: Subj → Book-set

obs\_Prof: Class × Department → Professor-set

obs\_Grade: Professor × Student × Subject × Test → Grade

Simple

Dean  
Id  
Book  
Professor  
Subject  
Test  
Grade  
Homework  
Student

Composite

Institute  
Department  
Office  
Classroom

Institute: is composed of {professors, students and books}

Department: is composed of {students, professors, classrooms, offices}

Office: is composed of {professors and books}

Classroom: is composed of {students and professors}

**Exercise 1.2 Domain Functions:** For the fixed topic, selected by you, list some half dozen or so domain functions: Give suitably short names to these functions, and describe their signatures, that is, which arguments they “take”, and what results in the “yield”.

1. grd: **Nat** → Grade
  2. passed: Student × Grade → **Bool**
  3. research\_subject: Institute → Subject
  4. classroom\_number: Classroom → Department × **Nat**
  5. attendance: Professor×Classroom → Student-set
  6. research\_interest: Professor → Subject-set
- 
1. Takes one argument: a natural number. Yields the corresponding grade, according to the grade distribution
  2. Takes two arguments: a student and his grade. Yields whether the student passed or not. This according to the grading rule.
  3. Takes one argument: an institute. Yields the subject area the institute works or studies.
  4. Takes one argument: a classroom. Yields the department name and the number associated with it.
  5. Takes two arguments: a professor and classroom. Yields the set of students that should be in the classroom.
  6. Takes one argument: a professor. Yields the set of research subject he or she is interested.

**Exercise 1.3 Domain Events:** For the fixed topic, selected by you, list some half dozen or so domain events. Give suitably short names to these events, and describe them briefly.

1. The professor assigns homework for next week.
2. Student enrolls a new course.
3. The professor hands a student his grade.
4. Professor obtains distinguished research award.
5. Student pays his tuition.
6. A student schedules an appointment with his professor.

**Exercise 1.4 Domain Behaviours:** For the fixed topic, selected by you, list, say, three behaviours. Give suitably short names to these behaviours, and describe them briefly.

**Hint:** Think of behaviours as processes, i.e., of a behaviour as “one” process. Then describe that behaviour as it may also interact, or communicate, thus exemplifying events, with other behaviours.

Student is enrolling a course:

**Type**

Student, Professor, Course, Section, Laptop, System, Date  
M == mkenroll(c:Course, sec:Section)

**Channel**

Laptop, System:M

**variable**

ct:M

**Value**

enroll\_course: Student × Course × Section → Date × Professor

can\_enroll: Student × Course × Section → Bool

sec\_list: Course → Section\*

Enroll\_Course (student) (course) (section) ≡

```
p: while System? do
    ct ! mkenroll (course)(section)
    if (can_enroll (student) (course) (section) == True)
        enroll_course(student)(course)(section)
    else
        sec_list (course)
end
```

A student is trying to enroll a specific course. While the system is available, the student will use the channel to send a message which tells the System the course he wants to enroll. If the student is allowed to enroll the course, then he receives the corresponding message of the date he will be attending and the professor. If he cannot enroll the course, the system provides a list of other available sections.

**Exercise 1.5 Domain Requirements:** For the fixed topic, selected by you, list, say three or four, domain requirements. Describe them briefly and informally.

1. Grading Policy: policy adopted between the students and the professor of the course as to how grades will be distributed.
2. Inquiries: questions between the student and the professor on a particular assignment
3. Syllabus: distribution of assignments, projects, tests and the topics to be discussed for a course.

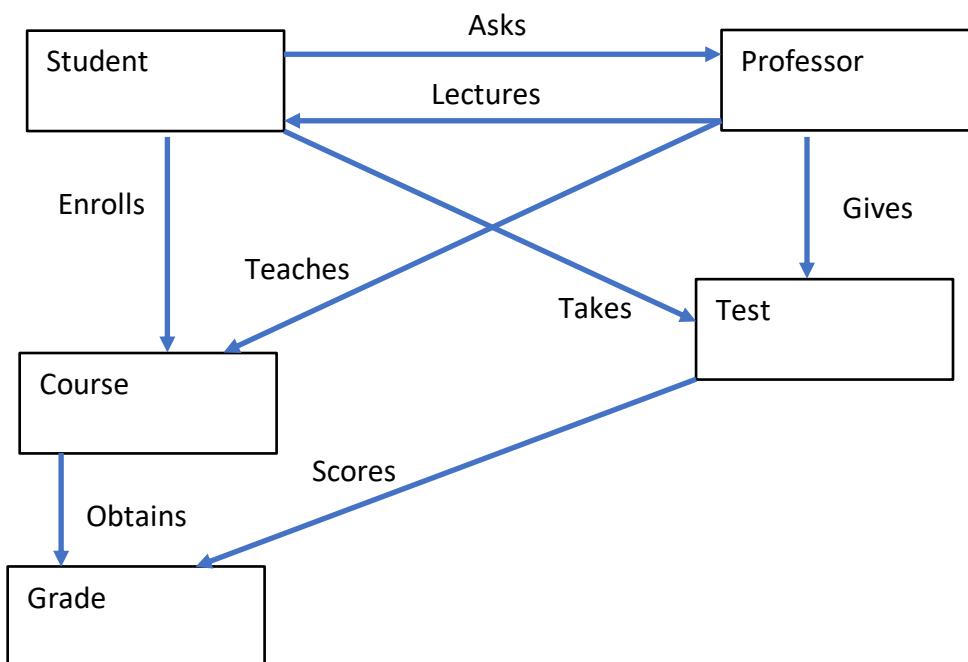
**Exercise 1.6 Interface Requirements:** For the fixed topic, selected by you, list two or three interface requirements. Describe them briefly and informally.

1. Students should be able to have a system where they can add, drop courses as well as manage assignments in each of the courses enrolled within the university.
2. A system that tracks the courses each student has taken and whether he or she has passed or failed the course and determines overall track record of a student.

**Exercise 1.7 Machine Requirements:** For the fixed topic, selected by you, list one machine requirements for each of the “standard” areas: performance, dependability, maintenance, platform, and documentation. Describe them briefly and informally.

- Performance – The system must serve up to 5,000 concurrent users, with a serving time no less than 3 seconds.
- Dependability - The computer should be able to provide an adequate amount of time to for each user with a restricting amount of 3 seconds per query made. In case more time is needed for any process the system should notify all relevant parties of the changes made.
- Maintenance – Any updates should be done under university law and consent.
- Platform – A system will all relevant data and metadata to the students and professor’s academic information. The system should have well defined levels of permissions in accordance to university law.
- Documentation – A file with all of the system functionalities and permissions allowed by administrators, students, professor’s, etc. These files of documentation should be updated regularly under university law.

**Exercise 1.8 Software Architecture Design:** For the fixed topic, selected by you, attempt, admittedly rather prematurely, to sketch a software architecture — say in terms of (briefly specified) boxes and (briefly specified) arrows, where boxes denote single-thread processes and arrows denote interactions (messages) between processes. Do this only informally.



**Exercise 1.9 Software Component Design:** This exercise is a continuation of Exercise 1.8. For the architecture sketch given, by you, in answer to Exercise 1.8, single out one or two “boxes” and specify their data structures and functions. Do this only informally.

Student is a tuple that contains the classes he is taking, grades so far, schedule for a week

Professor is a list of the courses he or she is lecturing for the semester.