Extended Entity-Relationship (EER) Modeling

Peter Y. Wu
Department of Computer and Information Systems
Robert Morris University

Extended Entity-Relationship Model

• EER – Extended or Enhanced ER Model
• Developed in the mid 1980s… bearing also the influence of object-oriented modeling.
• The main concept is: Generalization and Specialization – one and the same concept.
• Other concepts such as Categorization and Aggregation will not be covered; these are not considered main concepts.

EER: Generalization/Specialization

Consider modeling the employees of a company, such as a personnel database system. We need to have name, employee id, address, department. (as attributes for each employee…)

• Some employees have a monthly salary, but some others have an hourly wage rate and are paid by the hours they have worked. How should we then model that?

EER: Generalization/Specialization

• Let us try two kinds of employees: salaried-employee, and hourly-employee …
• Not a good solution: because we need to support the idea of employee – for applications such as sending a letter to each employee; should not need to deal with two kinds.

EER: Generalization/Specialization

• Other reasons that it is not a good solution: if we need to identify also the employee skill as secretary or technician, we will also need two kinds for each …
• More trouble when we need to support the simple idea of employee as an entity type.

SOLUTION: Specialization

• Define a subset of an entity set, called a subclass.
• Establish additional attributes for the subclass.
• Subclass entities inherit attributes from the superclass.

EER: Specialization

• Employee is an entity type, with these attributes: name, employee id, address, department…

Specialization:

• Hourly Employee is a subclass of Employee.
• Each entity of Hourly Employee is also an entity of Employee, but has the additional attribute: hourly rate.
• Employee is the superclass of Hourly Employee.
• New ER Diagram symbol:
**EER: Subclass/Superclass**

**Formal definition:**

For every entity $e \in E_2 \Rightarrow e \in E_1$.

- That is, every entity of the subclass $E_2$ is also an entity of the superclass $E_1$.
- Thus, $E_2$ inherits every attribute of $E_1$.
- Key attributes for $E_1$ also serve for $E_2$.
- Since $E_2$ inherits attributes from $E_1$, we cannot define an attribute for $E_2$ with the same name as an attribute of $E_1$.
- A similar rule applies to relationships.
- The key defined for $E_1$ applies also to $E_2$; we cannot define another key for the subclass. (But possibly an alternate key.)

**Example:**

For every entity $e \in E_2 \Rightarrow e \in E_1$.

- That is, every entity of the subclass $E_2$ is also an entity of the superclass $E_1$.
- Thus, $E_2$ inherits every attribute of $E_1$, and every relationship involved with $E_1$.
- Key attributes for $E_1$ also serve for $E_2$. (But possibly an alternate key.)

**EER: Specialization example**

- Every hourly employee ($e \in E_2$) is also an employee ($e \in E_1$).

**Example:**

Every hourly employee ($e \in E_2$) is also an employee ($e \in E_1$).

Specialization:

- Every hourly employee also has name, employee id, ….
- An hourly employee has the hourly rate (additional attribute).
- Employee id serves as key for all employees; it also serves for all hourly employees.
- If an employee is related to some projects (to work on), the same applies to an hourly employee.

**EER: Specialization Constraints**

- A salaried employee is NOT an hourly employee, and an hourly employee is NOT a salaried employee; the subclasses are disjoint.

**Example:**

- Every employee is either salaried or hourly; the superclass entity employee has total participation in the specialization.

- That is, $E_1 \Rightarrow e \in E_1$.
- For every entity $e$, $e \in E_1 \Rightarrow e \in E_11$ or $e \in E_12$.
- That is, $E_11 \cup E_12 = E_1$. 
EER: Specialization Constraints

- Some employees are technicians but not every one; the superclass entity employee has partial participation in the specialization.
- E1 has partial participation in the specialization into E2. 
- There may exist some entity e ∈ E1 such that e ∉ E2.

EER: Specialization Constraints

- A salaried employee may also be a technician; the two entity sets – subclasses of entity set Employee overlap.
- E11 and E12 are overlap. 
- There may exist entity e ∈ E1 such that e ∈ E11 and e ∈ E12. 
- That is, E11 ∩ E12 ≠ φ.

EER: Specialization Constraints

- Specialization constraints NOT specified…
- Quite possibly the two subclasses overlap, but the designer did not specify. The two subclasses were derived from two different paths of specialization.

EER: Specialization Constraints

- Example: total participation in overlapping subclasses…
- Every employee is one or more of the following: a technician, a secretary, or a manager – and these are overlapping subclasses.
- Overlapping subclasses will also lead to the concept of multiple superclasses of an entity type. (why?)

EER: Specialization Constraints

- Specialization of an entity set leads to subsets of the entity set; these are called subclasses of the superclass.
- The participation of the superclass in the specialization may be total or partial. (Do not confuse with participation constraint in a relationship!)
- The subclasses in the specialization may be disjoint or overlap. (Note: overlap may lead to complications.)
- A subclass as an entity set can be specialized further into other subclasses.
- But overlapping subclasses may specialize into the same subclass entity set, having multiple superclass entity sets.
- Consider the intersection of two overlap subclasses…
- It is a subclass which inherits from more than one superclass, attributes as well as relationships.
- It is called multiple inheritance.
**EER: Specialization**

Example of multiple inheritance...

- university member

- employee

- student assistant

- social sec no.

- wage rate

- student

- employee

- university

**EER: Generalization**

- For specialization, we start with the superclass.
- Specialization creates subsets – subclasses of the superclass.
- **Generalization** is the reverse of specialization, we start with the subclasses.
- Generalization formulates a collective way of characterizing the concept of all subclasses as an entity set, that is, the superclass.

**EER: Generalization Example**

Given trucks, and cars, as two entity sets...

- Vehicle

- Truck

- Car

- license no.

- max load

- max passengers

- license no.

- license no.

- license no.

**Understanding EER Model …**

- For every entity $e_2$ of $E_2$, $e_2$ is also an entity of $E_1$. Therefore, $e_2$ has values for all these attributes $A_1$, $K$, $A_2$.
- Attribute $K$ is key for $E_1$; it can also serve as a key attribute for $E_2$.
- However, an entity $e_1$ of $E_1$ may or may not be an entity of $E_2$.

**Understanding EER Model …**

- For every entity $a$ of $A$, there is an entity $c$ of $C$ such that the entities $a$ and $c$ are related by relationship $R_2$; for every entity $c$ of $C$, there is an entity $d$ of $D$ such that the entities $c$ and $d$ are related by relationship $R_1$.
- Therefore, $D$ has at least as many entities as $A$.