Sample Problems and Solutions for Modeling Quiz: Class Diagrams
CS 2720

Sample Problems

1. Answer the following questions regarding the diagram (and underlying model) given below.

(a) What are the attributes of a child?
(b) What are the operations of a child?
(c) Give the UML term for the Student Goal class and describe its meaning in the underlying model.
(d) Describe the relationship between a daycare center and a class. Include the name of the UML relationship and the meaning of the relationship.
(e) What can you say about the operations a teacher can perform?
(f) What is the minimum number of children required for a class?
(g) How many classes can a teacher teach?
(h) For each relationship with an explicit multiplicity, give the multiplicities using English sentences.
(i) Do you agree with the decision to make Person an abstract class? Why or why not?
2. Identify two “diagram smells” in the diagram below.

3. Create a class diagram for the model summarized below.

Consider a domain model for higher education. Faculty members can be one of two types: tenure-track professors and tenured professors. All professors have a name and all can be asked to serve on a committee. Only tenured professors can be asked to make promotion decisions.

Professors teach 0 to 3 courses each semester. Each course has between 8 and 100 students. Students have a name and can be asked to check their e-mail. For each course a student takes, we keep track of their attendance and scores. Students can be asked to complete assignments for each course they are registered for.
Solutions

You are strongly encouraged to also watch the “solution walkthrough” I have posted on eLearning. This walkthrough contains me talking about the problems and their solutions in more depth, as well as making some general comments about the quiz.

1. Sample answers for the class diagram of a daycare center model:
   
   (a) A child’s attributes are age and name.
   
   (b) A child’s operations are play() and getStats() (the parenthesis are optional in your answer).
   
   (c) The Student Goal class is an association class. Each instance contains information and operations that are specific to the relationship between a single child and a single class. In the model, it is saying each child may have zero or more student goals for each class they are in.
   
   (d) A daycare center is related to a class by composition – a daycare center is composed of 0 to 10 classes. The composition relationship means a class cannot exist without being associated with exactly one daycare center.
   
   (e) A teacher can evaluate, getStats, and do all the operations a reporter can do (aka can do the operations defined in the reporter interface).
   
   (f) A class requires at least 2 children.
   
   (g) A teacher may teach 0 or more classes.
   
   (h) • A daycare center is composed of zero to ten classes.
   
   • A class has two to twelve children.
   
   • A child has one to three guardians.
   
   • A class has one to two teachers.
   
   (i) This is of course up to you. You should make sure you defend your answer, though! I give example arguments both ways below.

   **Agree:** I agree with making Person abstract. The main question is whether or not we think a person can exist in our domain without being of a specific type (child, guardian, teacher). In this case, we would not save information about a person if they weren’t one of these three categories, so making person abstract is the appropriate decision.

   **Disagree:** I do not think Person should be abstract. Abstract classes are meant to generalize common behavior and attributes in a class that cannot exist by itself. In this case, though, a person can certainly exist without being a teacher, a child, or a guardian. Therefore, it should be a regular class, not an abstract one.
2. Several possible “diagram smells” are given below. Note that if the “smell” is something you have inferred from your knowledge of the domain, please state so explicitly and try to justify your inference.

- The interface “CatID Holder” is not realized by anything. This means we have defined a set of operations that something must agree to do, but have not created anything that can do them. This is OK if we are specifying how the user can interact with our system, but may suggest we forgot a concrete class.

- Faculty is an abstract class that is not generalized. This means we have defined a set of common attributes and operations but cannot ever create an object with these attributes and operations. This may be OK if we are specifying something for the user to complete, but it may suggest we have missed a concrete class or that faculty should be a concrete class.

- A faculty member is composed of (in part) some publications, meaning a publication should be associated with exactly 1 faculty member. However, the diagram shows a publication may have between 1 and 5 faculty members related to it. This means either the relationship is not a composition, or that we have incorrectly labeled the multiplicities.

- Given what I know about the domain, I do not think a student or a faculty member is a subtype of an “admin” – an admin is likely to do things that a student or faculty member cannot (like reset passwords). We might need to add another class called “person” and make admin, student, and faculty be subclasses of this, or we might need to rename the admin class to something like person.

- Given what I know about the domain, the multiplicities between Student and Course might be backwards. As given, it says a student takes between 8 and 200 courses, while a course is taken by 1 to 8 students. A student is considered a student before taking 8 courses, however, and is unlikely to take 200 courses. A course is also likely to have more than 8 students.

3. I have given one possible class diagram below. Note you do not need to put attribute types in your diagram – they are left there only because Modelio doesn’t have a way to hide them (that I can find).