More Enhanced Entity Relationship Models

CS157A

Chris Pollett

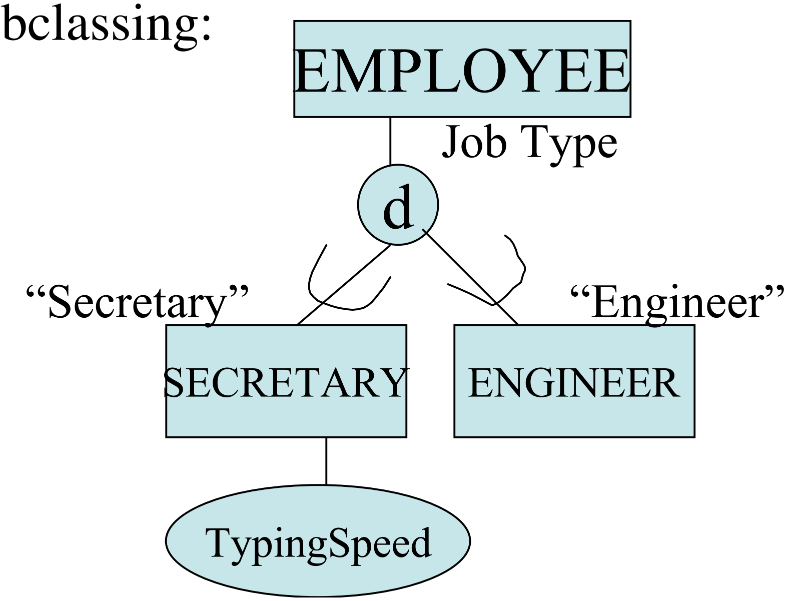
Sept. 14, 2005.

Outline

- More on Specialization and Generalization
- Union Types
- Formal Definitions
- UML variants of EER
- n-ary Relations
- Ontologies

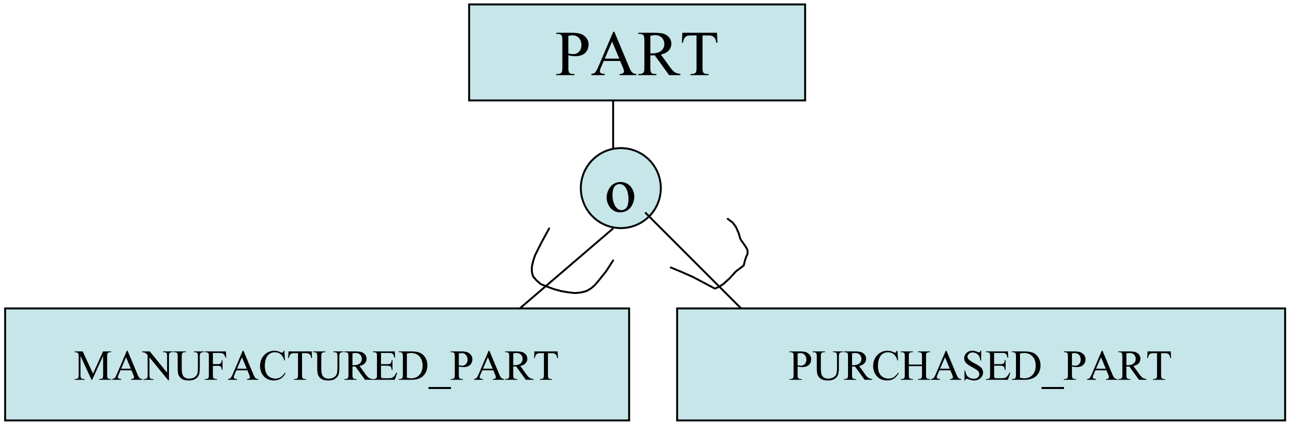
More on Specialization and Generalization

- Creating a subclass by placing a condition (for example, job description=secretary) on the value of some attribute of the superclass is called **predicate defined subclassing**.
- If all subclasses in a specialization are defined this way, then we say the specialization is an **attribute-defined specialization**.
- EER Notation for attribute defined specialization adds the attribute info to the lines in the subclassing:



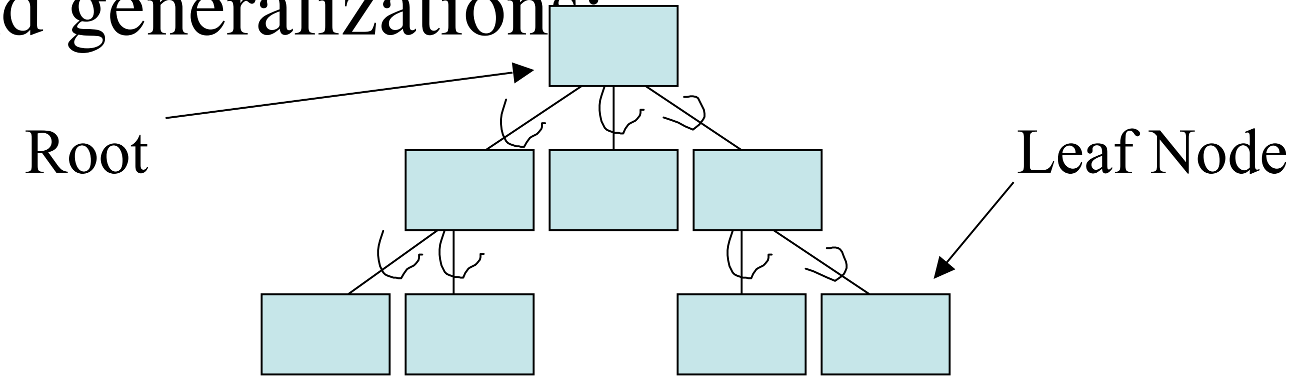
Yet More on Specialization and Generalization

- If the specialization cannot be automatically generated as above, it is called a **user-defined** specialization.
- In addition to disjoint subclasses we've already seen denoted with the d in a small circle one can also have overlapping specialization where the subclasses are not disjoint. This is denoted with an 'o'.

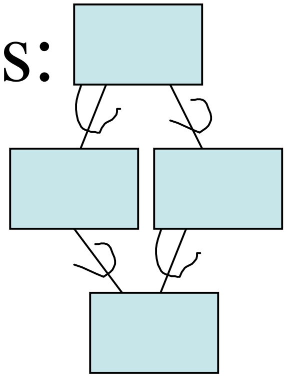


Hierarchies and Lattices

- Can create hierarchies of specialization and generalizations:



- Get a lattice if one subclass multiply inherits from two superclasses:

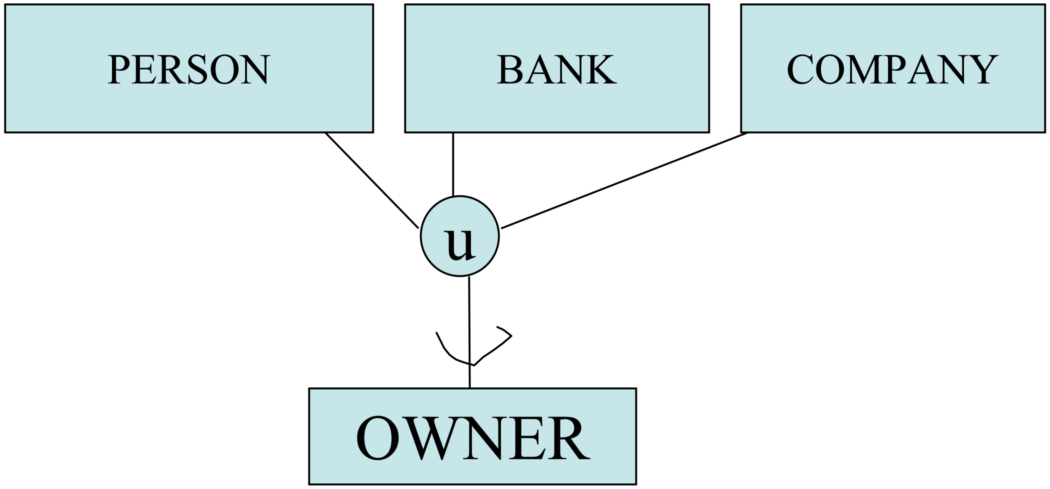


EER Schema Design

- Two approaches:
  - One can start with high level classes such as PERSON and specialize these into finer and finer subclasses such as STUDENT, TEACHER, GRADUATE, UNDERGRADUATE, etc. This is called a **top-down conceptual refinement process**.
  - The other approach is the reverse to start with specific classes and try to make more and more general classes out of them. This is called **bottom-up conceptual synthesis**.

Union Types (aka Categories)

- The owner of a vehicle might be a person, a bank, or a company.
- So if we have classes PERSON, BANK, COMPANY, we could subclass all of the them together to make a class OWNER. This is a called a **union type**. OWNER, might have its own 'new' attributes. It also shares the attributes of one of PERSON, BANK, or COMPANY, but not neccesarily the other two.
- It is drawn like:
- Like specializations/ generalizations it can also be total or partial.



Formal Definitions

- It is useful to have formal definitions in order to fix our understanding about the EER model.
- In EER, a **class** is a set of entities.
- A **subclass** is subset of entities of from a class.
- C is a **superclass** of S if C is a class and S is a subclass of C.
- Z={S_1, S_2,…S_n} is a **specialization** of G if each S_i is a subclass of G. G in this case is said to be a **generalization** of Z.
- Z is **total** if $\bigcup_{i=1}^{n} S_i = G$.
- Otherwise, Z is **partial**.
- Z is **disjoint** if if we always have $S_i \cap S_j = \emptyset$

  for i not equal to j. Otherwise, Z is **overlapping**.

More Definitions

- A subclass S of C is **predicate-defined** if a predicate p on the attributes of C is used to specify which entities in C are in S.
- We write this as S=C[p].
- If S is not predicate defined it is **user-defined**.
- A specialization Z of G is attribute-defined if a predicate $A=c_i$ for some attribute A of G and c_i a constant is used to specify membership in each S_i.

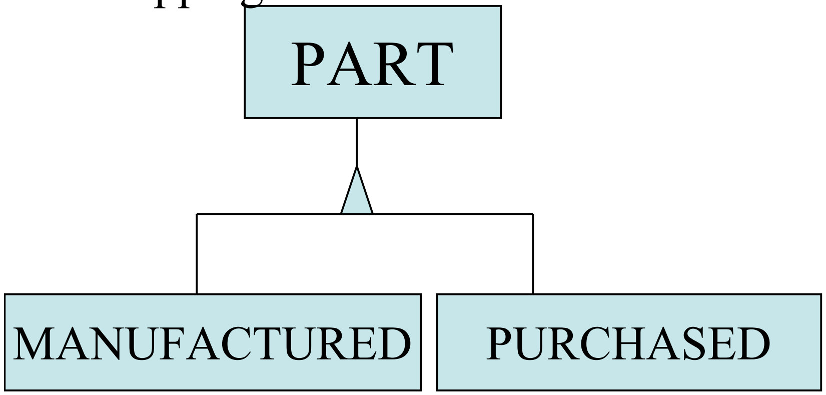
Yet More Definitions

- A **category** T is a class that is a subset of the union of n defining superclasses. That is: $T \subseteq (D_1 \cup D_2 \cup \ldots \cup D_n).$

UML variants of EER

- We’ve already seen how to map ER diagrams to UML diagrams
- You can model disjoint and overlapping specializations/generalization in UML as:

Overlapping

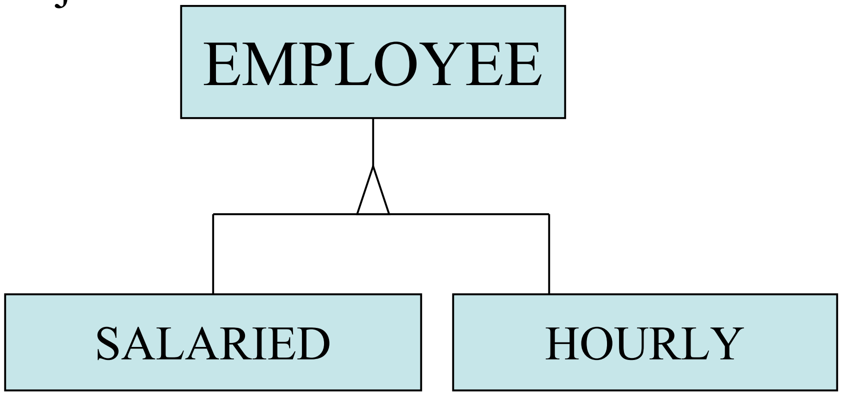


PART

MANUFACTURED

PURCHASED

Disjoint



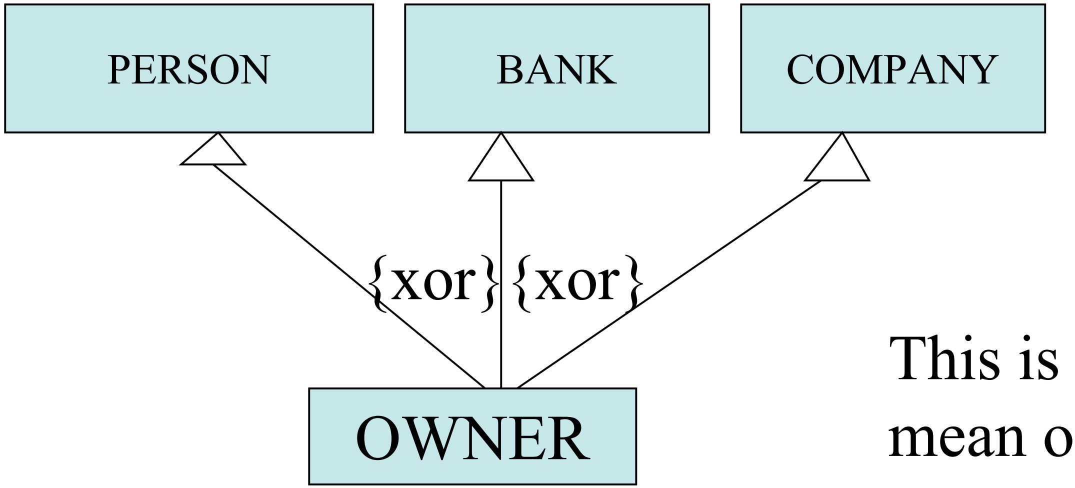
EMPLOYEE

SALARIED

HOURLY

UML for Union Types

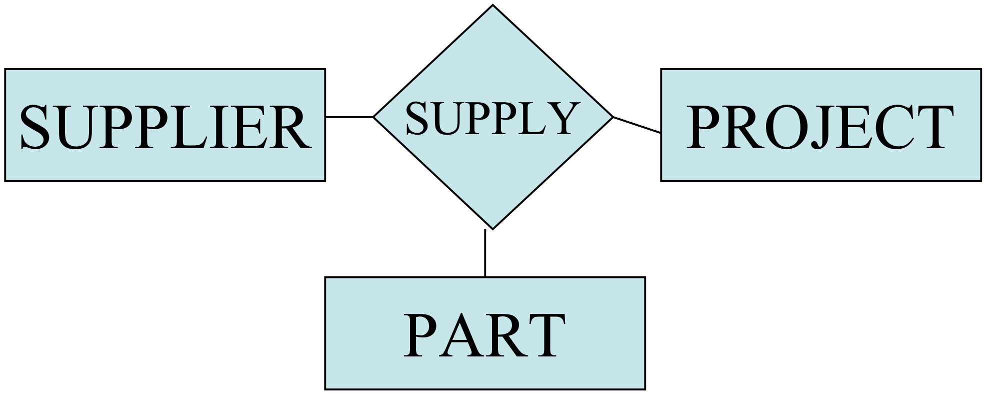
- Not standardized can do things like:



This is supposed to mean one of the three, not parity

n-ary Relations

- Recall the degree of a relationship type is the number of participating entity types.
- We usually deal with binary relationship types.
- But can have ternary or n-ary relationship types:



More on n-ary Relations

- Not all design tools support n-ary relations.
- So one is left with trying to split the n-ary relations into binary ones or use weak entity types.
- For instance, SUPPLY above might be split into SUPPLIES(SUPPLIER,PROJECT), CAN_SUPPLY(SUPPLIER,PART), and USES(PROJECT, PART).
- In 3D we have projected down onto the three canonical 2D planes. It won't necessarily be the case that we can reconstruct the SUPPLY object from this unless some additional constraints are satisfied by SUPPLY.
- In situations where such constraints don't apply we are better off if we can use a ternary or any relationship type.

Constraints on Ternary Relationships

- Constraints are indicated by putting one of 1, N, M on the arcs in the ternary relationship type in a diagram.
- A typical element of such a relationship R is a triple (s,j,p). If s can occur in only one such triple we would put a 1 on that arc; otherwise, we use an N or an M. Etc.

Ontologies

- The **Semantic Web** is a web project which attempts to create knowledge representation models for information on the web.
- A model for some domain of knowledge is called an **ontology** in the AI community.
- Many of the same conceptual modeling processes we have talked about are applicable when coming up with ontologies.
- In particular, in both settings we have a notion of classification and instantiation, identification, specialization and generalization, and aggregation and association.

Techniques to Describe Ontologies

- **Thesaurus** - describe the relationship between works that represent various concepts.
- **Taxonomy** - describes an area using structures similar to generalization specialization.
- **Database Schema -** what we're talking about this semester
- **Logical Theory -**describes an area using logical axioms.

Some XML based languages for ontologies have been proposed using RDF. For example, OWL.