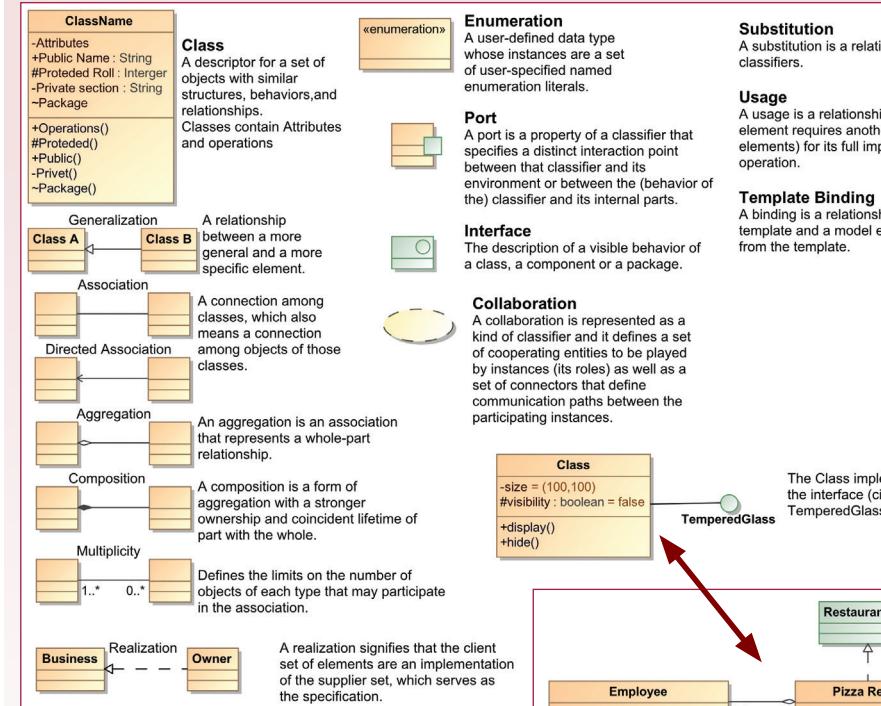


CLASS DIAGRAM:



```

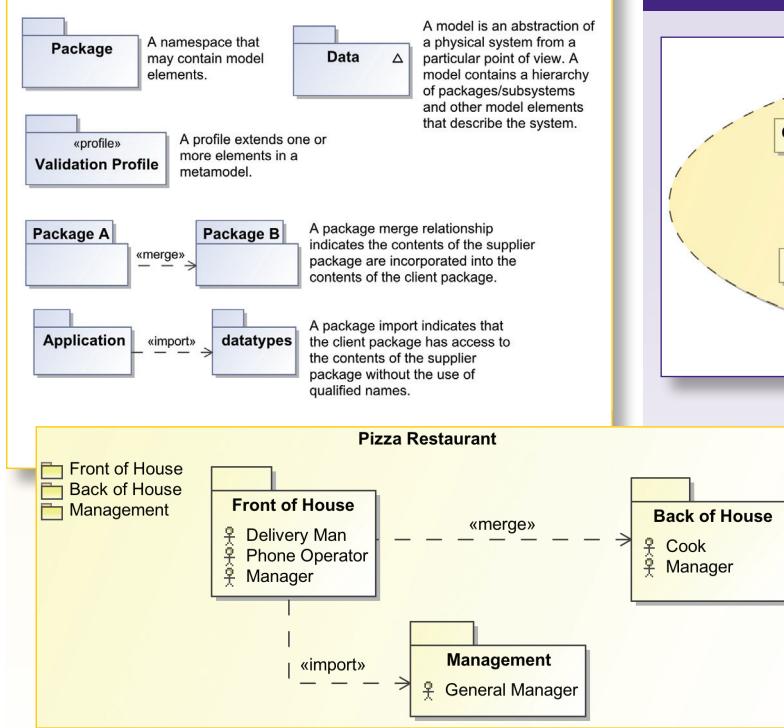
graph LR
    Business[Business] <--> Owner[Owner]

```

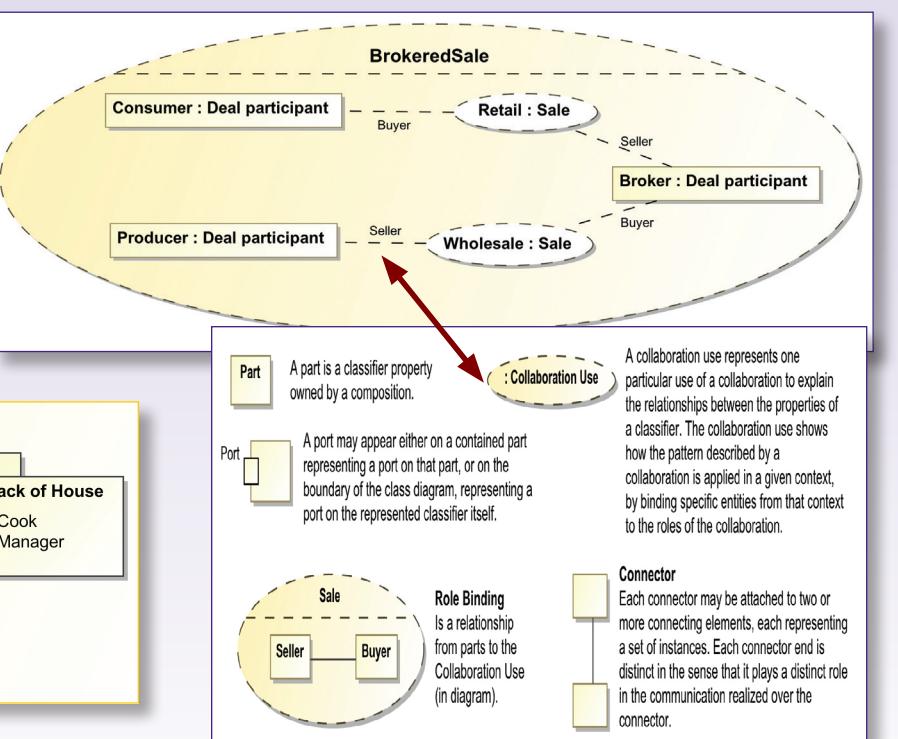
A realization signifies that the client set of elements are an implementation of the supplier set, which serves as the specification.

PACKAGE DIAGRAM:

A package is a namespace that may contain any model elements, including other packages. Packages may participate in the same set of relationships as a class. Package dependencies are derived from relationships between model elements contained in each pair of packages.



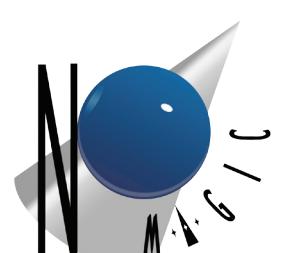
COMPOSITE STRUCTURAL DIAGRAM:



Part	A part is a classifier property owned by a composition.	: Collaboration Use	A collaboration use represents one particular use of a collaboration to explain the relationships between the properties of a classifier. The collaboration use shows how the pattern described by a collaboration is applied in a given context, by binding specific entities from that context to the roles of the collaboration.
Port	A port may appear either on a contained part representing a port on that part, or on the boundary of the class diagram, representing a port on the represented classifier itself.		

The diagram illustrates two concepts in UML Collaboration Diagrams:

- Role Binding:** A relationship between parts of the Collaboration Use (in diagram). It is shown as a line connecting a **Seller** box in the **Sale** lifeline to a **Buyer** box in the same lifeline.
- Connector:** Each connector may be attached to two or more connecting elements, each representing a set of instances. Each connector end is distinct in the sense that it plays a distinct role in the communication realized over the connector.



The Truth is in the Models®
www.nomagic.com

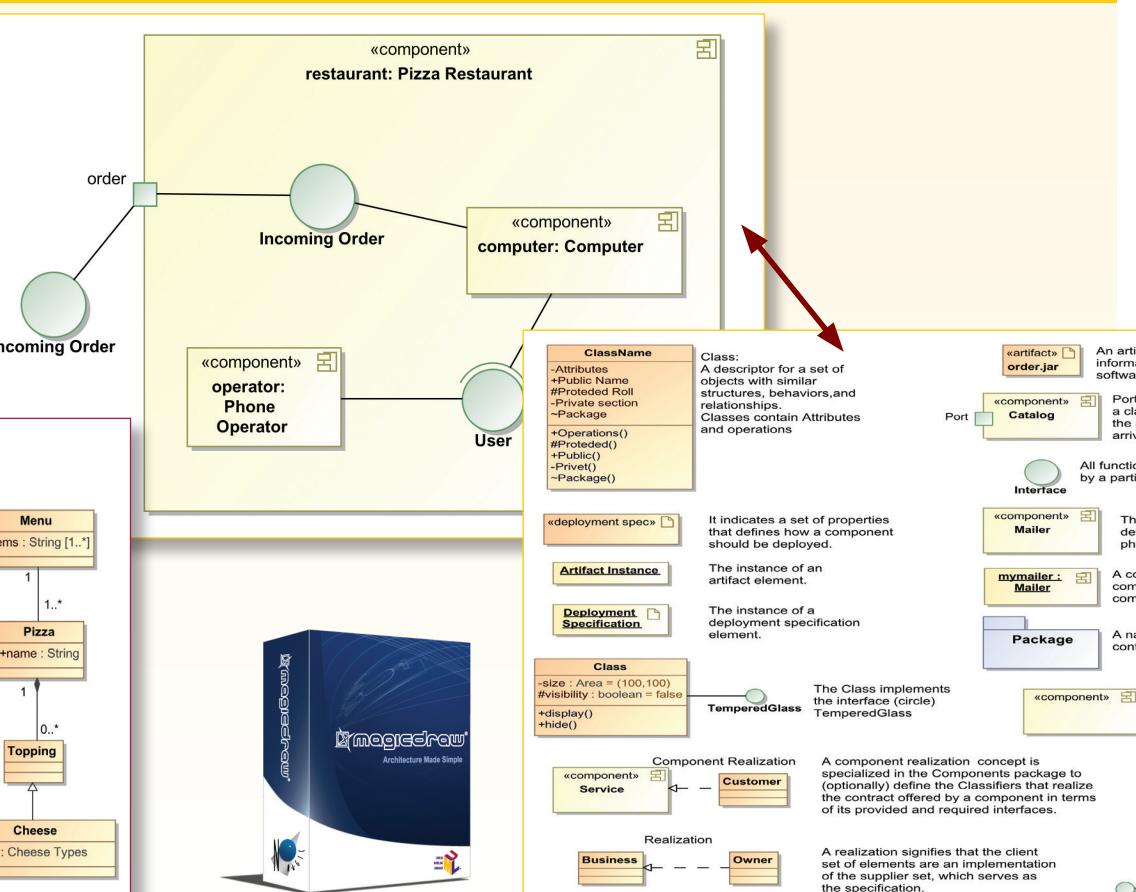


Quick Reference Guide

AT A GLANCE

The Truth is in the Models®

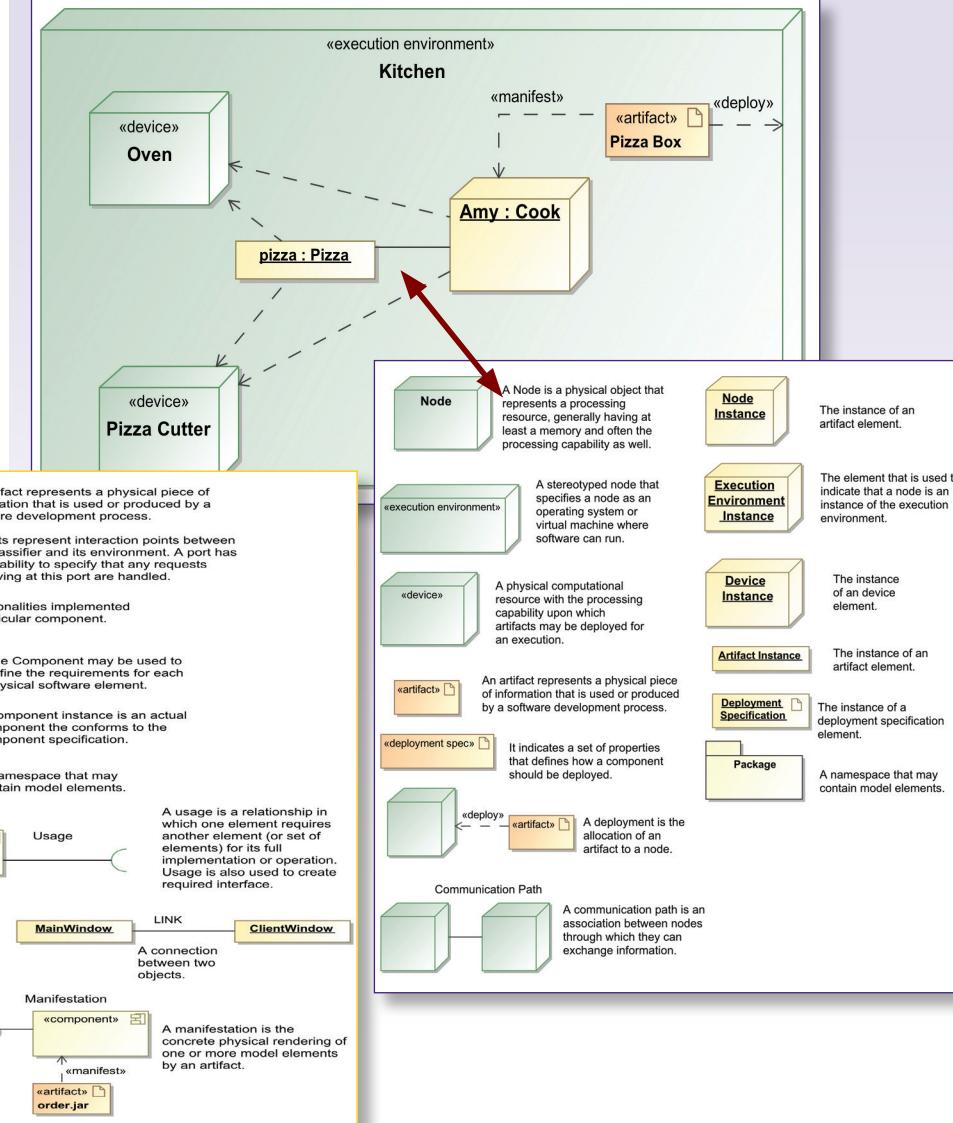
COMPONENT DIAGRAM:



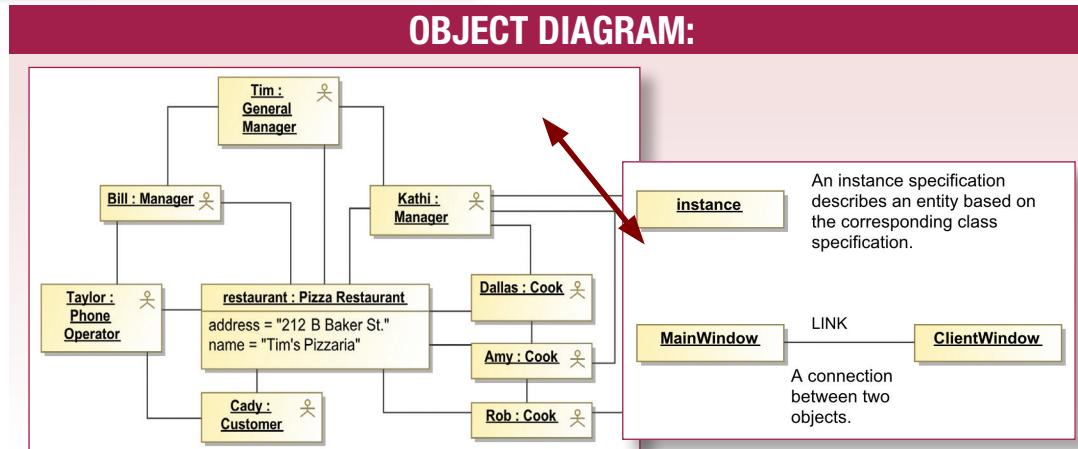
The diagram illustrates various UML elements representing components and their relationships:

- Descriptor**: A class with attributes `name`, `description`, and `version`.
- Port**: A dependency relationship between `Descriptor` and `Component`.
- Component Catalog**: A component containing `Component` instances.
- Component**: A class with attributes `name`, `description`, `version`, `interfaces`, and `operations`.
- Interface**: A class with attributes `name`, `description`, and `operations`. It has a dependency relationship `Realization` to `TemperedGlass`.
- Component Mailer**: A component containing `Component` instances.
- mymailer: Mailer**: A component instance of `Mailer`.
- Package**: A container for `Component` instances.
- TemperedGlass**: A class with attributes `name`, `description`, and `operations`. It has a dependency relationship `Realization` to `Customer`.
- Customer**: A class with attributes `name`, `description`, and `operations`. It has a dependency relationship `Realization` to `Owner`.
- Owner**: A class with attributes `name`, `description`, and `operations`.

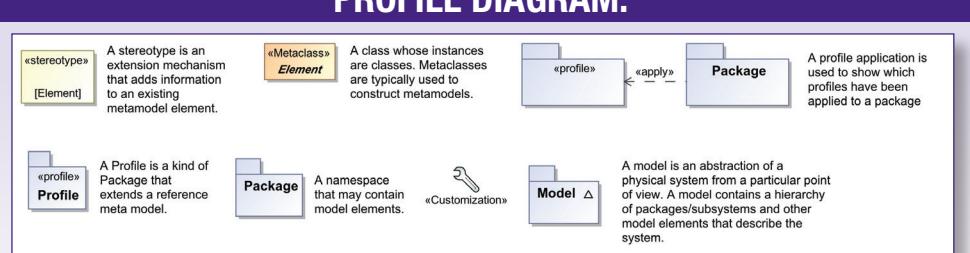
DEPLOYMENT DIAGRAM:



OBJECT DIAGRAM:



PROFILE DIAGRAM:





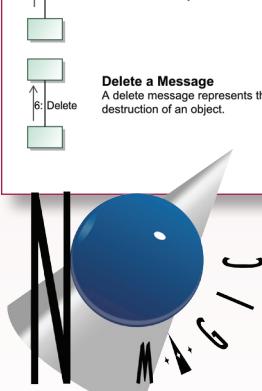
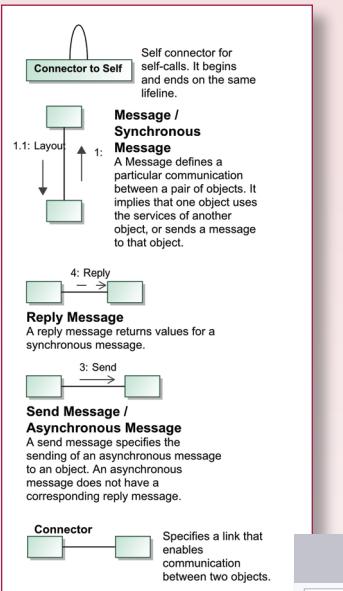
Quick Reference Guide AT A GLANCE

The Truth is in the Models®

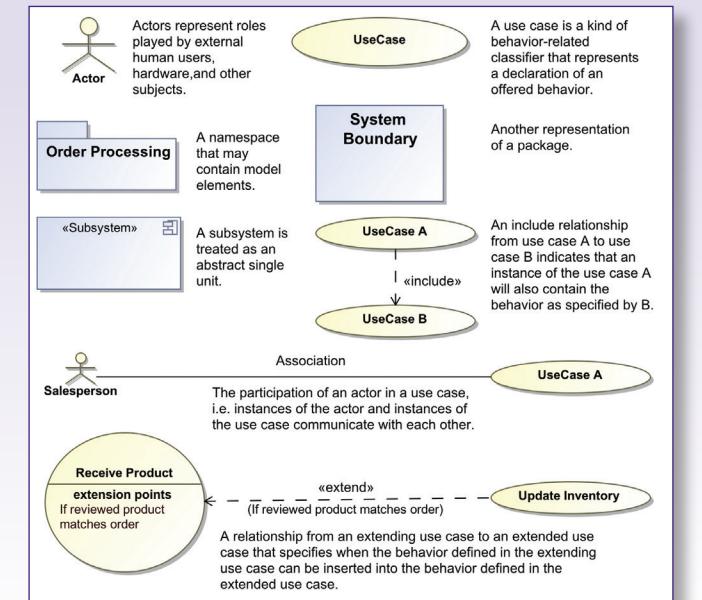
STATE MACHINE DIAGRAM:

PROTOCOL STATE MACHINE DIAGRAM:

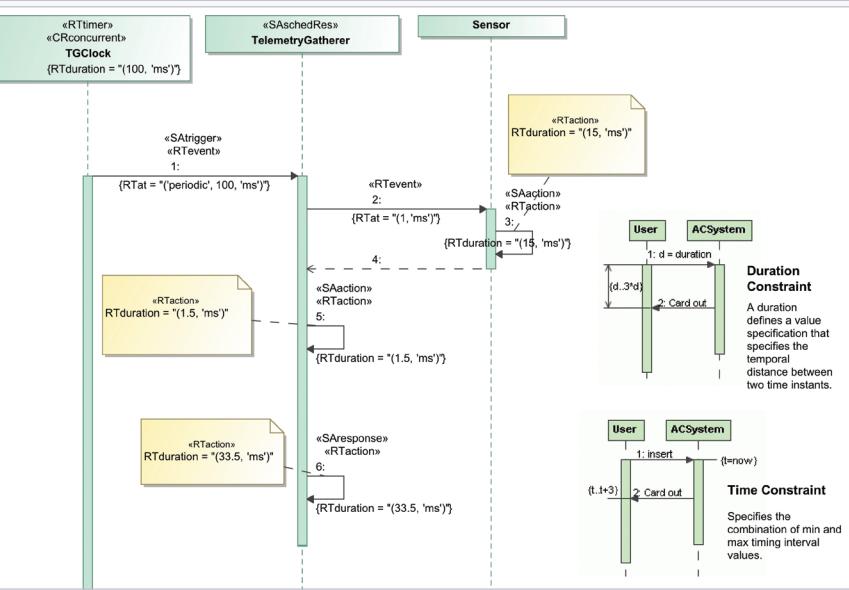
COMMUNICATION DIAGRAM:



USECASE DIAGRAM:



TIMING DIAGRAM:



ACTIVITY DIAGRAM:

