ER Model was later extended

- In many applications, entity subgrouping is important.
- Borrow *Generalization / Specialization* from the Object-Oriented Model.

Subclass / Superclass

- A subclass captures *special cases* of a parent (super-) class.
- So, typically, it has less number of instances.
- A subclass *inherits the attributes* of its superclass.

Subclass / Superclass

- A subclass is useful when it either
  - has special attributes (not in the parent); or
  - is involved in special relationships (unlike the parent);
  - or both.

Subclass / Superclass

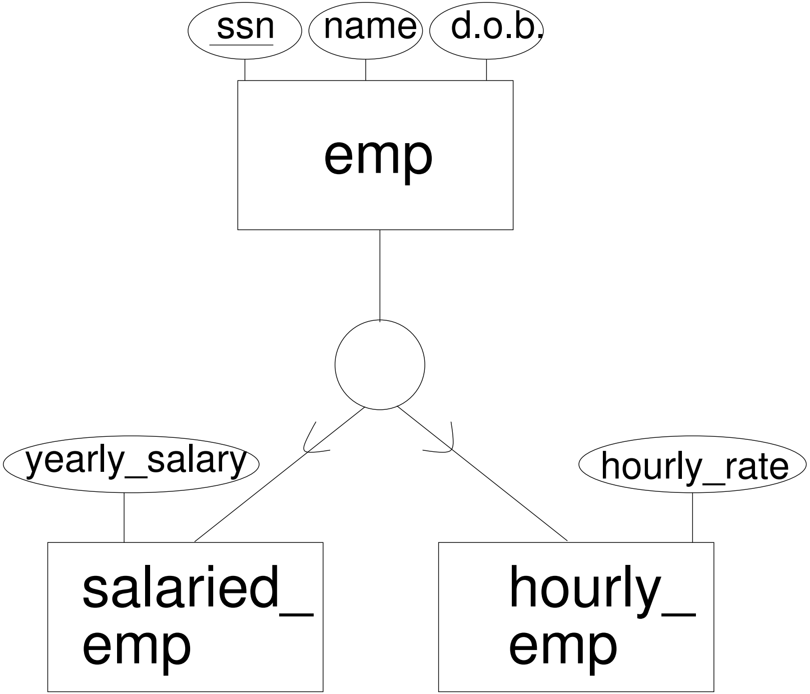
- A subclass is useful when it either
  - has special attributes (not in the parent);
    - ☞ Owing to inheritance of attributes, only those attributes special to the subclass need to be shown.

    or
  - is involved in special relationships (unlike the parent);
  - or both.

Subclass / Superclass

- A class can have many subclasses
- Most often, the subclasses form a tree.
- Denote by a circle and superset symbols on the arcs towards subclasses.

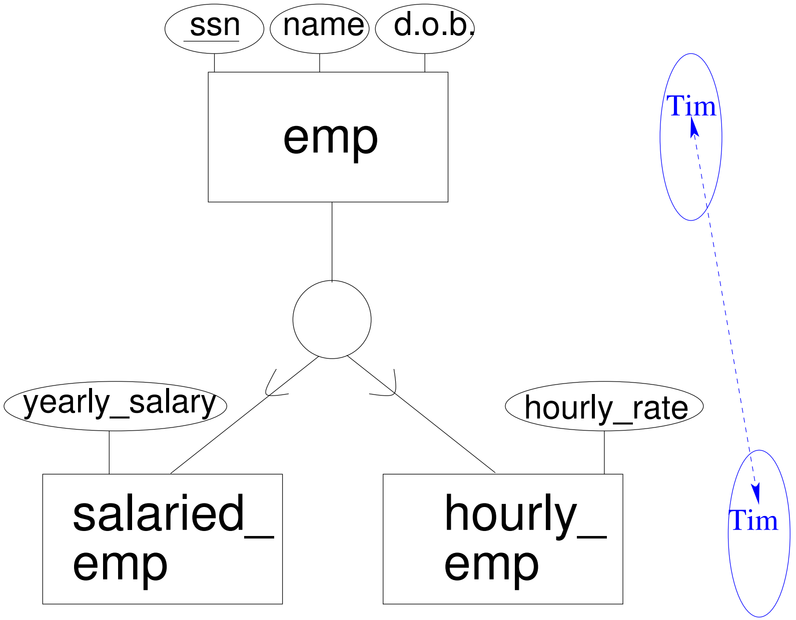
Subclass / Superclass



Accommodating Generalization

- Entities have representatives in all subclasses to which they belong.

Simultaneous existence



Keys in Hierarchies

- In an is-a hierarchy, only the root entity has a key (identifying attribute(s)),
- and it must serve as the key for all entities in the hierarchy.

Existence in subclass

- An entity cannot exist in a subclass and not in a superclass.

Insert / delete

- Delete instance from superclass ⇒ delete from subclass where the instance exists.
- Insert into superclass ⇒ insert into appropriate subclass if any (at least one in the case of total coverage).

Attribute-defined specialization

- Specialization may be based on a defining attribute.
- Example:
  emp may have an attribute EmployeeType; the value of that attribute is hourly or salaried when the instance belongs to hourly_emp or salaried_emp respectively.

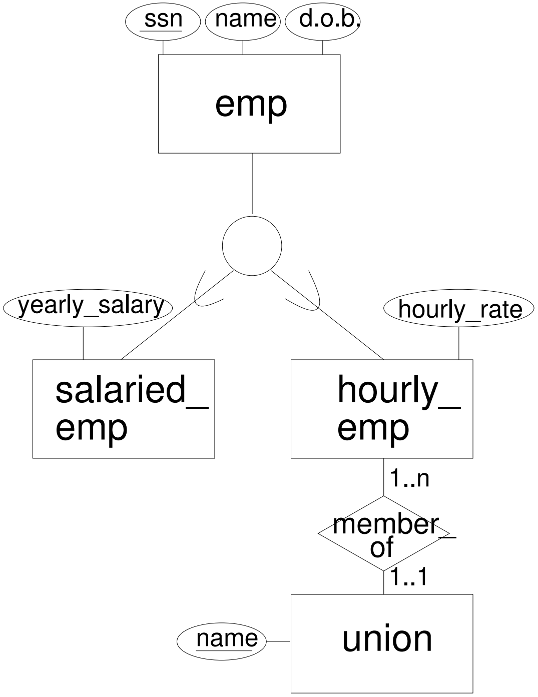
Predicate-defined specialization

- Specialization may be based on a predicate.
- Example: All employees less than 20 years old are hourly employees; otherwise, they are salaried.
- Attribute-defined implies predicate-defined. Why?

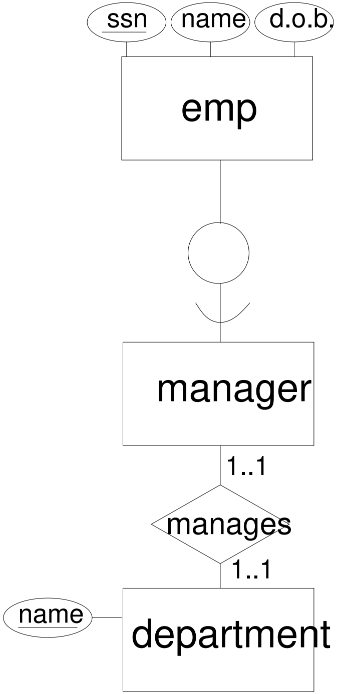
User-defined specialization

- If no such predicate is definable, then the generalization is *user-defined*.

Specialized Relationships



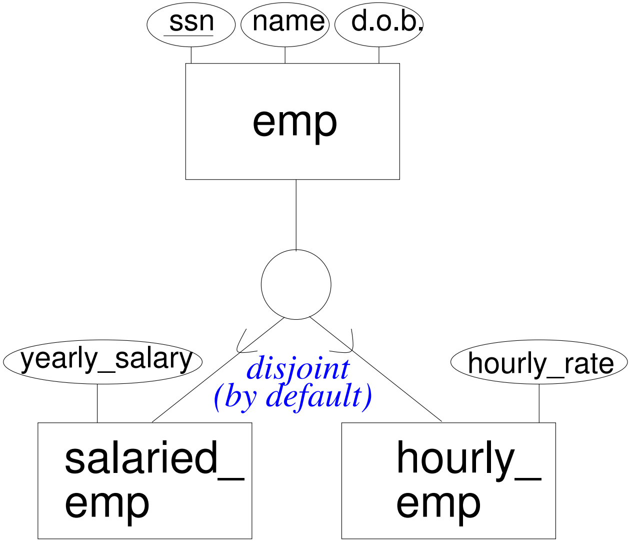
Specialized Relationships



Disjoint subclasses

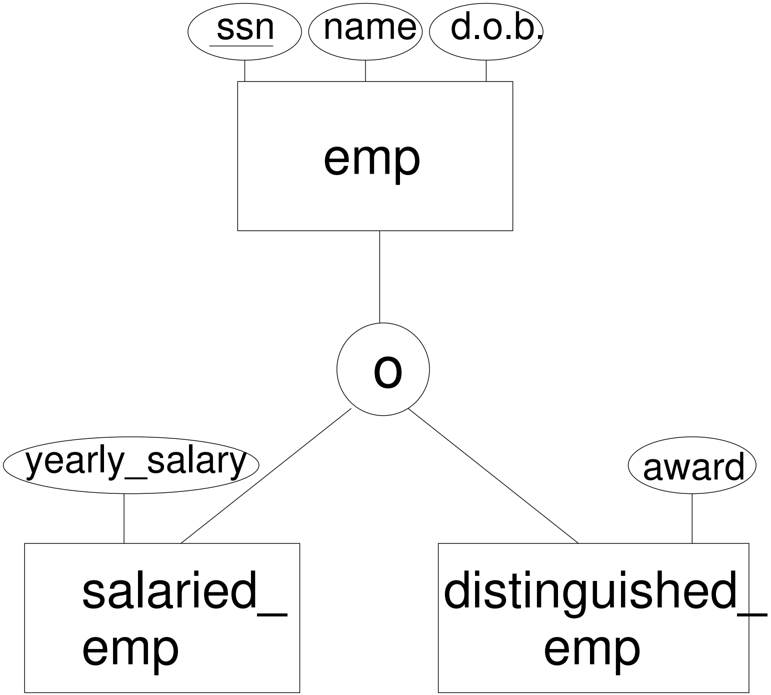
- Subclasses are *typically* disjoint from each other but *not necessarily*.
- Disjoint is the default.
- Overlap must be explicitly specified by an *′o′* within the specialization construct (circle).
- It doesn't hurt to write *′d′* for disjoint!

Disjoint subclasses



(Abbreviation: *d.o.b.* = *date of birth*)

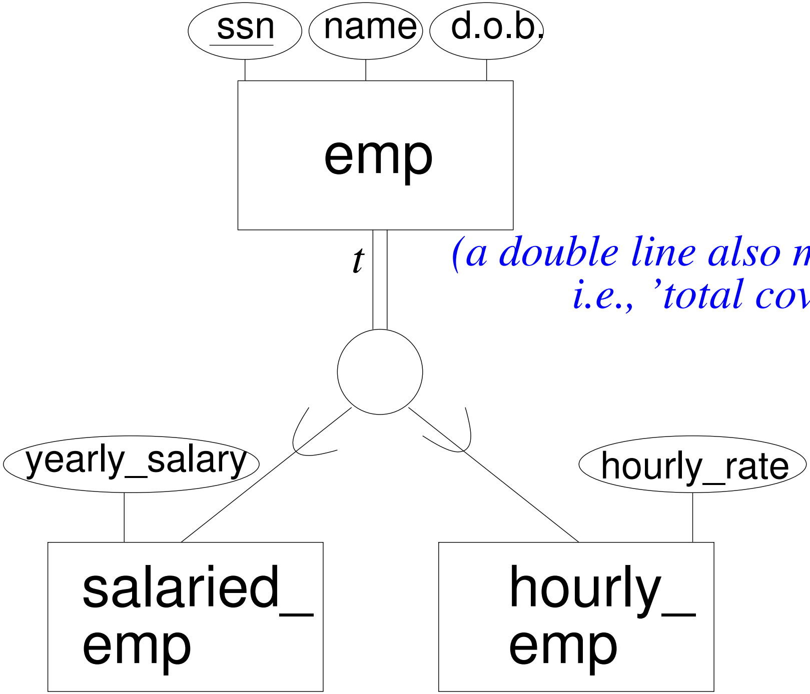
Overlapping subclasses



Total Coverage of superclass

- If every instance of the superclass belongs to one of the subclasses defined by a specialization, then that total coverage is indicated through the line joining the superclass to the specialization construct (circle): either
  - by replacing it with a *double line*; or
  - by annotating it with *t*.
- *Partial coverage* is the default. But it doesn't hurt to annotate with p!

Total Coverage of superclass



(a double line also means 't
i.e., 'total coverage')

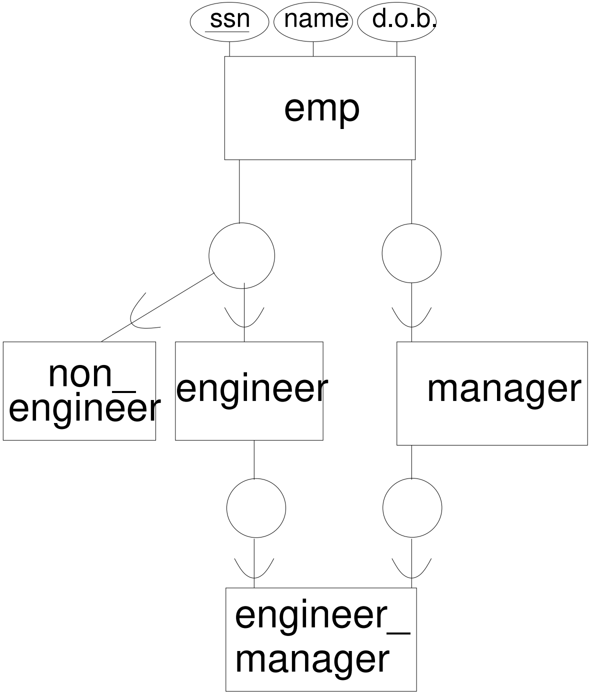
Multiple Inheritance

- Most of the time, a subclass has exactly one superclass. But sometimes, a shared subclass may have *more than one parent*.

  Such a shared subclass inherits attributes from all parents and ancestors.
  This is called *multiple inheritance.*

  Note: ultimately there is only one ancestor.

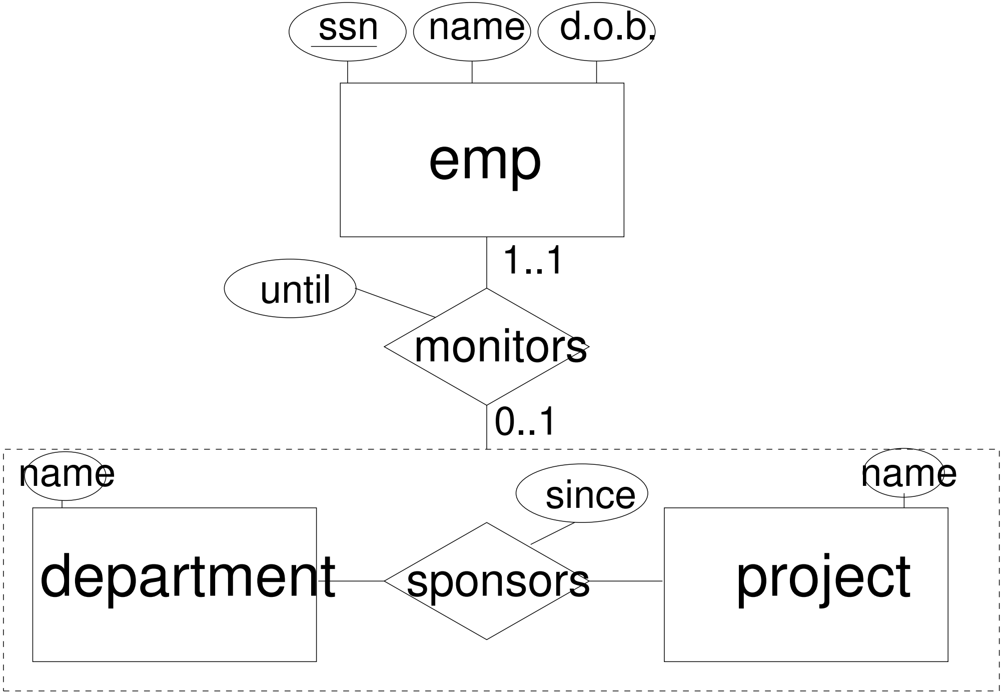
Multiple Inheritance



Aggregation

- It has been suggested that sometimes we need a relationship between two relationships!

Aggregation



Finessing Aggregation

- Usually, we can avoid aggregation by using a relationship with higher arity and careful naming of attributes.

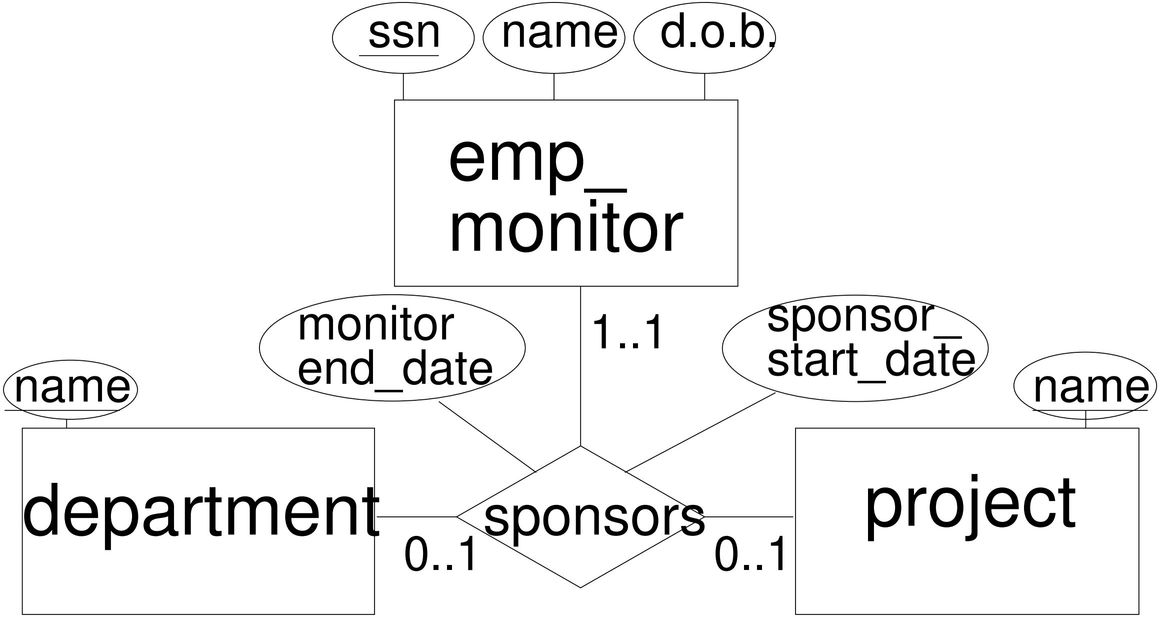
Aggregation with constraints

- Often, aggregation is accompanied by constraints;

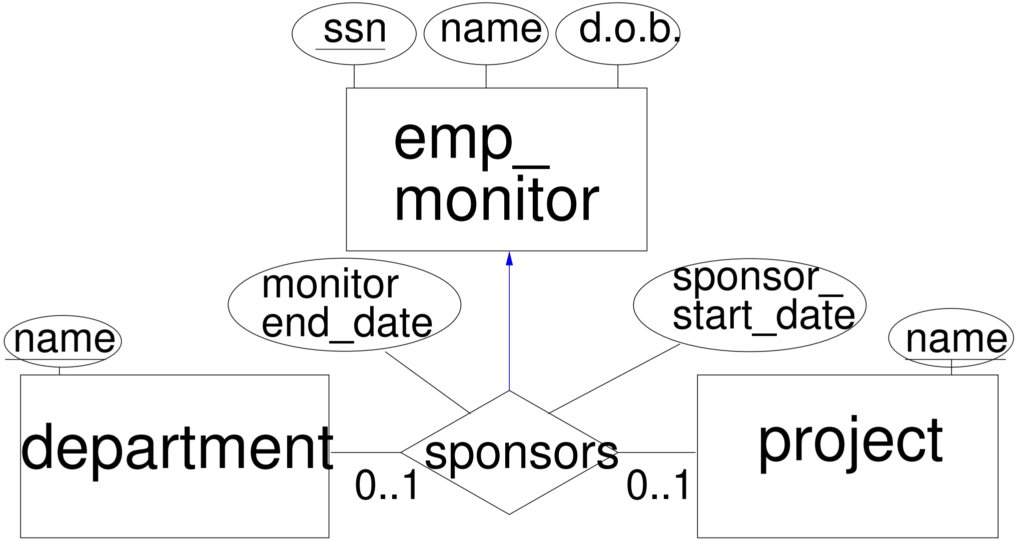
  e.g., In the above example, we are told that exactly one employee can monitor a particular sponsorship of a project by a department.

  Capture using cardinality constraints.

Aggregation with constraints



Aggregation with constraints

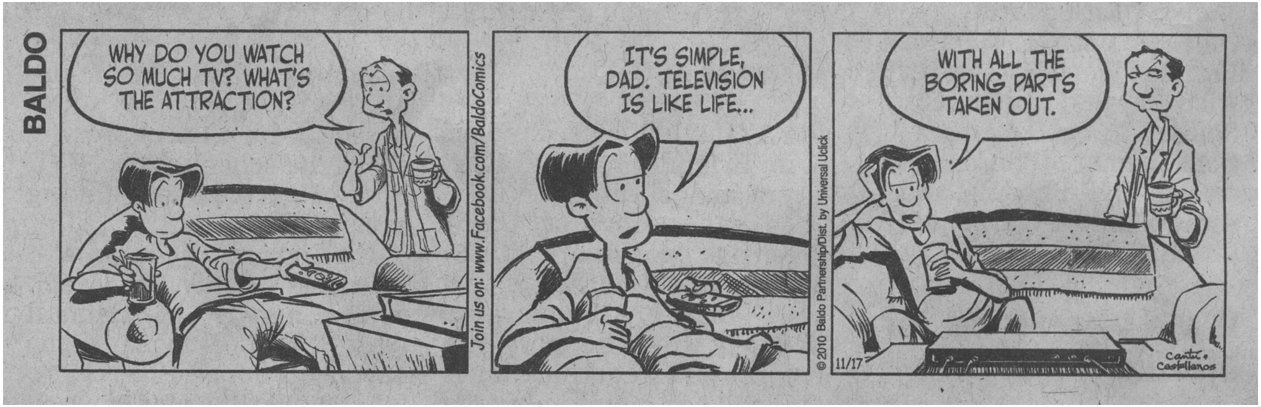


How is this different?

Additional Constraints

- maybe necessary when cardinality ratios are inadequate.
- are specified by naming entities and/or relationships they refer to.
- informally in English?
- formally using logic on the instance sets?

Model of the world


BALDO
WHY DO YOU WATCH SO MUCH TV? WHAT'S THE ATTRACTION?
Join us on: www.Facebook.com/BaldoComics
IT'S SIMPLE, DAD. TELEVISION IS LIKE LIFE...
© 2010 Baldo Partnership/Dist. by Universal Uclick
11/17
WITH ALL THE BORING PARTS TAKEN OUT.
Cantú • Castellanos