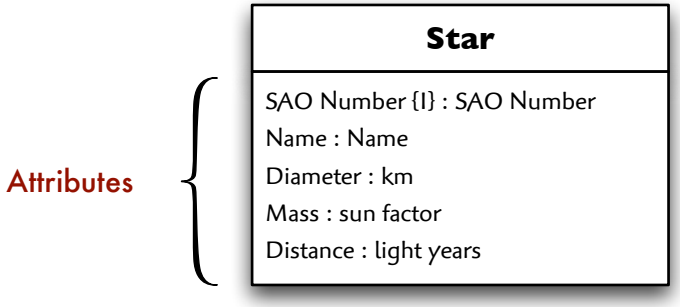
miUML Class and Attribute Descriptions Scrapbook

Leon Starr
Tue May 17 2011

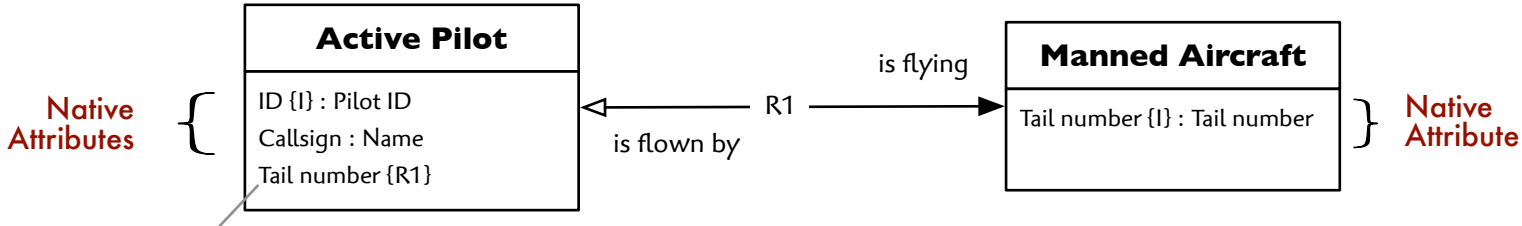
This is a repository for graphics pasted into another document.



Format: `<name> {<tags>} : <type>`

The Type is omitted from the miUML class diagram for Referential Attributes.

Appropriate tags appear for each referential or identification role played by a given Attribute.

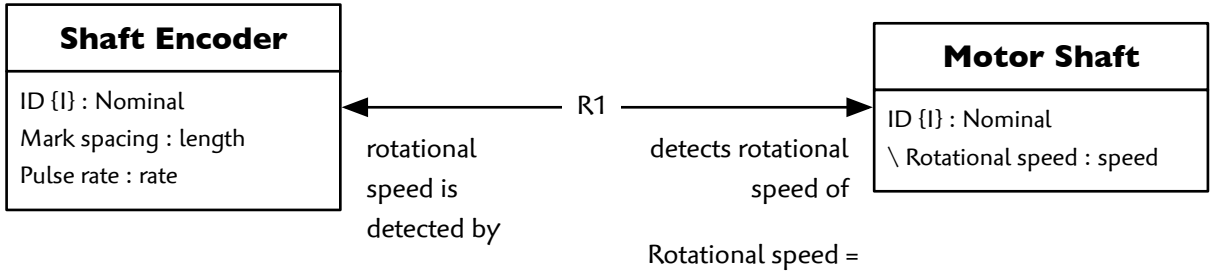


Referential Attribute

The tail number Attribute is defined and Typed'd in the manned aircraft Class, and referenced from the active pilot Class .

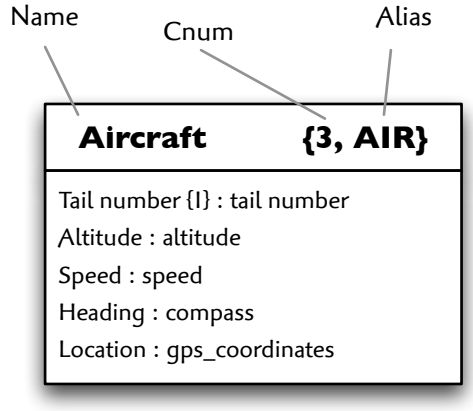
Attributes in Derivation

Shaft Encoder.Mark spacing -> Motor Shaft.Rotational speed
Shaft Encoder.Pulse rate -> Motor Shaft.Rotational speed

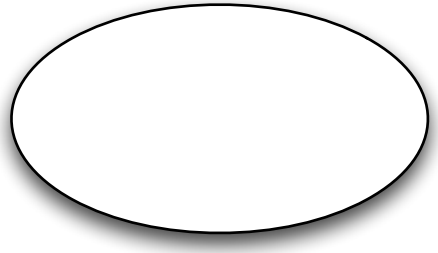


Rotational speed =
self->Shaft Encoder[R1].Mark spacing *
self->Shaft Encoder[R1].Pulse rate

Class



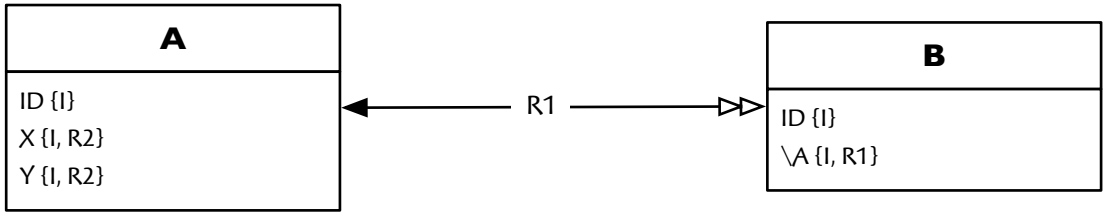
Set of Aircraft



The aircraft class is the definition of a set that may be populated with zero or more elements, each abstracted as an instance of the aircraft class.

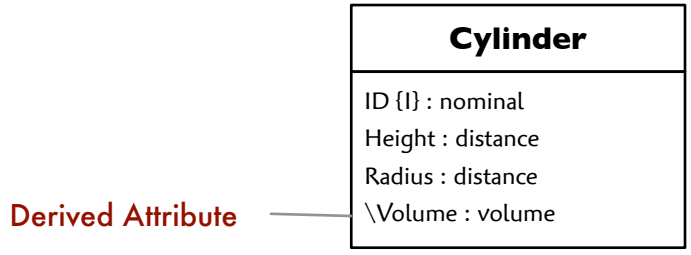
In miUML a Class is just a set definition. It's not a Java class or a C++ class or an SQL table or any particular data structure. It could, however, be implemented as any of these.

Deriving the value of a Referential Attribute

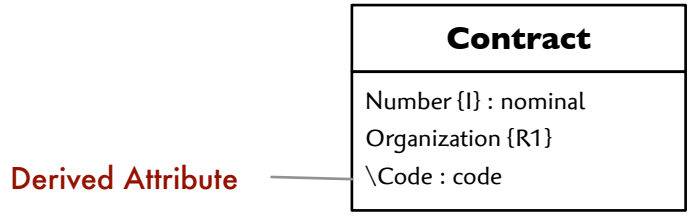


Derivation formula:
B.A = { A.ID, A.X, A.Y }

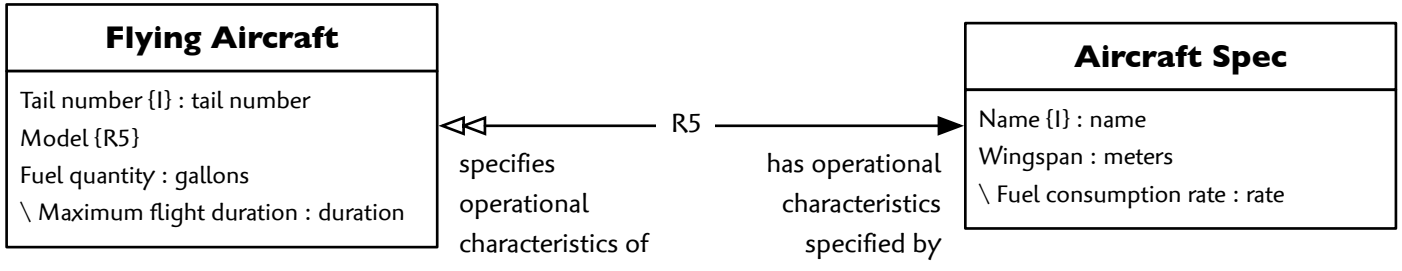
Where { , } yields a set of attribute values



Volume = Height * Radius



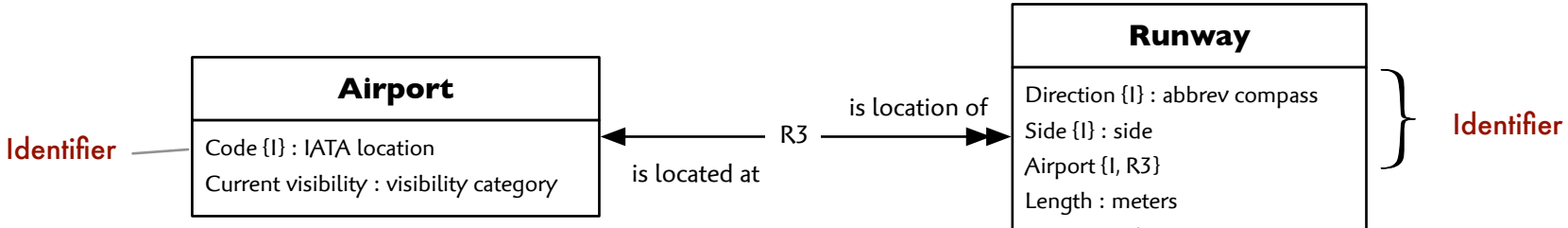
Code = concat(Number, '-', Organization)



Maximum flight duration =

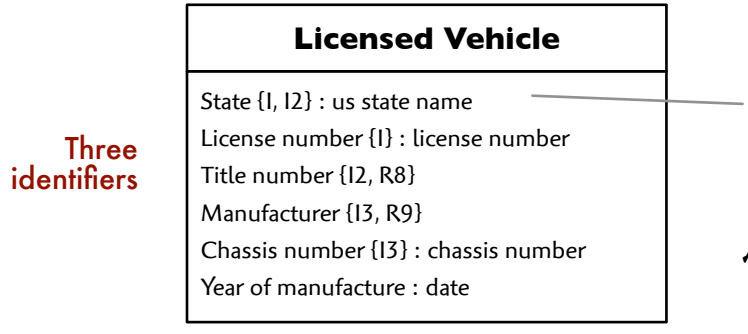
Fuel quantity / self->Aircraft Spec[R5].Fuel consumption rate

Fuel consumption rate = ... (other performance factors)



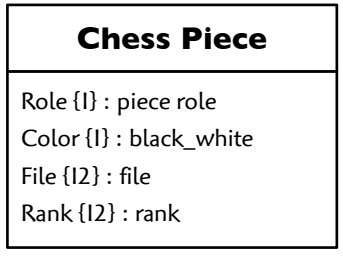
An Identifier may consist of a single Attribute.

An attribute may simultaneously play the role of Referential and Identifier Attribute. An Identifier may consist of multiple Attributes.



The same Attribute may participate in any number of Identifiers.

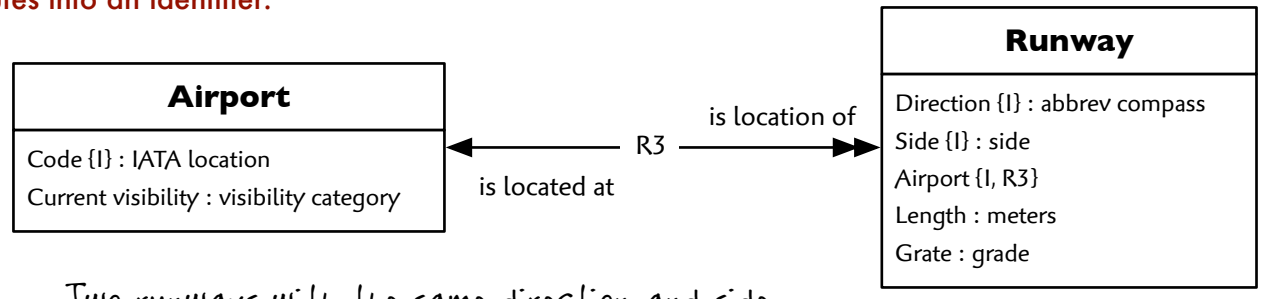
A Class may have multiple Identifiers.



Identity is a constraint

According to the {I2} constraint, no two chess pieces may be present on the same square.

The value of integrating referential attributes into an identifier.



Two runways with the same direction and side (27R) for example may not exist at the same airport. But two separate airports may each have a 27R runway.

The airport code is referenced by the runway.airport Referential Attribute and is a critical component of the runway identifier.

Licensed Vehicle
State {I, I2} : us state name
License number {I} : license number
Title number {I2, R8}
Manufacturer {I3, R9}
Chassis number {I3} : chassis number
Year of manufacture : date

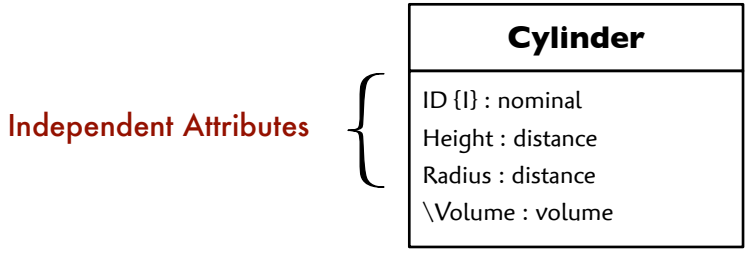
Identifier Attributes

1> State
2> State
1> License number
2 Title number
3> Manufacturer
3> Chassis number

The state Attribute participates in Identifiers 1 and 2 yielding two distinct Identifier Attributes.

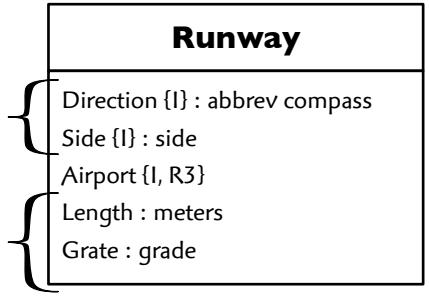
A Class may have multiple Identifiers.

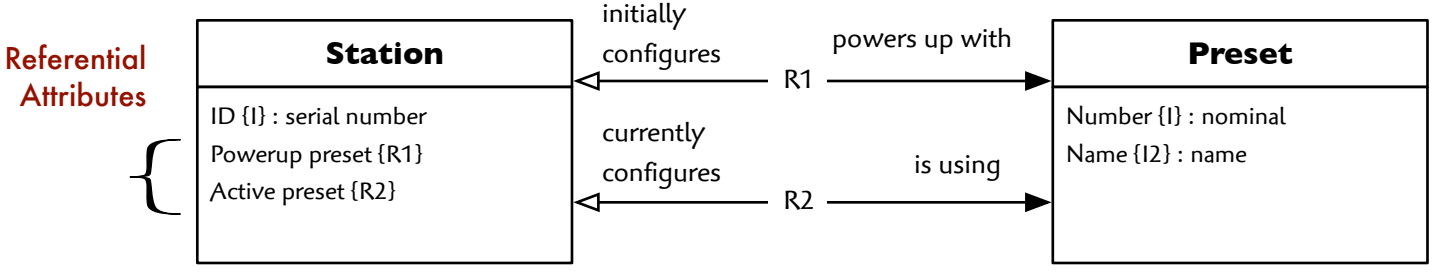
According to the {I2} constraint, no two chess pieces may be present on the same square.



Native Attributes

(non referential)

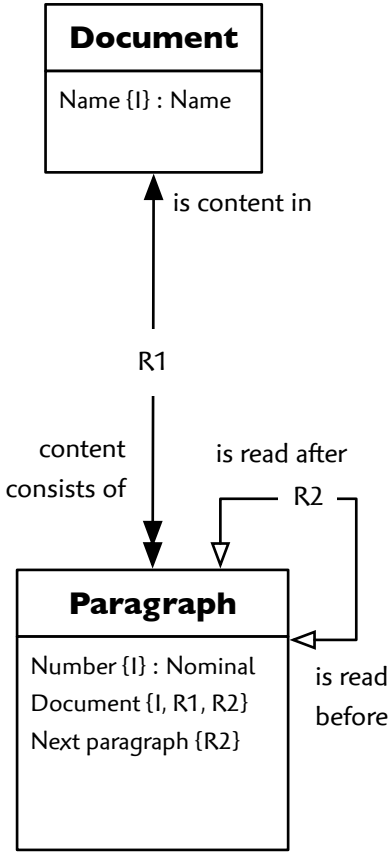




Station.Powerup preset -> Preset.Number

Station.Active preset -> Preset.Number

In this example, the Type of each Referential Attribute is nominal since each refers to Preset.Number.



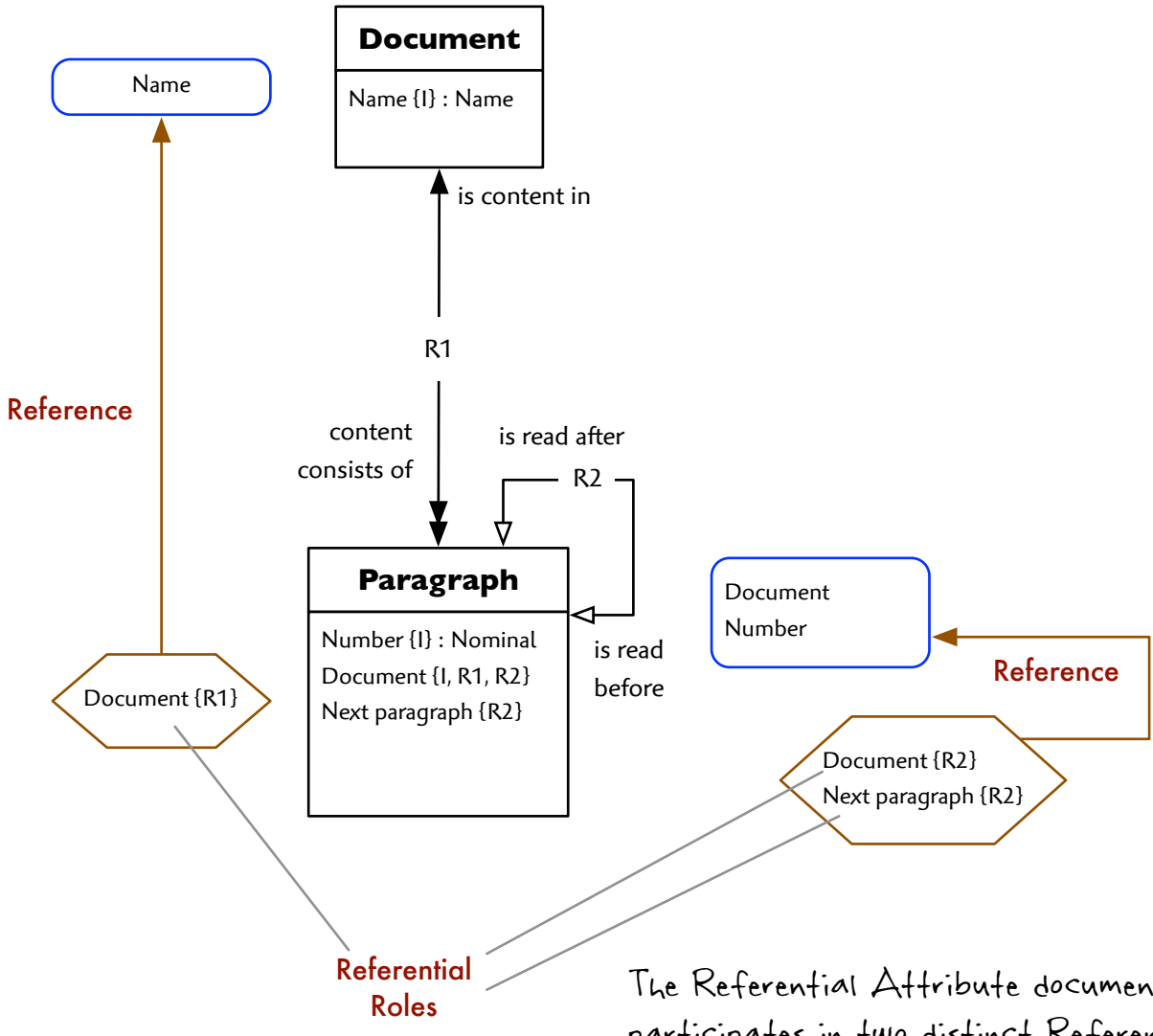
Referential Roles

R1 : Paragraph.Document

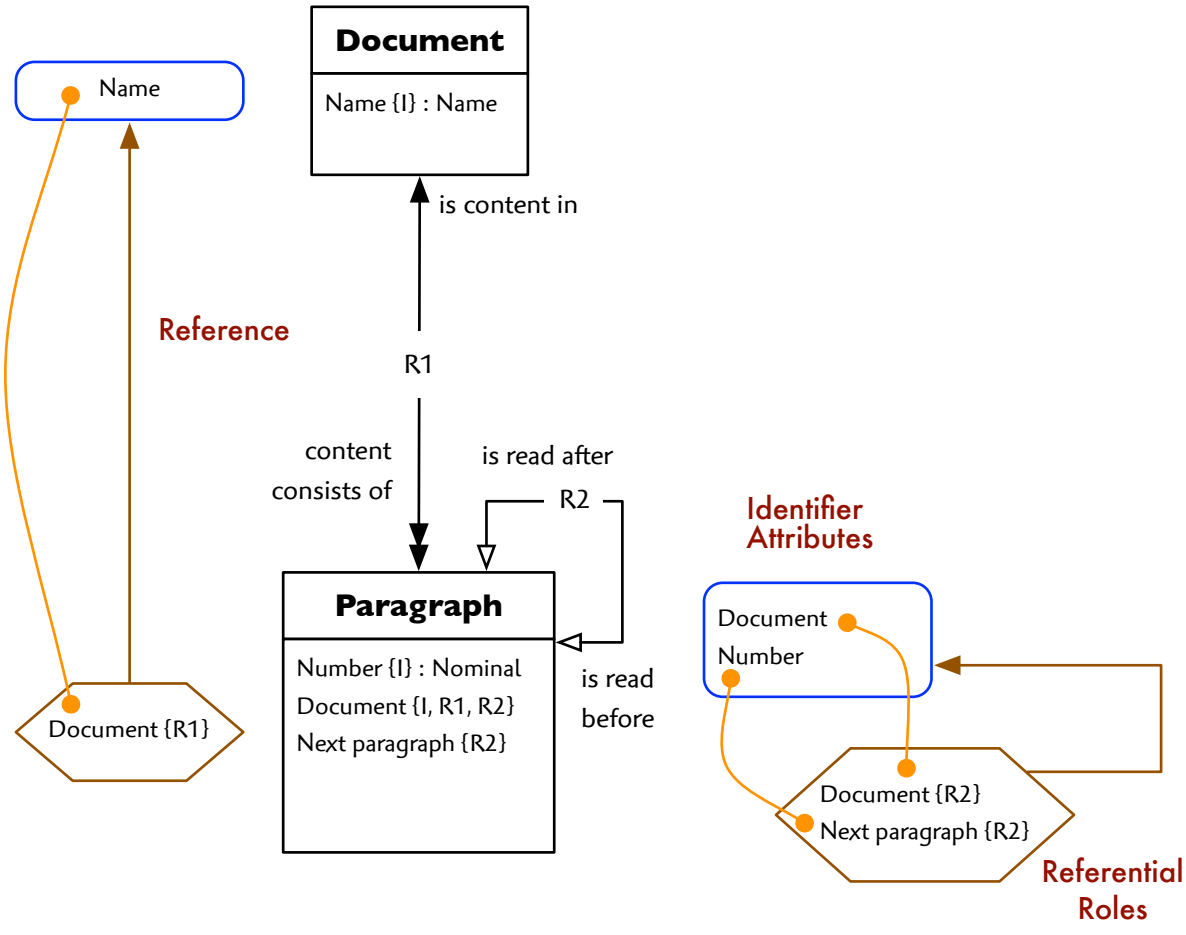
R2 : Paragraph.Document

R2 : Paragraph.Next paragraph

The Referential Attribute paragraph.document participates in two distinct References and, hence, plays two Referential Roles.

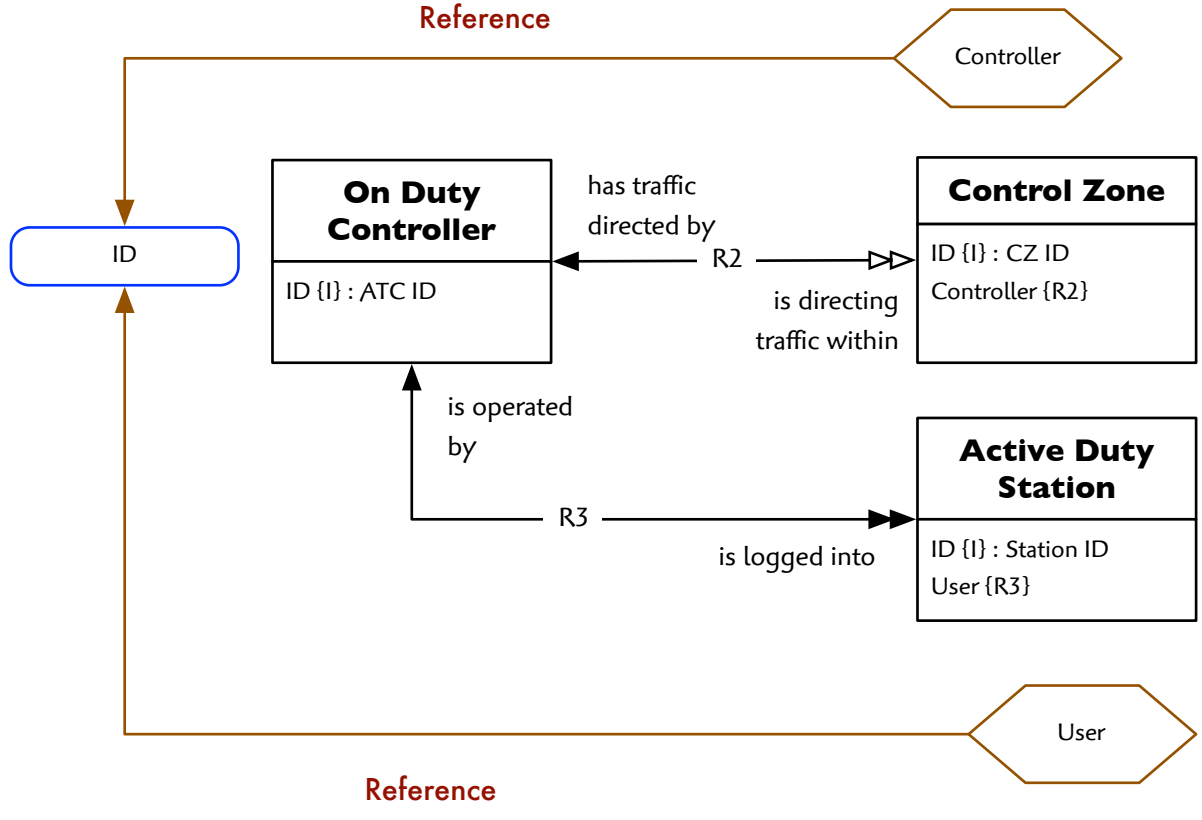


The Referential Attribute document participates in two distinct References and, hence, plays two Referential Roles.

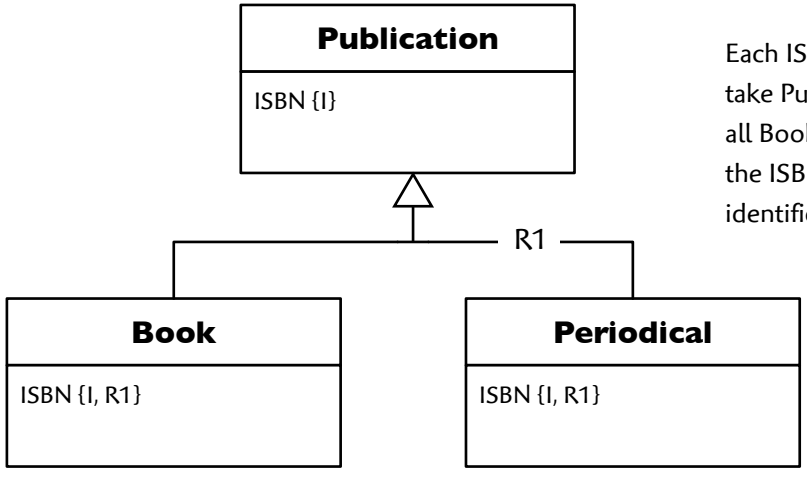


Each Referential Role points to a single Identifier Attribute.

An Identifier may be pointed to by any number of Referential Roles, including none.

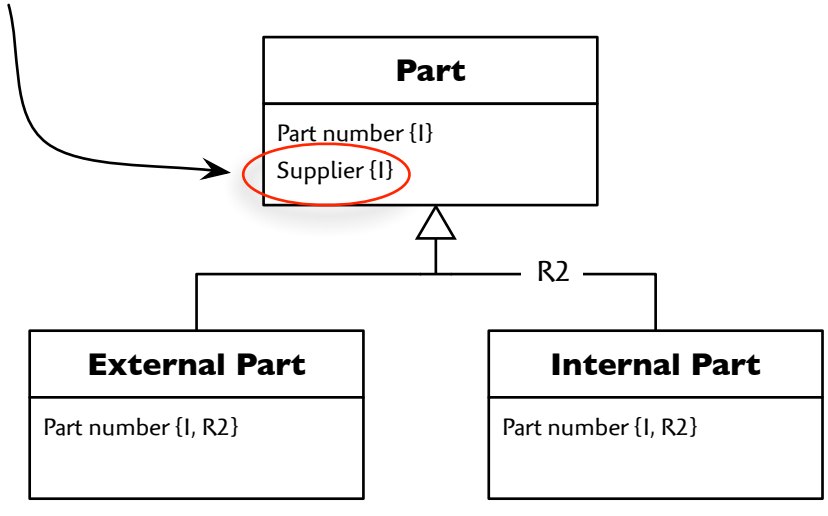


No discrimination attribute is required here:



Each ISBN is unique, so when we take Publication as the union of all Book and Periodical instances, the ISBN serves as the complete identifier.

Discrimination required:



Part numbers are assigned either by an external supplier or internally using independent numbering systems. This means that Part.Part_number is insufficient for identification purposes when the External and Internal Part instance sets are taken as a union to form the Part population.