This is an exercise from the Merode course at KU Leuven.
The solution is given as a UML model and as an EDG (part of a Merode model)

APPLYING THE THREE PARTY PATTERN TO A CASE

Case

(simplified version of exam case January 2010)

Students can ask questions about the courses they are following by making an appointment with the teaching assistants of these respective courses. To be able to manage the high demand for appointments with the teaching assistants, a system is made to keep track of every appointment of a teaching assistant and a student(s). A student can only request an appointment for a particular course when (s)he is also registered for the respective course. A teaching assistant can also only provide help for the courses (s)he is teaching. An appointment is always for only one specific course, and with one particular assistant who is TA (teaching assistant) for that course.

General Context

- In the case, there are three business objects that are potentially related: Student, Course, Assistant

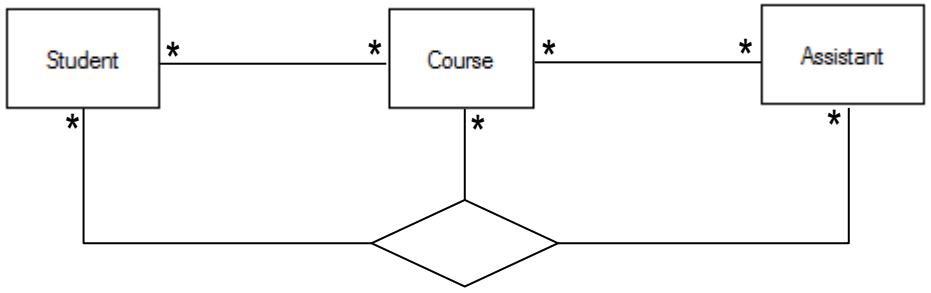
- Students are related to course through their subscription
- Courses and Assistants are related through teaching assignment
- Students and Assistants could be related if students are assigned to particular teaching assistants. In the case, there is no indication of such a relationship, so we –for the moment- assume we don’t need such a relationship

Step 1a: the need for a ternary relationship

- Knowing which student is subscribed to which course and which assistant teaches which course, and even which student is assigned to which teaching assistent, does not allow to know which student has an appointment with which teaching assistant for which course.
- Suppose for example that Malcolm is subscribed for both D0I71A and for D0I70A, and that Helen is teaching assistant for both courses, and if we know that Malcolm has an appointment with Helen, we still don’t know if that’s for D0I70A or D0I71A.
- So we need a ternary association that relates student, course and assistant

Step 1b: needing the binary associations as well

- As an appointment for a course can only be made if the student is subscribed to the course, we need to keep the binary association that tells us which student is subscribed to which course
- As an appointment for a course can only be made with an assistant who is a teaching assitant (TA) for that course, we need to keep the binary association that tells us which assistant is TA for which course.
- So the resulting UML model is as follows

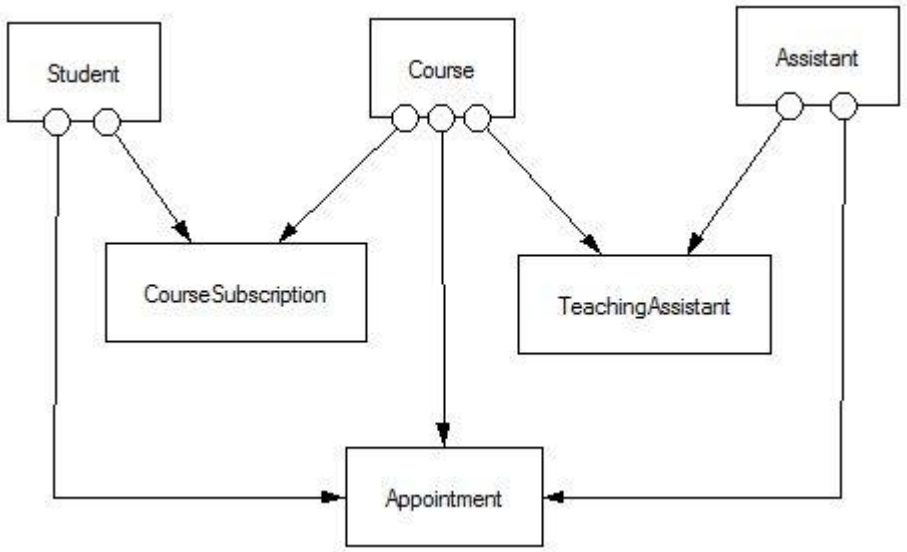


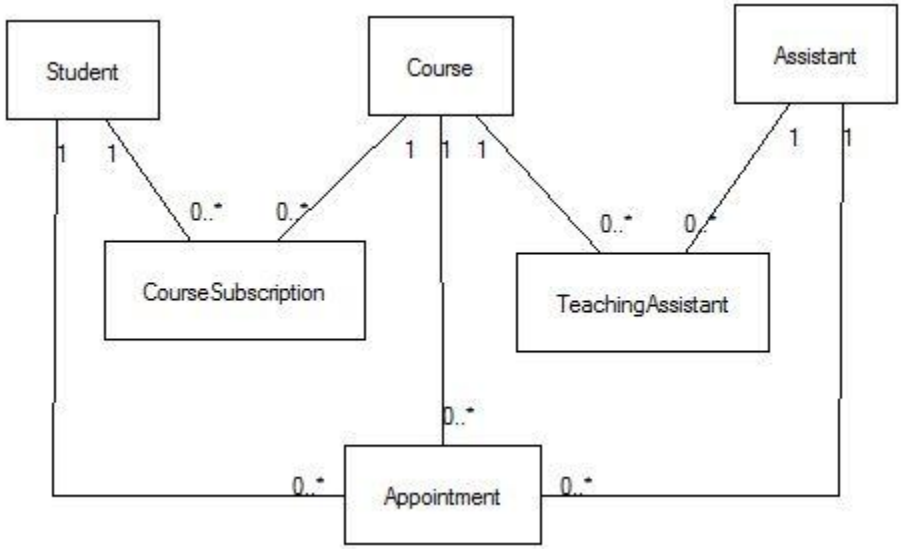
Step 2: Managing the evolution of associations

- The associations give valuable information, but there are potential additional requirements for history, evolution over time and life cycle information
  - When a student subscribes to a course (i.e. adds this course to his/her ISP), this subscription must be approved by the students office. Once subscribed a student will have two chances to pass exam to obtain a credit for this course. Once the credit is obtained, this credit can be used to obtain a degree.
  - When an assistant is assigned as teaching assistant for a course, this is registered as a certain percentage of Full Time Equivalent (FTE) teaching load. The overall teaching load should not exceed 20% FTE for PhD students with a scholarship.
  - When an appointment is registered, the date& time of the appointment allows to determine whether this is a future or past appointment
- In conclusion, all three associations have a relevant life cycle.

Step 2: Managing the evolution of associations

- Applying the Association Pattern to each association yields four potential assiociation classes
  - http://dirkriehle.com/computer-science/programming/patterns/association-objects/index.html
- This yields the same result as if one would apply the rules for transforming a UML class diagram with many to many relations to an EDG
- The result is the following EDG/ UML Class diagram





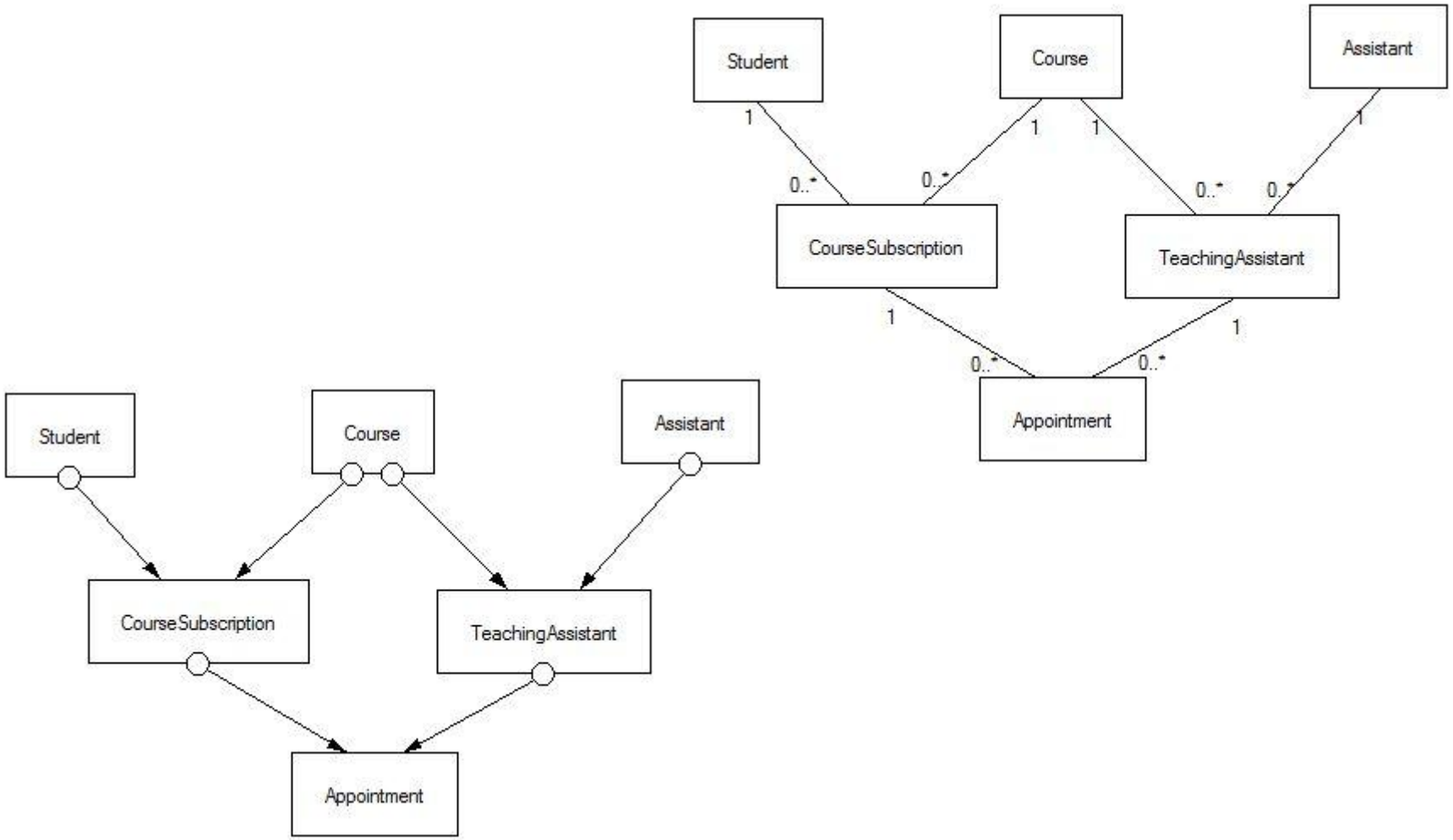
Step 3: Three Party Pattern

- The information contained in the binary association is strongly related to the information contained in the ternary association:
  - The appointment can only be made in the context of a course subscription as the student needs to be subscribed to be allowed making an appointment
  - The appointment can only be made in the context of a Teaching Assistant assignment as the appointment can only be made with a TA for that course.

- The existing relationships between all objects are reorganised. Instead of letting the ternary association object collaborating with the business objects directly, it should collaborate with each of the two binary association objects, as shown in the diagrams on the next slide

  - As a result, the appointment is ED of the course subscription and the TA

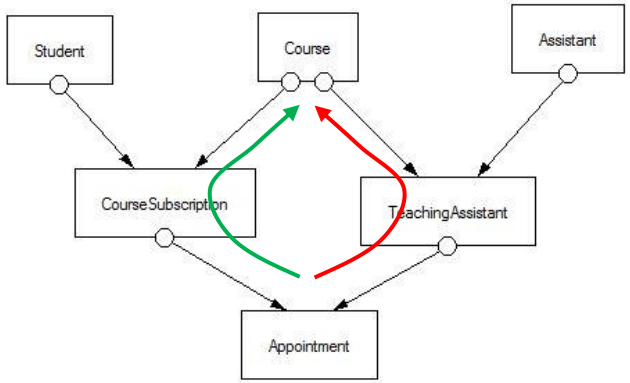
Step 3: Three Party Pattern

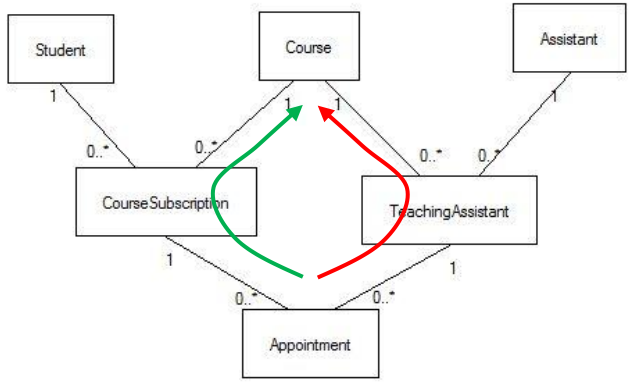


Advanced considerations

cr_Appointment	A/M	A/M A/M	A/M	A/M	A/M	O/C
end_Appointment	A/M	A/M A/M	A/M	A/M	A/M	O/E

- Looking at the OET, one immediately sees that potentially two courses could be involved in the creation of an appointment. The double paths from APPOINTMENT to COURSE are present both in the EDG and in the UML diagram, but are (more) easily discovered by looking at the OET





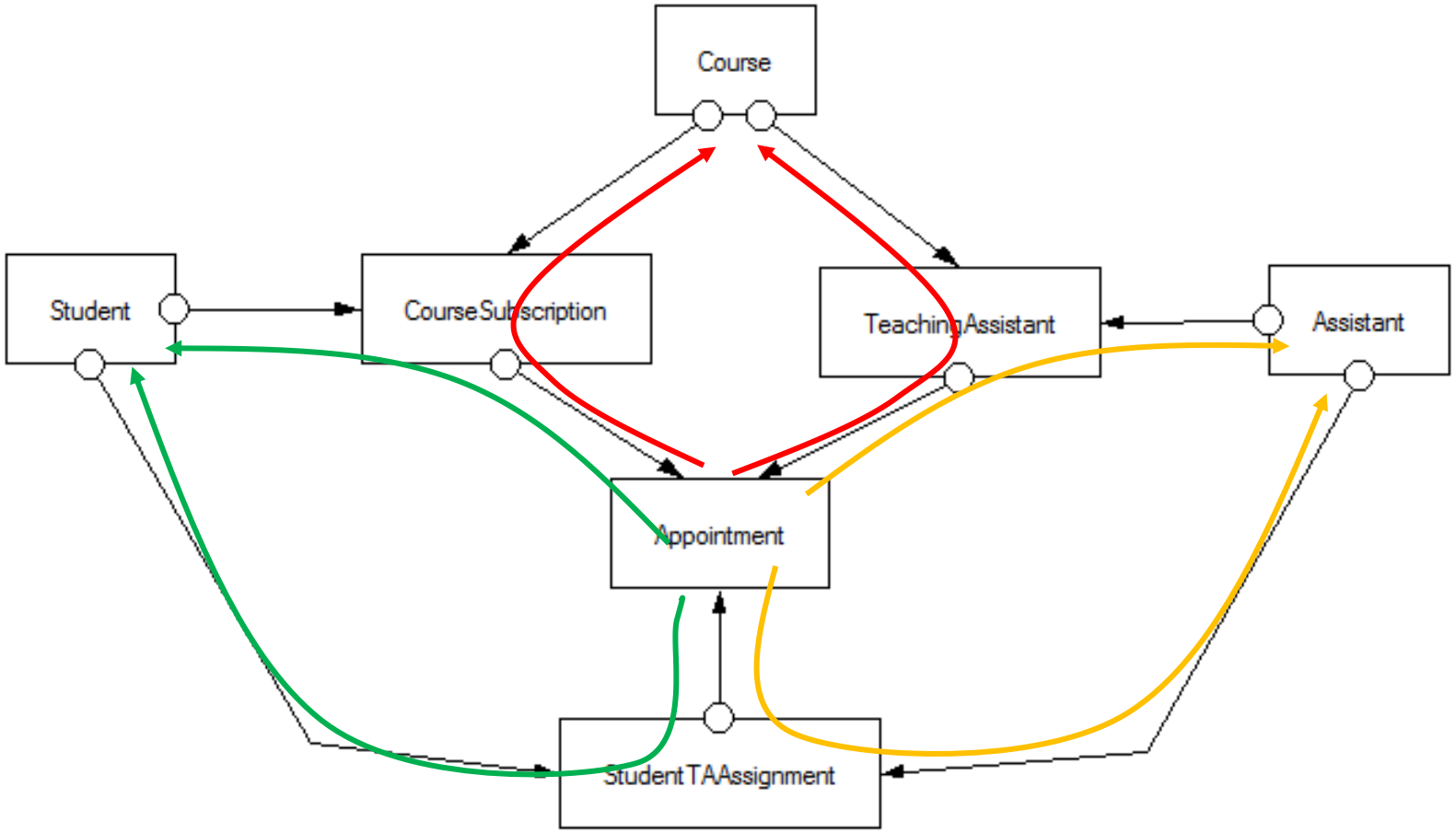
- A constraint should be set in APPOINTMENT, stating that the course the student is subscribed to (green arrow), should be the same as the course the assistant is TA for (red arrow).

Variation 1

- Suppose that students are assigned to teaching assistants. For example, assume that for D0I71A, the system allows to register that students from the Master of Information Management are assigned to Tom and that all other students are assigned to Pieter.
- Assume also that students can only register for an appointment with the TA they are assigned to.
- Then in the context at the start, the third association between Student and Assistant becomes relevant, and we obtain the “pure” three party pattern solution

- The arrows indicate multiple constraints to consider adding to the model

Variation 1

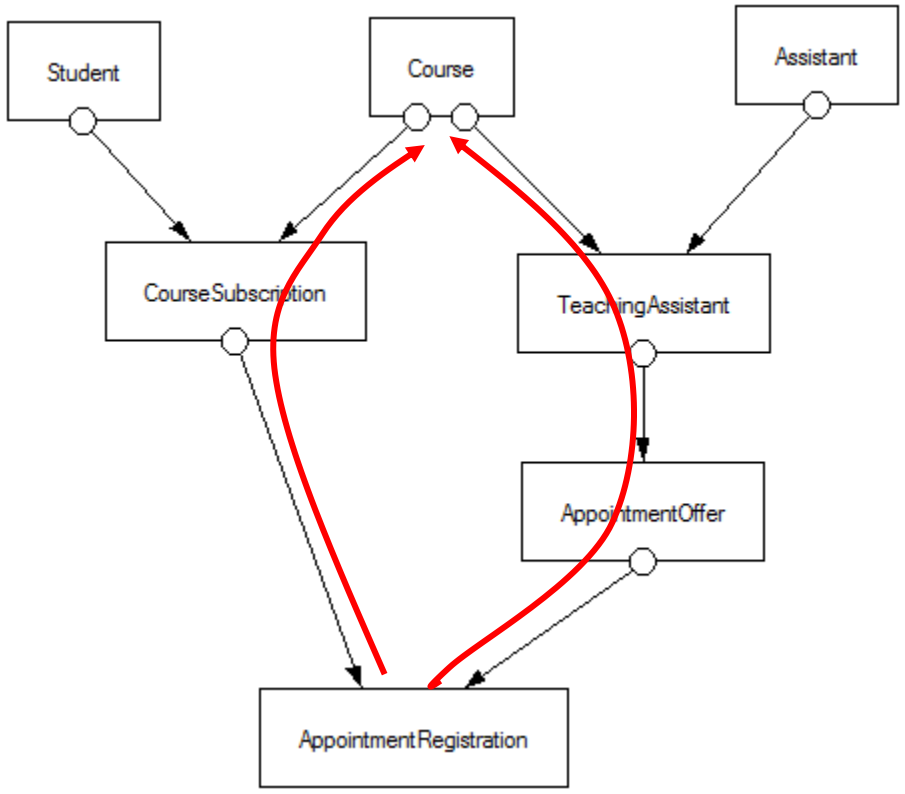

Course
Student
CourseSubscription
TeachingAssistant
Assistant
Appointment
StudentTAAssignment

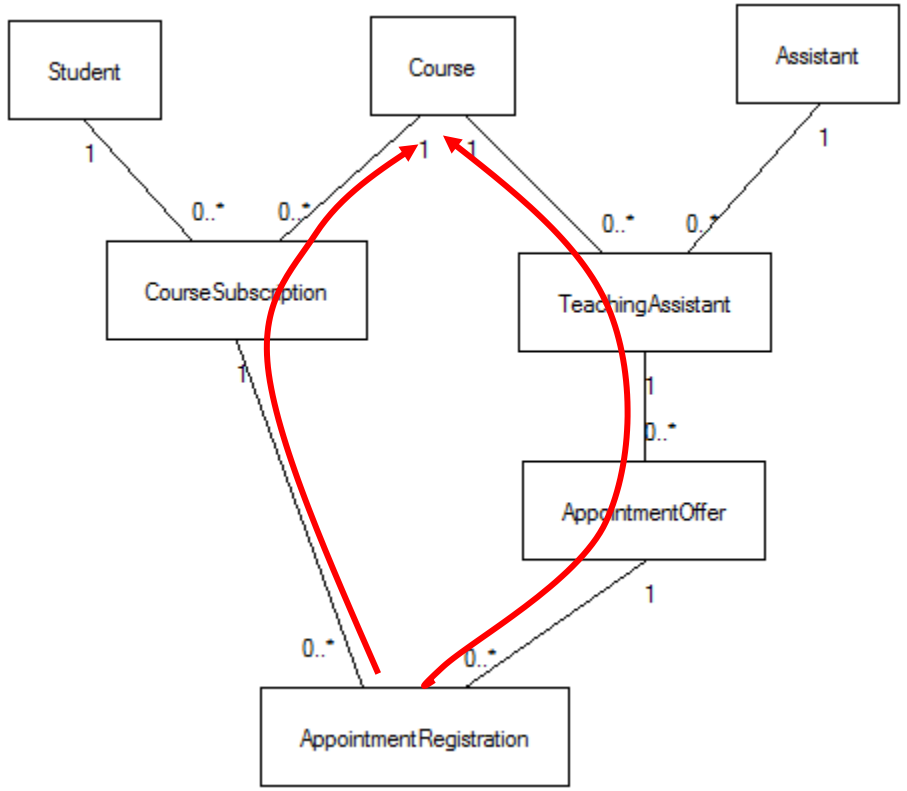
Variation 2

- The present solution models a situation where appointment objects are created at the moment a student agrees on an appointment with a teaching assistant. It does not allow to offer a set of appointments (i.e. time slots) to which students can subscribe. If we would like to offer the functionality that a TA can create a list of time slots for which students can subscribe, we need to make a difference between two objects
  - AppointmentOffer = time slot published e.g. in Toledo
  - AppointmentRegistration = student registered for an AppointmentOffer.

- Additionally, an AppointmentOffer could have a number of available places, enabling multiple subscriptions for the same time slot. In this way, an assistant can help several students in one session

Variation 2


Student
Course
Assistant
CourseSubscription
TeachingAssistant
AppointmentOffer
AppointmentRegistration


Student
Course
Assistant
1
1
1
0..*
0..*
0..*
0..*
CourseSubscription
TeachingAssistant
1
1
0..*
AppointmentOffer
1
0..*
0..*
AppointmentRegistration

Variation 3 = Variation 1 + Variation 2