Software engineering Cheat Sheet

by [deleted] via cheatography.com/28772/cs/8501/

Entity-relationship model

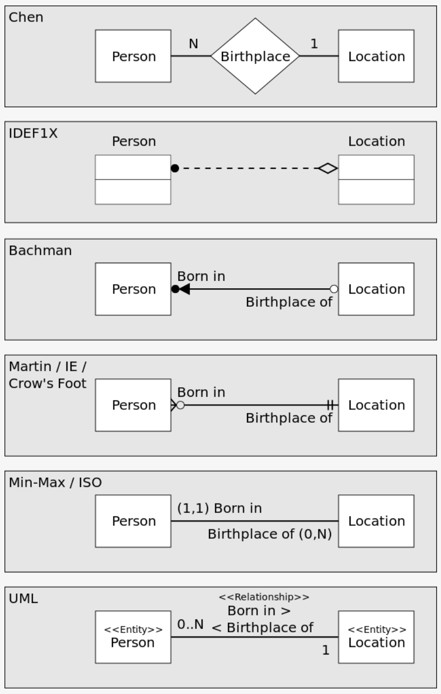
An **entity** is a thing that exists either physically or logically. Entities can be thought of as nouns: *a company*, *a computer*.

A **relationship** captures how entities are related to one another. Relationships can be thought of as verbs, linking two or more nouns.

Entities and relationships can both have **attributes**.

Every entity must have a minimal set of uniquely identifying attributes, which is called the entity's **primary key**.

Relation representation



Various methods of representing the same one to many relationship. In each case, the diagram shows the relationship between a person and a place of birth: each person must have been born at one, and only one, location, but each location may have had zero or more people born at it.

Links

Entity-relationship models

Class diagram

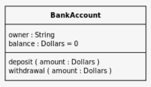
Data modeling

Use case

UML diagrams

class diagram	a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations, and the relationships among objects.
object diagramm	shows a complete or partial view of the system at a given moment of time
domain model	conceptual model of the domain that incorporates both: behaviour and data

Class diagramm



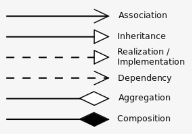
Three compartments of class diagramm:
- Name of the class;
- Attributes of the class;
- Methods of the class;

Class members visibility

+	public
-	private
#	protected
/	derived
~	package

To specify the visibility of a class member (i.e. any attribute or method), these notations must be placed before the member's name

UML relations notation



Instance-level relationships in class diagramms

dependency	connection between dependent and independent model elements; exists when changes to one element may cause changes in dependent element; this relation in uni-directional
association	association is a relationship between two classes when, that allows one instance to perform an action on behalf of another;
aggregation	aggregation is a variant of "has a" relationship; it can occur when a class is a collection of other classes; contained classes are not automatically destroyed when the container is
composition	more specific version of aggregation; when container destroyed every insance if contains will be destroyed as well; composition unlike aggregation is a "whole--part" relationship

Both *aggregation* and *composition* are types of association betweetn classes. The aggregation relationship is often "catalog" containment to distinguish it from composition's "physical" containment.

Class-level relationship

inheritance	indicates that subcless is a cperialized form of superclass; implements "is a" relationship;
realization	relationship between compotent and it's interface;

General relationship

dependency	weaker form of bond that indicates that one class is dependent on the other; one class depends on another when the independent class is a paramter or local variable;

Sponsored by **Readable.com**
Measure your website readability!
https://readable.com