Dynamic modeling

G. Falquet, L. Nerima

Outline

- More on classes generalization and inheritance
- The link between static (class) and behavioral modeling
- Interaction diagrams
  - Collaboration diagrams
- States and state charts
- Sequence diagrams
- Exercices

More on generalization

- Generalization: taxonomic relationship
- hierarchy: parent – children (superclass – subclass)
- simplest case: a class has a single parent
- more complicated: a class has more than one parent
- the child inherits (from all its parents):
  - structure (attributes + associations)
  - behavior
  - constraints

Generalization: aims

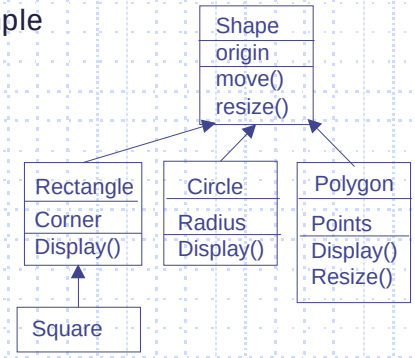
To share

- Attributs (statique)
- Operations (behavioral)

To specify that more specific classes inherits from the more general class

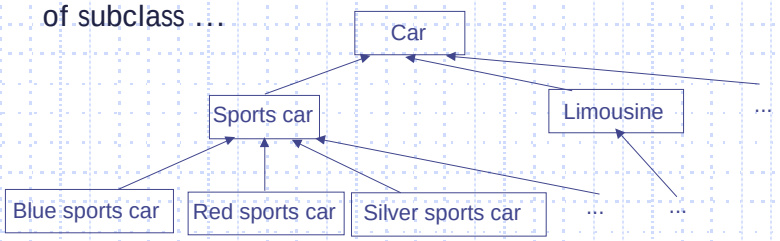
Criteria for building taxonomies

- many shared elements (structural and behavioral)
- but significative differences must exist between the children
- Example

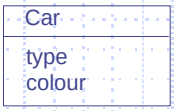


Counterexample

- In theory, every attribute can be substituded by a level of subclass …



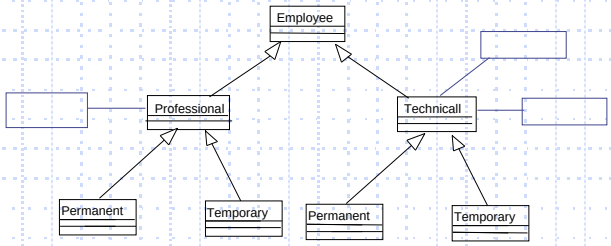
- … but probably the solution below is preferable



Generalization with covariance

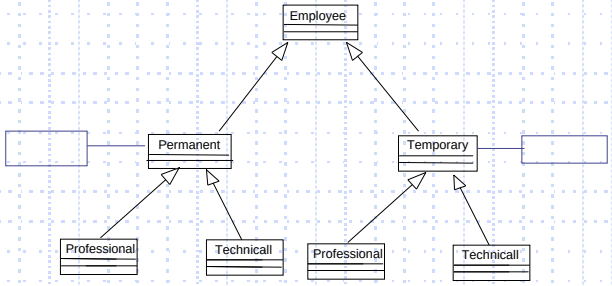
- Principle: choose the criteria that maximize common elements
- Example:

The type of employee classes (professional or technical) shares more structural elements (attributes, association,…)

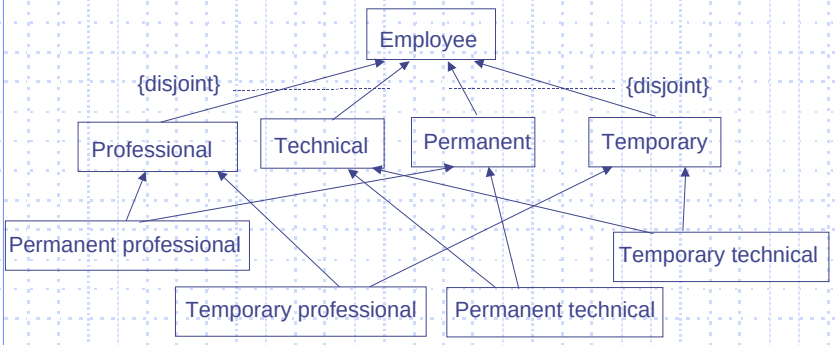


Generalization with covariance (cont)

The type of employee position (permanent or temporary) shares more structural elements (attributes, association,…)

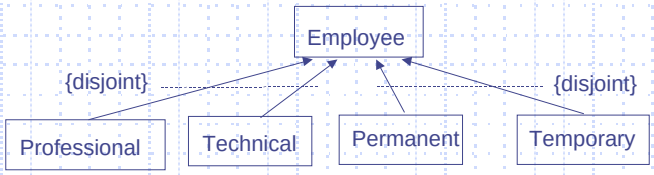


Multiple inheritance or simple heritance ?



- Expressive modeling
- Implementation: C++ supports multiple inheritance, Java does not

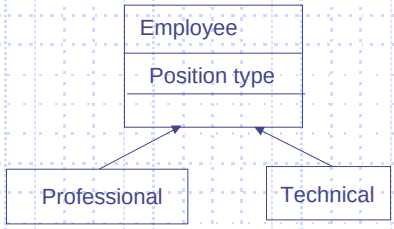
Multiple inheritance or simple heritance ?



- An instance in Employee class will have two instances in child classes: one in Professional/Technical, one in Permant/Temporary
- Implementation: difficult (no prog. language supports this feature)
- class migration not supported

Multiple inheritance or simple heritance ?

An attribute can also be solution



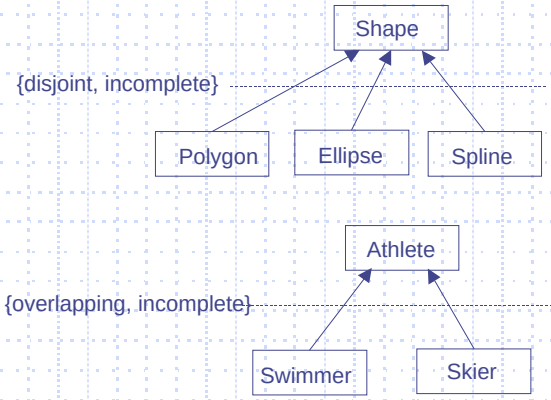
Is it a good solution ? Modeling test:
- do permanent/temporary employees participate to the same associations ?
- are the behaviors (more or less) the same, eg salary calculation, pension ?

Constraint on a set of generalizations

- disjoint
  - no element may have two children
- overlapping
  - an element may have two or more children
- complete
  - all possible children have been enumerated
- incomplete
  - all possible children have not been enumerated yet

- Default: disjoint , complete (may be redefined)

Examples

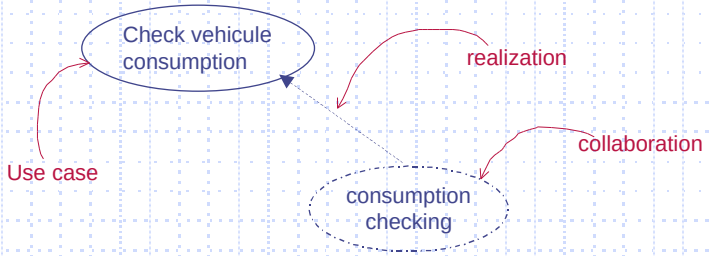


Class Diagram recapitulation

- Typical use of class diagram
  - To model the vocabulary of a system
  - To model the static view of a simple collaboration (set of classes that work together to provide a behaviour)
  - To model a logical database schema

The link between class and behavioral diagrams

- The realization of a use case by a collaboration
- An example of collaboration in AT3

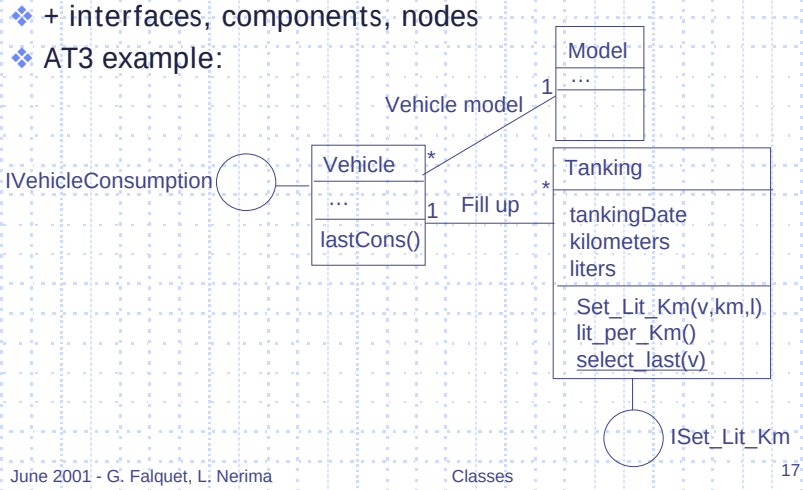


Collaboration

- A collaboration gives a name to a conceptual building block of the system
- A collaboration encompass
  - static elements
  - behavior elements
- Note: a collaboration doesn't own any of its structural elements (it references the classes, interfaces, ...)

Collaboration : structural aspect

- mainly classes
- + interfaces, components, nodes
- AT3 example:

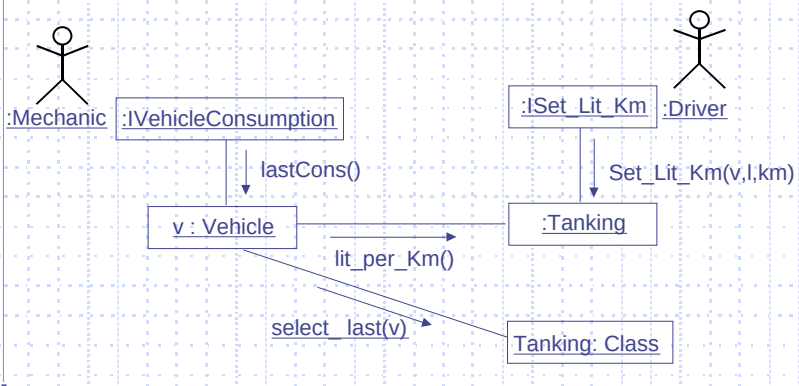


Collaboration: behavioral aspect

- Use an interaction diagram
  - collaboration -> emphasis on structural relationship of collaborating objects
  - or sequence -> emphasis on time ordering of messages

Behavioral aspect of collaboration

- Example of collaboration diagram



Behavioral aspect of collaboration

- Example of collaboration diagram