Constraints

Object-Oriented Analysis and Design

CSCI 6448 - Fall 1998

Kenneth M. Anderson

Goals of the Lecture

- Describe Constraints
  - How are they used in design?
  - How are they specified in UML?
  - Provide examples of their use

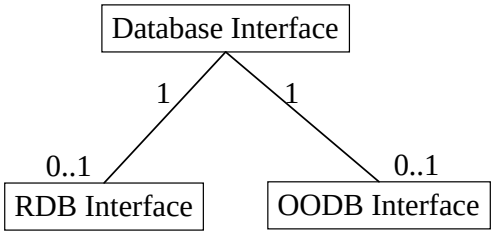
Why do we need constraints?

- Graphical elements of a diagram
  - Take advantage of human visual system to convey a lot of information quickly
  - Good for capturing basic semantics
  - Slippery Slope: Users will sometimes try to specify something the notation wasn't meant to address
  - Poor at capturing meta-information

Constraints

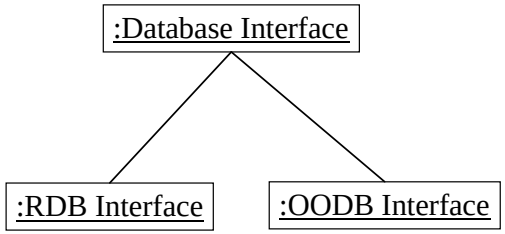
- Allow a designer to step outside of the basic abilities of the graphical notation
  - Meta-Information can be easily specified
  - Ambiguities can be eliminated
- Can be expressed using a formal language (Object Constraint Language)
  - Provides formal basis for pre- and post-conditions

Quick Example



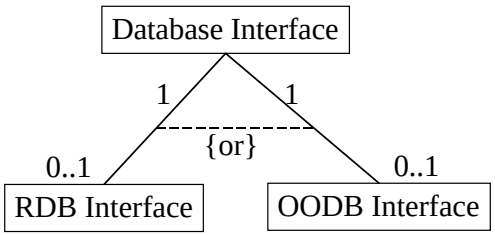
Here we see a database class that hides from its users the actual type of database being used to store the application's persistent information

Example, continued



Unfortunately, the class diagram allows object diagrams like this. Here the database interface is communicating with both a relational database and an object-oriented database. This was unintended—it should be just one at a time!

Example, concluded



In order to specify the intended behavior, a constraint is used. Here the standard UML constraint `or` is used to specify that only one of these associations will hold at a time.

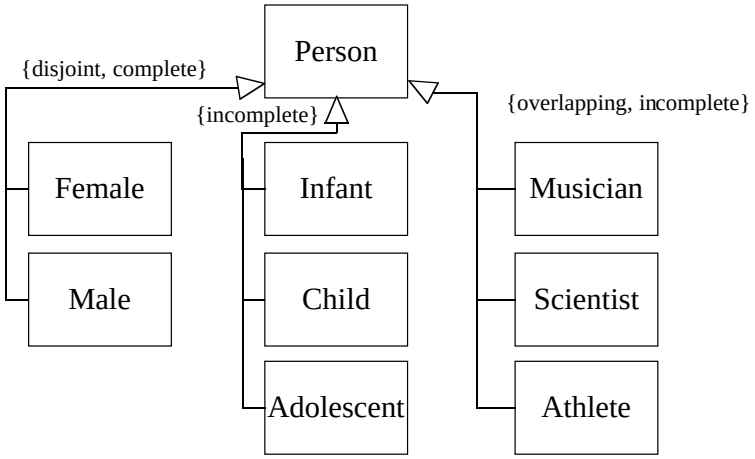
More on Constraints

- Constraints
  - allow you to specify new semantics
  - specify conditions that must be true for a model to be valid
- Notation
  - Constraints are rendered as a string enclosed in brackets
  - Placed near the element(s) they modify

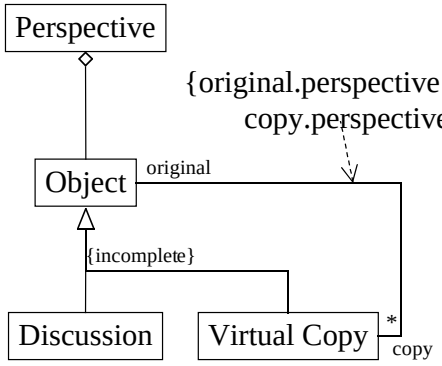
Constraints on Generalization

- We've seen constraints before
  - When creating a set of subtypes
    - {complete}
    - {incomplete}
    - {disjoint}
    - {overlapping}

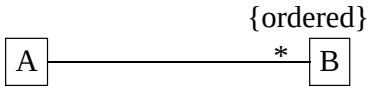
Example (from Lecture 10)



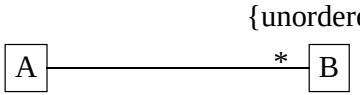
From the Perspectives Project



Constraints on Associations

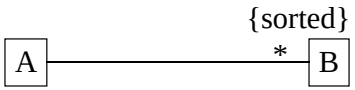


This association specifies that A is connected to zero or more Bs. These Bs are ordered in some fashion and duplicates are prohibited.

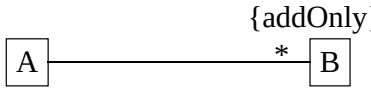


This association specifies that A is connected to zero or more Bs. These Bs are not ordered and duplicates are prohibited. This constraint is the default.

Association Constraints, cont.

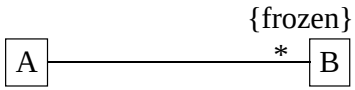


This association specifies that A is connected to zero or more Bs. These Bs are sorted based on their internal values. This is a stronger variation of {ordered}.

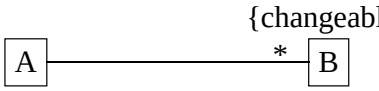


The addOnly constraint specifies that new links may be added, but existing links can not be modified or deleted

Association Constraints, cont.



No links may be added, deleted, or modified after the object has been created and initialized

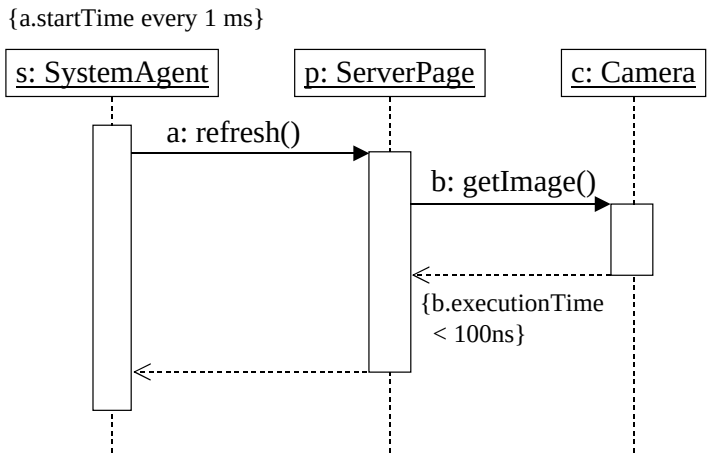


Links may be added, modified and deleted freely.

Changeable, addOnly, Frozen

- These constraints can be applied to
  - association ends and attributes
- With attributes
  - **Changeable**: no restrictions on modification
  - **addOnly**: attributes with a multiplicity greater than one can have additional values added but they can not be changed
  - **Frozen**: the value may not change

Modeling Timing Constraints



Standard Timing Functions

- startTime - when a message starts
- stopTime - when a message ends
- executionTime = stopTime - startTime
- Use these functions to specify constraints that regulate the timing aspects of operations

Format of Constraints

- Standard keywords (see prior examples)
- Natural language text
  - Easy to generate; can still be ambiguous!
- Object Constraint Language
  - A formal language used to express constraints free from side-effects

  <http://www.rational.com/uml/html/ocl/>

  (The OCL is presented in detail next lecture)