# Class-Object arrow diagrams

## Explanation of Class and Object Diagrams in this documentation:

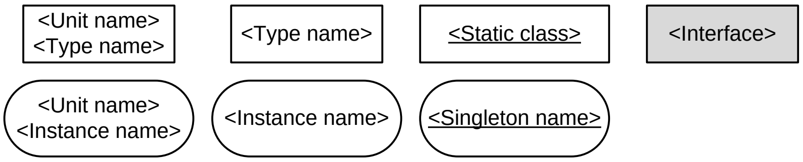
This is a system I have used for many years and is, in my view, more legible than straight UML, and the point of diagrams is communication.

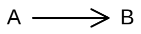
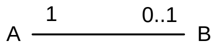
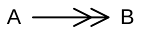
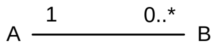
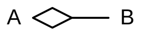
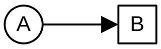
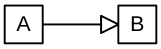
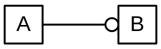
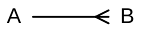
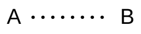
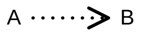
It’s a combination of the old Pascal diagramming that I was taught at university and UML.

Note that I usually mix Class Diagrams and Object Diagrams, because when showing one without the other, one tends to lose valuable context.

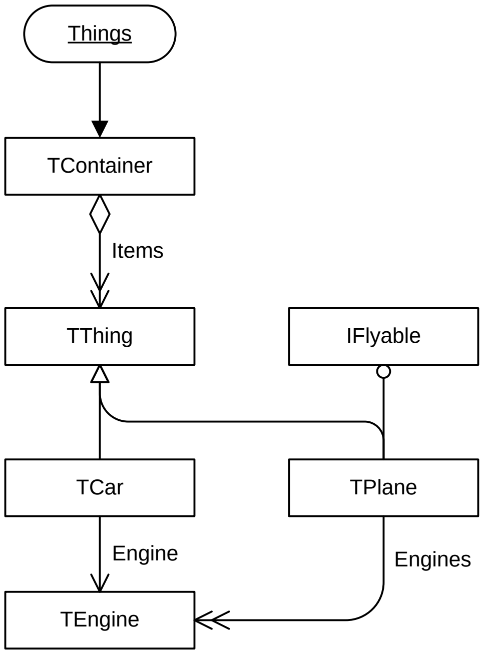
I use other types of UML diagrams without modification.

## Definitions:



| | | |
|---|---|---|
|  | A points to B | UML:  |
|  | A points to multiple B | UML:  |
|  | A owns B (lifetime management) | Based on UML Aggregation, but Visio doesn’t seem to have a filled-in diamond. |
|  | Object A is an instance of B | |
|  | Class A inherits from class B | UML Generalisation. |
|  | Class A implements interface B | Based on UML Interface, but used as an arrow. |
|  | One A to many B | Used mainly in Database Diagrams. |
|  | Unspecified/Loose association between A and B. | |
|  | Usually a method call. | |

## Examples:



Global variable Things is an instance of TContainer.
(This probably means that TContainer is a singleton.)

TContainer manages the lifetimes of multiple instances of the TThing class accessible using the property Items[].

TCar and TPlane descend from TThing.
TPlane implements IFlyable.

Each instance of TCar has a single reference to a TEngine object accessible using the property Engine.

Each instance of TPlane has 0-* references to TEngine objects accessible using the property Engines[].