From UML to Relations

Carla Teixeira Lopes

Bases de Dados
Mestrado Integrado em Engenharia Informática e Computação, FEUP

Based on Jennifer Widom slides

UML key concepts

Classes

Associations

Association Classes

Generalizations

Composition & Aggregation

Constraints

Derived Elements

Classes

Every class becomes a relation

Student (SID, SName, Grade)

College (CName, State, Enrollment)

UML key concepts

~~Classes~~

Associations

Association Classes

Generalizations

Composition & Aggregation

Constraints

Derived Elements

Many-to-many associations

Add a relation with key from each side

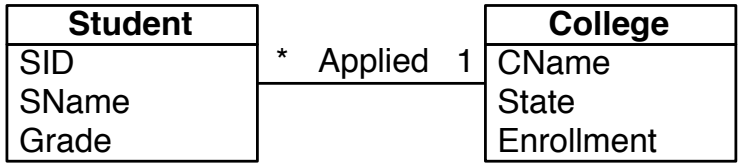
Student (SID, SName, Grade)

College (CName, State, Enrollment)

Applied (SID->Student, Cname->College)

Many-to-one associations

Add a foreign key to the **many** side of the relationship to the relation in the one side

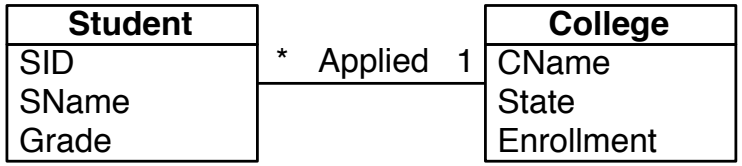


Student (SID, SName, Grade, College->College)

College (CName, State, Enrollment)

Many-to-one associations

Add a relation with key from the many side



Student (SID, SName, Grade)

College (CName, State, Enrollment)

Applied (SID->Student, Cname->College)

Many-to-one associations

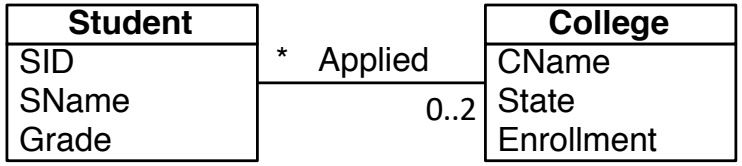
Add a foreign key to the many side of the relationship to the relation in the one side

- Most common
- Less relations in the schema
- Increased performance due to a smaller number of relations

Add a relation with key from the many side

- Increased rigour of the schema
- Increased extensibility

Question



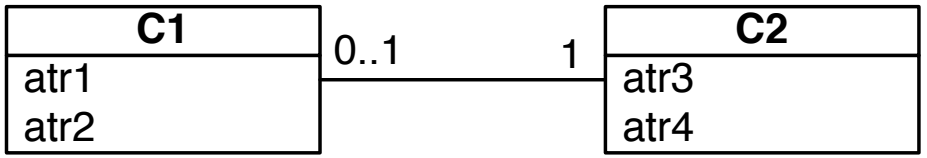
Suppose we had 0..2 on the right-hand side, so students can apply to up to 2 colleges. Is there still a way to "fold in" the association relation in this case, or must we have a separate Applied relation?

Yes there is a way

No, if it's not 0..1 or 1..1 then Applied is required

One-to-one associations

Add a foreign key from one of the relations to the other



C1 (atr1, atr2, c2_id->C2)

C2 (atr3, atr4)

Add the foreign key to the relation that is expected to have less tuples

Add a unique key constraint to the foreign key

UML key concepts

~~Classes~~

~~Associations~~

Association Classes

Generalizations

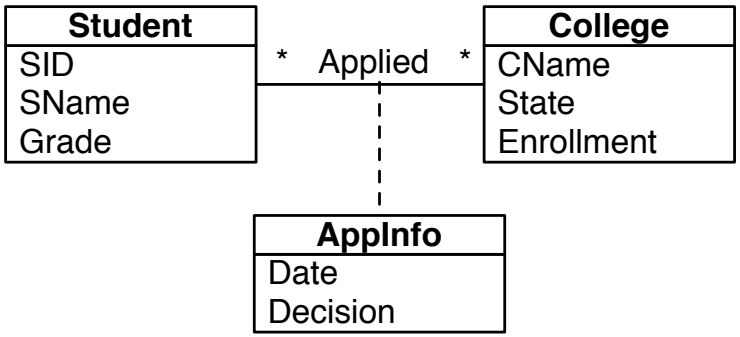
Composition & Aggregation

Constraints

Derived Elements

Association classes

Add attributes to relation for association



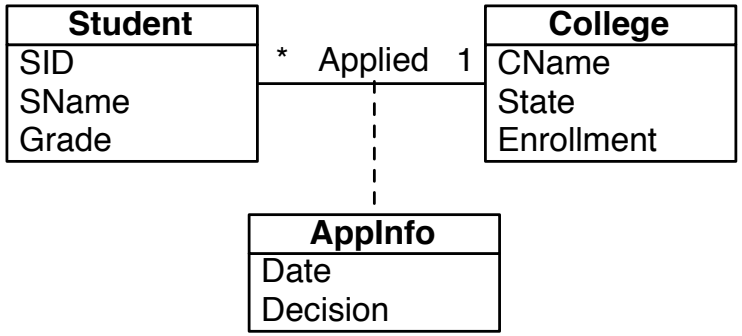
Student (SID, SName, Grade)

College (CName, State, Enrollment)

Applied (SID->Student, Cname->College, Date, Decision)

Association classes

Add attributes to relation for association

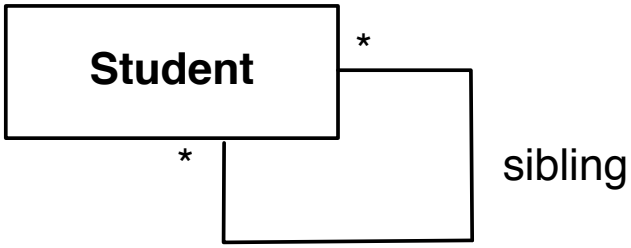


Student (SID, SName, Grade)

College (CName, State, Enrollment)

Applied (SID->Student, Cname->College, Date, Decision)

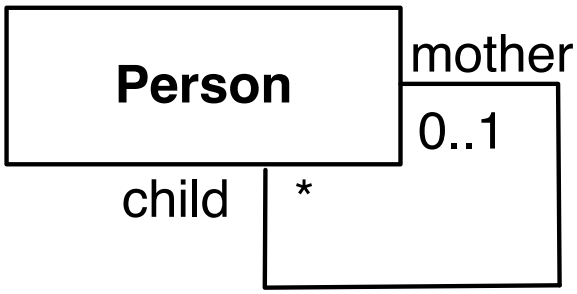
Self associations



Student (id, …)

Sibling (sid1->Student, sid2->Student)

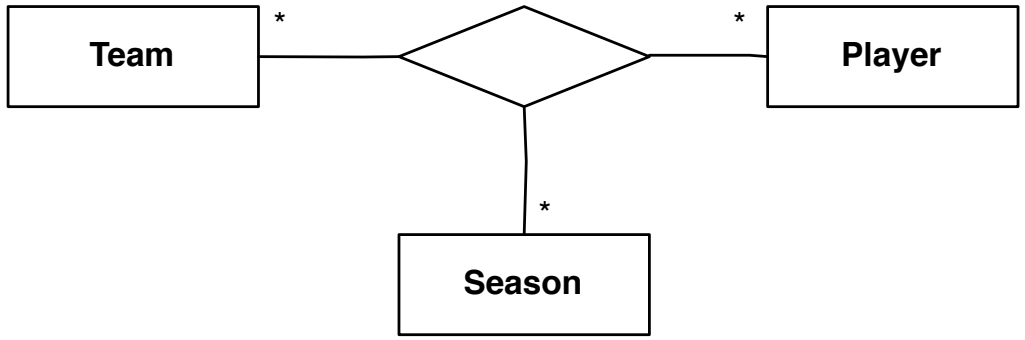
Self associations



Person (id, …)

Relationship (mother->Person, child->Person)

Associations n-ary



Relation with key from each side

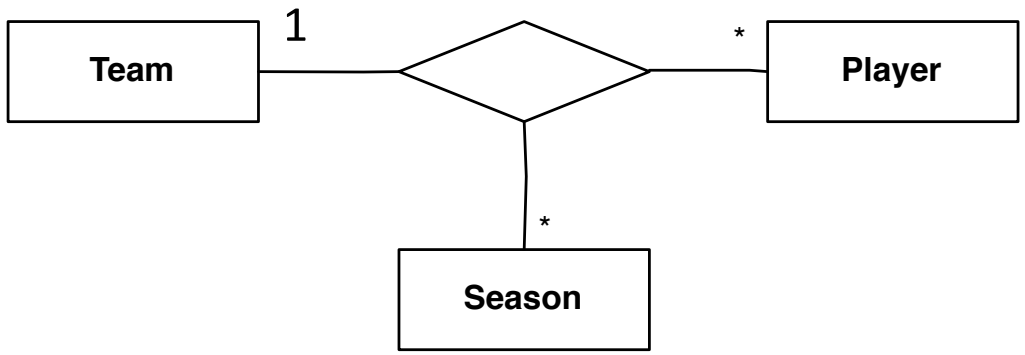
Team (ID, …)

Player (ID, …)

Season (ID, …)

PlayerSeasonTeam (PlayerID->Player, SeasonID->Season, TeamID->Team)

Associations n-ary



Relation with key from each side

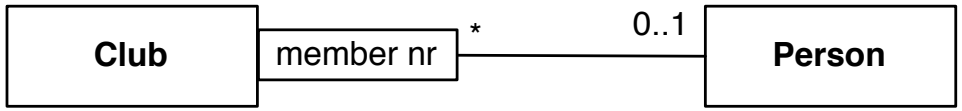
Team (ID, …)

Player (ID, …)

Season (ID, …)

PlayerSeasonTeam (PlayerID->Player, SeasonID->Season, TeamID->Team)

Qualified associations



Club (ClubID, …)

Person (PersonID, …)

Member (ClubID->Club, PersonID->Person, MemberNr)

{ClubID, MemberNr} UK

UML key concepts

~~Classes~~

~~Associations~~

~~Association Classes~~

Generalizations

Composition & Aggregation

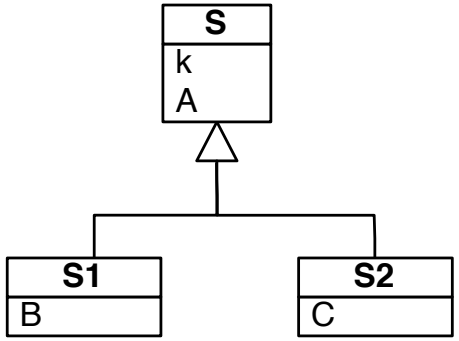
Constraints

Derived Elements

Generalizations

3 conversion strategies

- E/R style
- Object-oriented
- Use nulls



Best translation may depend on the properties of the generalization

Generalizations – E/R style

A relation per each class

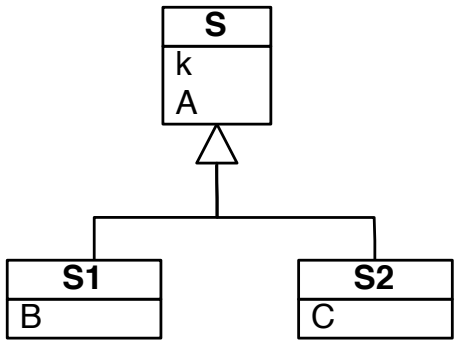
Subclass relations contain superclass key + specialized attributes

Good for overlapping generalizations with a large number of subclasses

S(k, A)

S1(k->S, B)

S2(k->S, C)



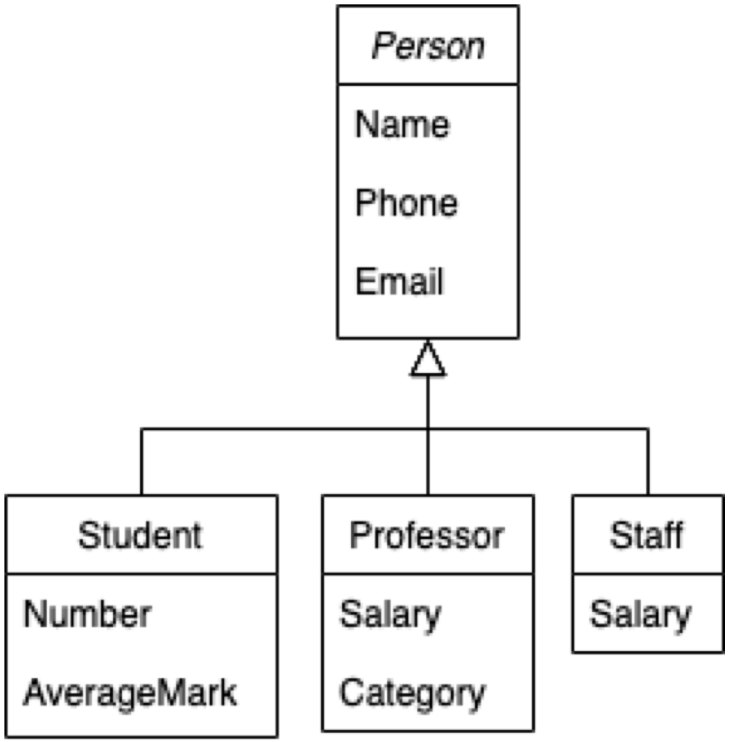
Generalizations – E/R style

Person (id, Name, Phone, Email)

Student (id->Person, Number, AverageMark)

Professor (id->Person, Salary, Category)

Staff (id->Person, Salary)



Generalizations – Object-oriented

Subclass relations contain all attributes

In complete generalizations, the relation for the superclass may be eliminated

- Cannot guarantee the uniqueness of the values of the superclass

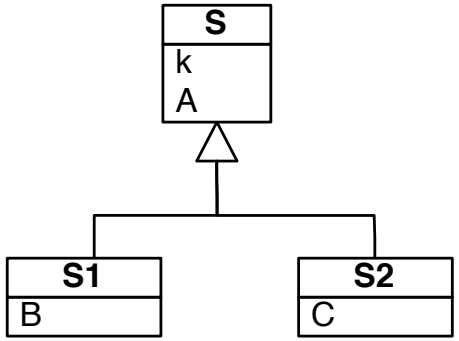
Good for

- disjoint generalizations
- superclass has few attributes and subclasses many attributes

S (k, A)

S1(k ->S, A, B)

S2(k->S, A, C)

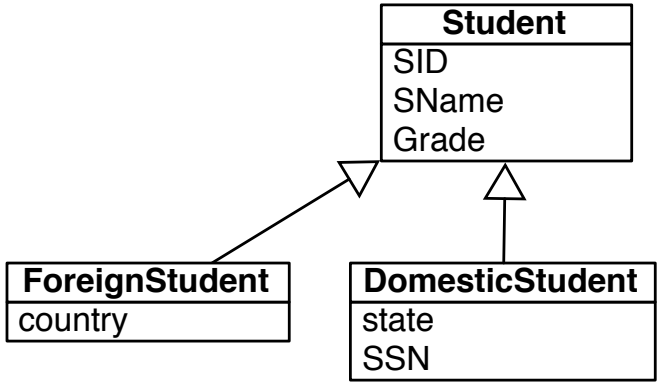


Generalizations – Object-oriented

Student (SID, SName, Grade)

ForeignStudent (SID ->Student, SName, Grade, country)

DomesticStudent (SID->Student, SName, Grade, state, SSN)



Or, because it is complete:

ForeignStudent (SID, SName, Grade, country)

DomesticStudent (SID, SName, Grade, state, SSN)

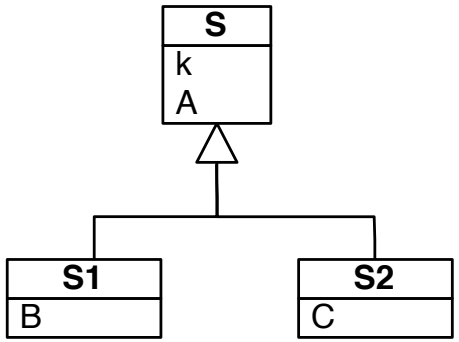
Generalizations – Use nulls

One relation with all the attributes of all the classes

NULL values on non-existing attributes for a specific object

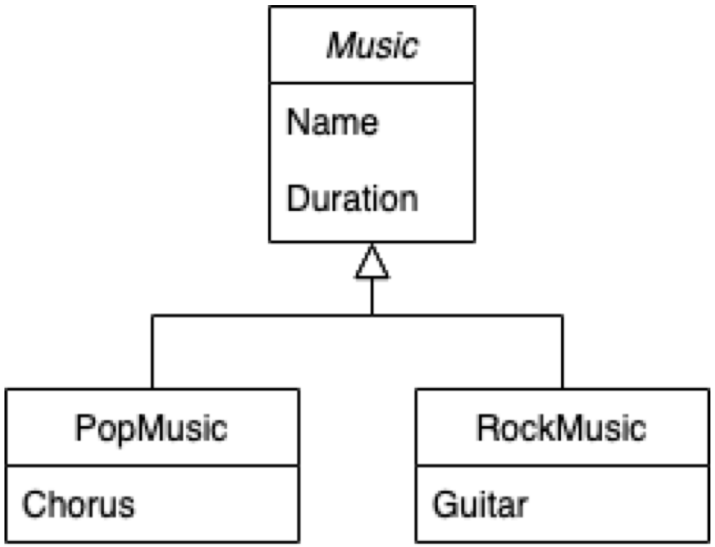
Good for heavily overlapping generalizations with a small number of subclasses

S(k, A, B, C)

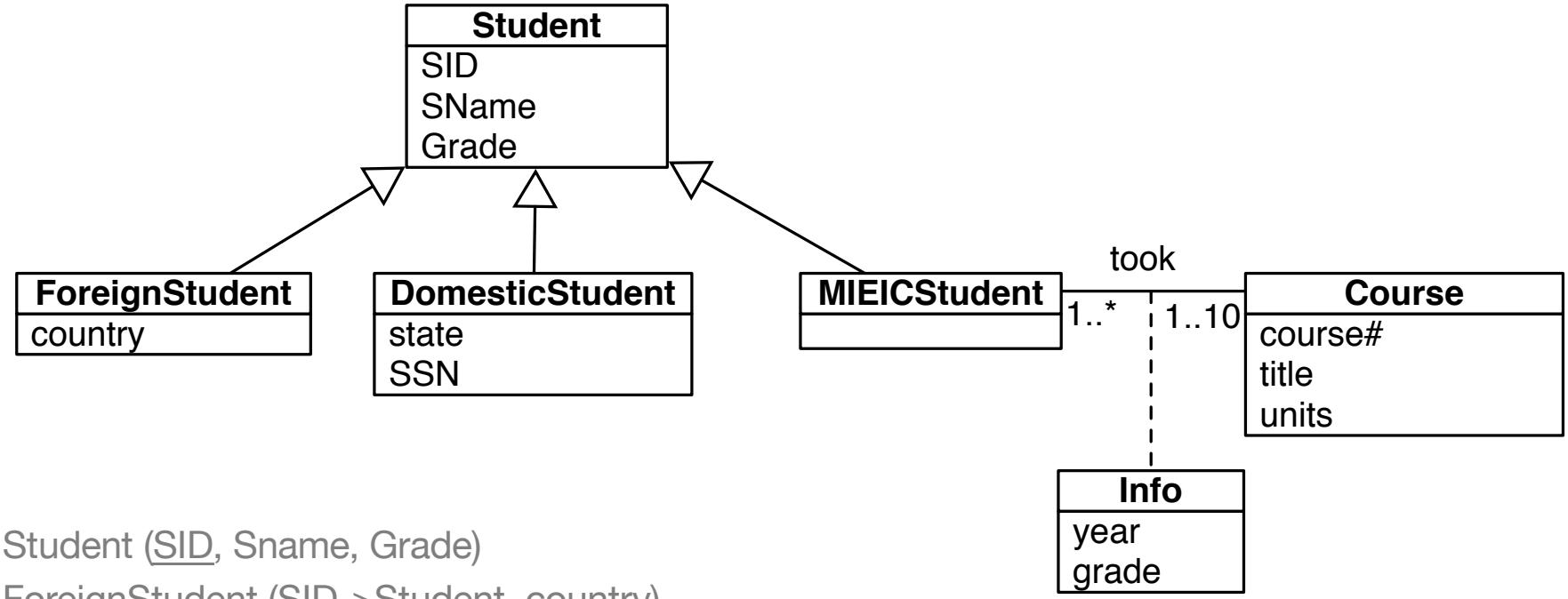


Generalizations – Use nulls

Music (id, Name, Duration, Chorus, Guitar)



Generalizations – Example



Student (SID, Sname, Grade)

ForeignStudent (SID->Student, country)

DomesticStudent (SID->Student, state, SSN)

MIEICStudent (SID->Student)

Course (course#, title, units)

Took (SID->MIEICStudent, course#->Course, year, grade)

Can we simplify the schema?

UML key concepts

~~Classes~~

~~Associations~~

~~Association Classes~~

~~Generalizations~~

Composition & Aggregation

Constraints

Derived Elements

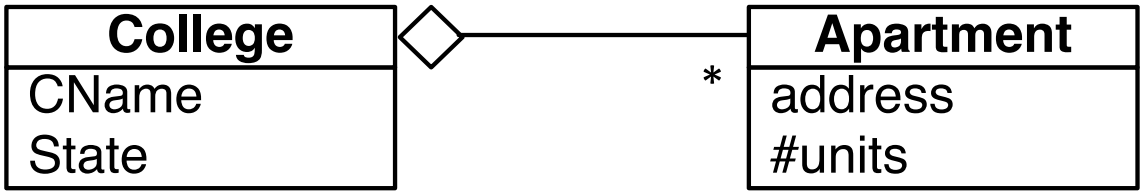
Composition

Treat it as a regular association

College (CName, State)

Department (DName, Building, CName->College)

Aggregation



Treat it as a regular association

College (CName, State)

Apartment (address, #units, CName->College)

NULL

UML key concepts

~~Classes~~

~~Associations~~

~~Association Classes~~

~~Generalizations~~

~~Composition & Aggregation~~

Constraints

Derived Elements

Constraints and Derived Elements

Constraints

- NOT NULL
- UNIQUE
- PRIMARY KEY
- FOREIGN KEY
- CHECK
  - Ensures that the value in a column meets a specific condition
- DEFAULT
  - Specifies a default value for a column

Derived Elements

- Treat them as regular elements

Kahoot time!

Any doubts?

Readings

Jeffrey Ullman, Jennifer Widom, A first course in Database Systems 3rd Edition

- Section 2.1 – Basics of the Relational Model
- Section 4.8 – From UML Diagrams to Relations
- Section 4.6 – Converting Subclass Structures to Relations