INTERFACES TYPES AND ROLES

Interface

An interface is a collection of operations that are used to specify a service of a class or a component.***Graphically***, an interface is rendered(represented) as a circle; in its expanded form, an interface may be rendered as a stereotyped class(a class with sterotype interface) as shown in Figure:1. An interface name must be unique within its enclosing package. ***Two naming mechanism***; a simple name(only name of the interface), a path name is the interface name prefixed by the name of the package in which that interface lives represented in Figure: 2. To distinguish an interface from a class, prepend an 'I' to every interface name. Operations in an interface may be adorned with visibility properties, concurrency properties, stereotypes, tagged values, and constraints.interface dont have attributes. interfaces span model boundaries and it does'nt have direct instances.

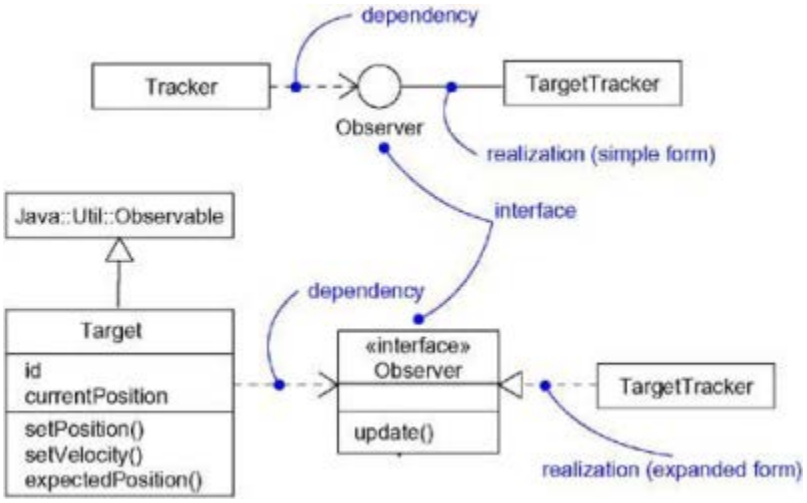

Figure:1

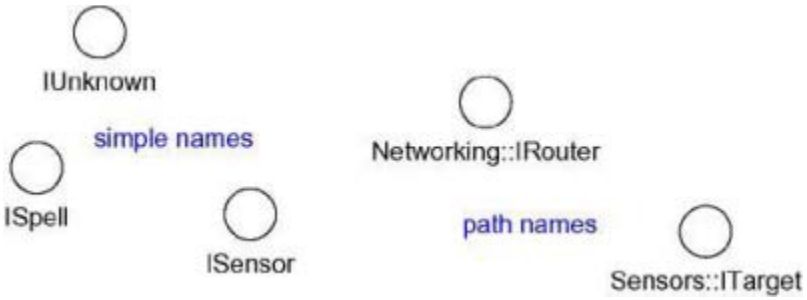

Figure:2

Interface relationships

An interface may participate in generalization, association, dependency and realization relationships.

Note: Interfaces may also be used to specify a contract for a use case or subsystem.

Type

A type is a stereotype of a class used to specify a domain of objects, together with the operations (but not the methods) applicable to the object of that type. To distinguish a type from an interface or a class, prepend a 'T' to every type. Stereotype type is used to formally model the semantics of an abstraction and its conformance to a specific interface.

Role

A role names(indicates) a behavior of an entity participating in a particular context. Or, a role is the face that an abstraction presents to the world. For example, consider an instance of the class Person. Depending on the context, that Person instance may play the role of Mother, Comforter, PayerOfBills, Employee, Customer, Manager, Pilot, Singer, and so on. When an object plays a particular role, it presents a face to the world, and clients that interact with it expect a certain behavior depending on the role that it plays at the time. For example, an instance of Person in the role of Manager would present a different set of properties than if the instance were playing the role of Mother. Figure:3 indicates a role employee played by person and is represented statically there.

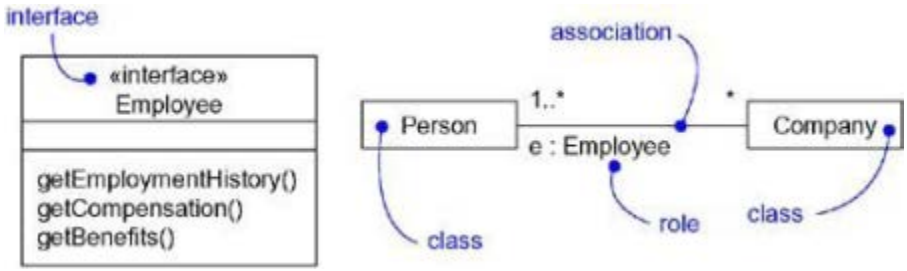

Figure:3 Roles

In Figure:3 the Person presents the role of Employee to the Company, and in that context, only the properties specified by Employee are visible and relevant to the Company.

Static and Dynamic modeling in UML

A class diagram that indicates a particular role is useful for modeling the static binding of an abstraction to its interface. To model the dynamic binding of an abstraction to its interface by using the become stereotype in an interaction diagram, showing an object changing from one role to another.

To model a dynamic type,

- Specify the different possible types of that object by rendering each type as a class stereotyped as type (if the abstraction requires structure and behavior) or as interface (if the abstraction requires only behavior).
- Model all the roles the class of the object may take on at any point in time. It can be done in two ways:
  1. First, in a class diagram, explicitly type each role that the class plays in its association with other classes. Doing this specifies the face instances of that class put on in the context of the associated object.
  2. Second, also in a class diagram, specify the class-to-type relationships using generalization
- In an interaction diagram, properly render each instance of the dynamically typed class. Display the role of the instance in brackets below the object's name.
- To show the change in role of an object, render the object once for each role it plays in the interaction, and connect these objects with a message stereotyped as become.

They are represented in the following figures:

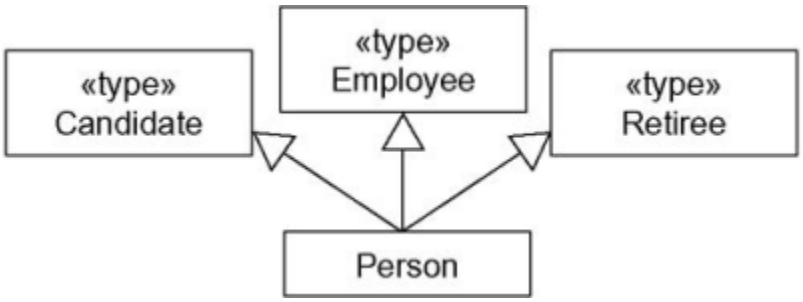

Figure:4 Static modeling

Figure:4 shows statically that instances of the Person class may be any of the three types namely, Candidate, Employee, or Retiree.

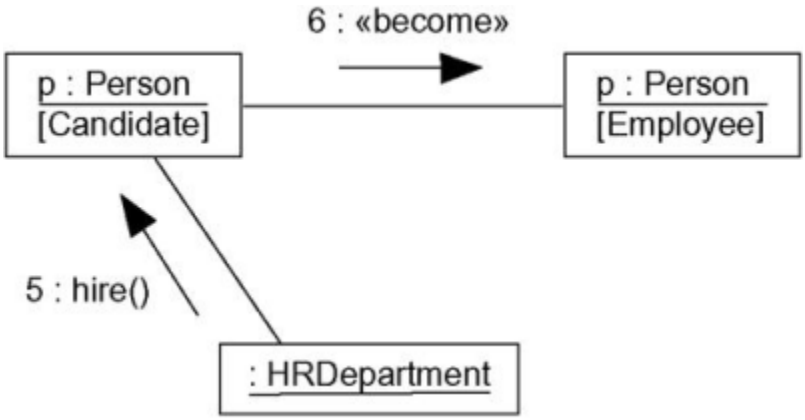

Figure:5 Dynamic-modeling

Figure:5 shows the dynamic nature of a person's type. In this fragment of an interaction diagram, p (the Person object) changes its role from Candidate to Employee.

Source : http://praveenthomasln.wordpress.com/2012/03/03/interfaces-types-and-roles-s8-cs/