

Seminar 2 - Logical and Physical Models

Data Storage Paradigms, IV1351

Kevin O'Flaherty, kevinof@kth.se

December 9, 2022

1 Introduction

In the project's first task we were expected to create a Conceptual Model based on the business description of the Soundgood Music School, using UML or other crow foot notation. For a higher grade the requirement was to use and demonstrate valid cases of inheritance in this model.

For this task I started off working with Vilhelmina Andersson, Patricia Lagerhult and Johan Seller-Fredlund. We collaborated in the early stages, while the work centered on brainstorming and discussion. For pruning entity candidates etc. I worked mostly on my own, returning to my colleagues sporadically.

2 Literature Study

Before doing the task I watched the videos provided regarding the Conceptual Model. I also reviewed my notes from taking the course IV1350 in the spring of 2022. These notes were in turn based on A First Course in Object Oriented Development - A Hands-On Approach by Leif Lindbäck. In particular I used them for how to generate a Domain Model (adapting it for the Conceptual Model as described in the Method chapter).

Further, I read the course material pdfs tips-and-tricks-task1 and assessment-criteria-seminar1 to guide my approach.

3 Method

For this assignment I created the Conceptual Model using an ER-diagram in Astah. Working with three other students, the initial skeleton of the model was produced by analyzing the business description through noun identification, adding all nouns in the text as possible entity candidates.

Following that, we brainstormed other possible candidates using the category list from A First Course in Object Oriented Development - A Hands-On Approach, page 27. We spent the better part of the day discussing the naming of these and other entity candidates, trying to be exhaustive without going too far off target.

Even so, I ended up with a lot more candidates than seemed reasonable, and the next step was eliminating the inappropriate ones. Re-reading the business description

revealed several nouns that were (intentionally?) red herrings. Many of the candidates produced through the category lists were more than one step away from the model in question.

Having pruned the model somewhat I started moving several candidates into other candidates, transforming them into attributes instead. There was no real discrete step from removing class candidates to identifying attributes. The two processes overlapped to a point.

I then made sure not to have any attribute-empty entities, and to define the cardinality of each attribute and whether they were allowed to be NULL or not. Lastly, I drew crow foot-associations between the remaining entities focusing on the data relationship between them. This last step was done in tandem with the same study group as earlier. Throughout the steps I continually had to return to the business description, substituting conversation with an actual domain expert from the client with my best guesses and reasoning with my colleagues.

4 Result

The final result was the following diagram:

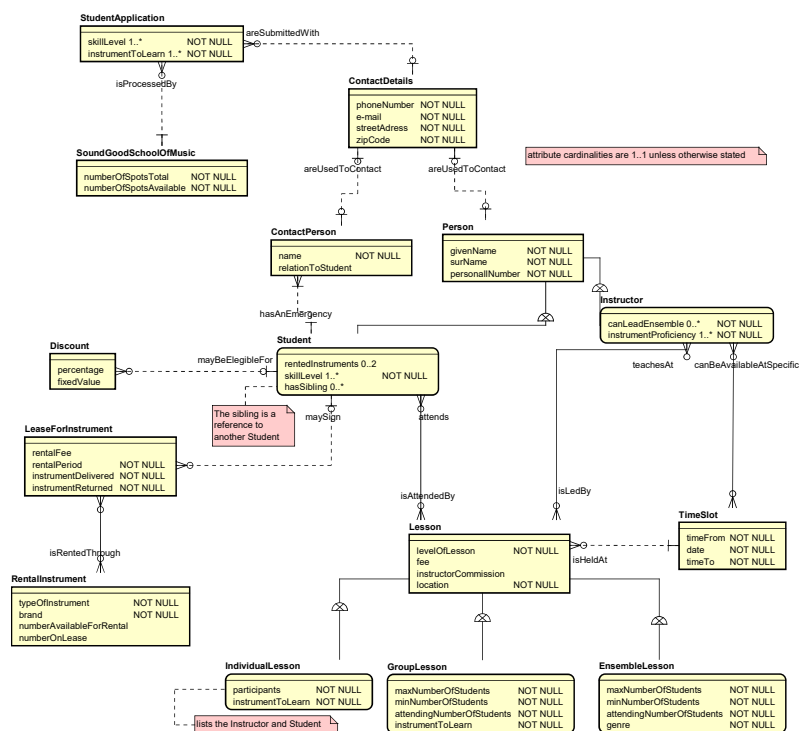


Figure 4.1: Conceptual Model for the Soundgood Music School

The two most central entities are the Person entity (extending into both Student and Instructor) and Lesson (extending into IndividualLesson, GroupLesson and EnsembleLesson). These entities are central to the business logic, and are likely increasingly linked due to them being super entities to their respective subgroups.

There is an offshoot connection concerning application of new students to the school, only connected to the rest of the model by the fact that their contact details have to be processed before being admitted. Since each student needs a ContactPerson, these are also related to the ContactDetails entity, but note that they despite their name aren't extensions of Person under this model. Another branch is the one from Student, regarding the management of rental Instruments.

Lessons are related to TimeSlots as well as Instructors to facilitate booking. Matters concerning payment, both the fees collected from students as well as the commissions paid to instructors, can be derived from other entities' attributes, and so don't exist outright in the model.

5 Discussion

I have tried to model the business description closely, but this has not been a trivial task and many questions remain whether I've arrived at something approaching a well executed one. These questions follow in this chapter with discussion concerning pros and cons of the decisions made.

There are no entities without attributes, a requirement for the Conceptual Model. I also went further, making sure to only include entities with at least two attributes, considering that an entity with a single attribute could be construed to be renamed to that attribute and so likely could fit within another entity or have been insufficiently explored. This took a while, but also was a natural progression of trying to reduce the model complexity to something possible to overview without becoming dizzy.

A difficult case concerning this was the ContactPerson. This ought to be an entity in its own right, being a stated business rule element and having a defined relation to ContactDetails. In the business description it didn't have any other attributes however, so to not be an empty entity some attributes needed to be invented. Hence their name and relationToStudent (optional). This solution would naturally have to be discussed with the client, since Soundgood may not at all be interested in these attributes. An alternate solution would be to allow students to have multiple relations to ContactDetails, one of whom would be the contact person's.

A business rule was that a discount should be applied for students who has a sibling enrolled at the school, and who that sibling is. But the sibling is a student in their own right, making them eligible for a discount. With the diagram tools at hand I wasn't allowed to make a circular reference. I instead mention this rule in a note attached to Student. Though this makes the model slightly more cluttered it was a necessary inclusion.

Further, the description stated that Soundgood wanted flexibility in pricing schemes, so it's not inconceivable that other discounts could be applied, perhaps not limited to percentages, which expanded the entity's attributes.

I decided on two cases of inheritance. The first case is that the three types of Lesson (Individual, Group and Ensemble) all share commonalities. They all are a Lesson in that they share the levelOfLesson, fee, instructorCommission and location as relevant attributes. Lesson is therefore a generalisation of them, and they are specialisations in that they have unique attributes respectively. I could have gone further here, since there is still some duplication in the attributes of GroupLessons and EnsembleLessons, but felt that grandinheritance would have increased the complexity of the diagram more than it would have helped.

The other case of inheritance is that both Students and Instructors are Persons. Persons is the general case, where they share the need for the attributes givenName, surName and personalNumber. The Student and Instructor do however have opposed roles and therefore to no great surprise require different additional attributes.

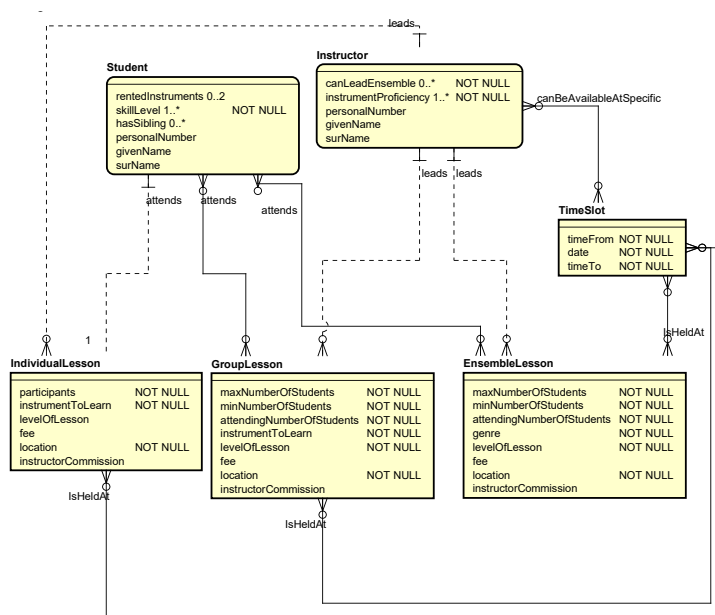


Figure 5.1: The case without inheritance is cluttered

An interesting case connected to the second inheritance was ContactPerson, who could be argued to also be an extension of Person. However, nowhere is it stated that their social security details need to be managed, and just what goes into the concept of ContactDetails would have to be discussed with the client before reaching a decision on this.

An issue from a database perspective was that of derived data. I removed both an entity StudentPayment and InstructorComission, each of which simply contained the

attribute amount, since these could clearly be derived by data already existing in the system. Another attribute on the verge of removal is fee in Lesson, considering the business description that the fee is dependant with respect to the skill level of each lesson and its type. In a sense the fee can be derived from the mere existence of a Lesson and its assigned levelOfLesson, however I decided to keep the attribute in since neither of these parameters give any data on the actual value of the fee.

Another case on the chopping block is the School entity itself. It survives on the periphery on the merit that there needs to be data on if applying students can be offered a spot.