1. Introduction to UML Graphical Notations (Symbols)

Aim: To learn UML graphical notations and understaning basics of UML modeling using Hello Word applet.

Description: Modeling is a proven and well-accepted engineering technique. Modeling is not just a part of the building industry. A model is a simplification of reality.

Unified Modeling Language (UML) is a standard language for writing software blueprints. The Unified Modeling Language (UML) is a graphical language for visualizing, specifying, constructing, and documenting the artifacts of a software-intensive system. The UML is applicable to anyone involved in the production, deployment, and maintenance of software.

In software, there are several ways to approach a model. The two most common ways are from an algorithmic perspective and from an object-oriented perspective. In algorithmic perspective, the main building block of all software is the procedure or function. In object-oriented perspective, the main building block of all software systems is the object or class.

Building Blocks of the UML

Three kinds of building blocks:

1. Things
2. Relationships
3. Diagrams

Things are the abstractions that are first-class citizens in a model; relationships tie these things together; diagrams group interesting collections of things.

Things in the UML

There are four kinds of things in the UML:

1. Structural things
2. Behavioral things
3. Grouping things
4. Annotational things

Structural things are the nouns of UML models. These are the mostly static parts of a model, representing elements that are either conceptual or physical. There are seven structural things.

1. Class

A class is a description of a set of objects that share the same attributes, operations, relationships, and semantics. A class implements one or more interfaces. **Graphically, a class is rendered** as a rectangle, usually including its name, attributes, and operations.

2. Interface

An interface is a collection of operations that specify a service of a class or component. An interface therefore describes the externally visible behavior of that element. Graphically, an interface is rendered as a circle together with its name.

3. Collaboration

A collaboration defines an interaction and is a society of roles and other elements that work together to provide some cooperative behavior that's bigger than the sum of all the elements. Graphically, a collaboration is rendered as an ellipse with dashed lines.

4. Use Case

A use case is a description of set of sequence of actions that a system performs that yields an observable result of value to a particular actor. A use case is used to structure the behavioral things in a model. A use case is realized by a collaboration. Graphically, a use case is rendered as an ellipse with solid lines.

5. Active Class

An active class is a class whose objects own one or more processes or threads and therefore can initiate control activity. Graphically, an active class is rendered just like a class, but with heavy lines

6. Component

A component is a physical and replaceable part of a system that conforms to and provides the realization of a set of interfaces. Graphically, a component is rendered as a rectangle with tabs.

7. Node

A node is a physical element that exists at run time and represents a computational resource, generally having at least some memory and, often, processing capability. Graphically, a node is rendered as a cube.

Behavioral things are the dynamic parts of UML models. These are the verbs of a model, representing behavior over time and space. There are two behavioral things. First, an interaction is a behavior that comprises a set of messages exchanged among a set of objects within a particular context to accomplish a specific purpose. Graphically, a message is rendered as a directed line. Second, a state machine is a behavior that specifies the sequences of states an object. Graphically, a state is rendered as a rounded rectangle.

Grouping things are the organizational parts of UML models. These are the boxes into which a model can be decomposed. There is one grouping thing, namely, packages. A package is a general-purpose mechanism for organizing elements into groups. Graphically, a package is rendered as a tabbed folder.

Annotational things are the explanatory parts of UML models. These are the comments you may apply to describe, illuminate, and remark about any element in a model. There is one annotational thing, called a note. There is one primary kind of annotational thing, called a note. A note is simply a symbol for rendering constraints and comments attached to an element or a collection of elements. Graphically, a note is rendered as a rectangle with a dog-eared corner

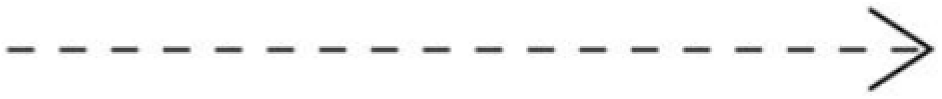
Relationships in the UML

There are four kinds of relationships in the UML:

1. Dependency

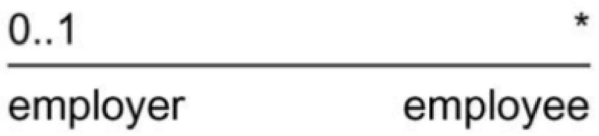
First, a dependency is a semantic relationship between two things in which a change to one thing (the independent thing) may affect the semantics of the other thing (the dependent thing). Graphically, a dependency is rendered as a dashed line, possibly directed, and occasionally including a label.

Figure 2-12 Dependencies



2. Association

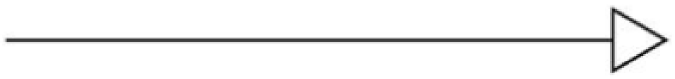
An association is a structural relationship that describes a set of links, a link being a connection among objects. Graphically, an association is rendered as a solid line.



3. Generalization

A generalization is a specialization/generalization relationship in which objects of the specialized element (the child) are substitutable for objects of the generalized element (the parent). Graphically, a generalization relationship is rendered as a solid line with a hollow arrowhead pointing to the parent

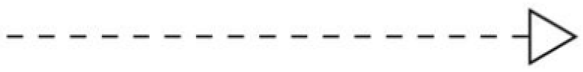
Figure 2-14 Generalizations



4. Realization

A realization is a semantic relationship between classifiers, wherein one classifier specifies a contract that another classifier guarantees to carry out. Graphically, a realization relationship is rendered as a cross between a generalization and a dependency relationship

Figure 2-15 Realization



Diagrams in the UML

A diagram is the graphical presentation of a set of elements, most often rendered as a connected graph of vertices (things) and arcs (relationships). We draw diagrams to visualize a system from different perspectives, so a diagram is a projection into a system. The UML includes nine such diagrams:

1. Class diagram
2. Object diagram
3. Use case diagram
4. Sequence diagram
5. Collaboration diagram
6. Statechart diagram
7. Activity diagram
8. Component diagram
9. Deployment diagram

In this SE lab, we learn modeling software systems to address the given projects or problem statements. Commonly used UML models are Class diagram, Use case, Sequence, Activity, and Component diagram.

Hello Word Applet:

Here we undersatnd use of UML Graphical symbols using Java code of Hello world Applet.

In Java, the applet for printing "Hello, World!" in a Web browser is quite simple:

```
import java.awt.Graphics;
class HelloWorld extends java.applet.Applet {
public void paint (Graphics g) {
g.drawString("Hello, World!", 10, 10);
}
}
```

Conclusion:

We have learnt UML graphical symbols and the application of those symbols using Hello World Applet Java Code. So prior to coding, attempting UML diagrams is improtant for understanding the given problem.

Viva Questions:

1. What is UML?
2. What is a model?

3. State Building blocks of UML?

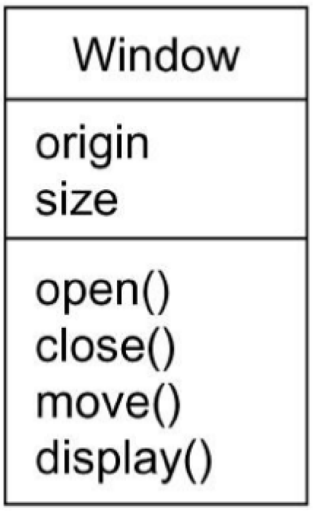
4. How many structural things are there? List them.

UML User Guide:

https://www.researchgate.net/publication/234785986_Unified_Modeling_Language_User_Guide_The_2nd_Edition_Addison-Wesley_Object_Technology_Series

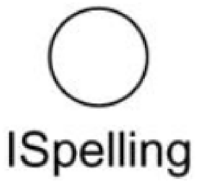
Graphical Notations:

1. Class

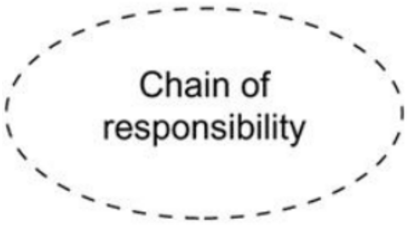


2. Interface

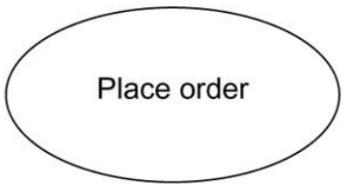
Figure 2-2 Interfaces



3. Collaboration

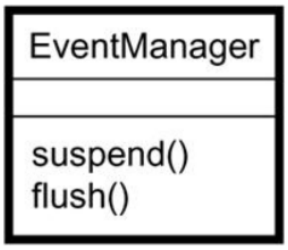


4. Use case



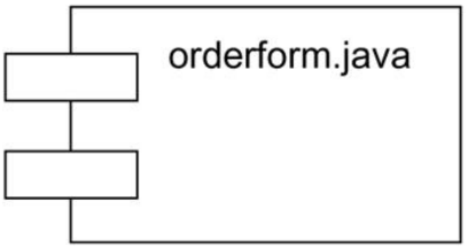
5. Active class

Figure 2-5 Active Classes

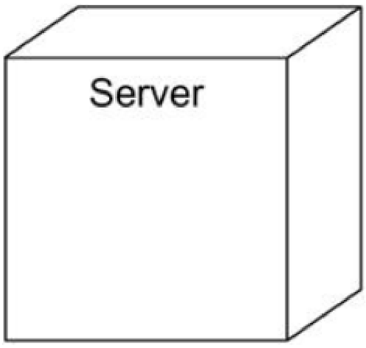


6. Component

Figure 2-6 Components



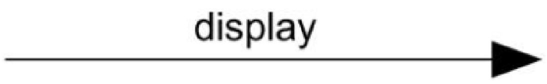
7. Node



Behavioral things:

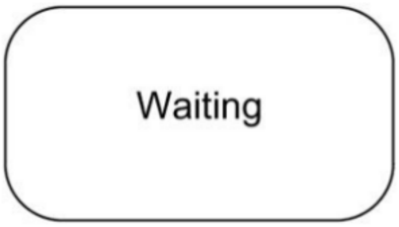
1. Message

Figure 2-8 Messages



2. State machine

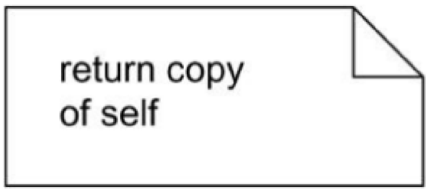
Figure 2-9 States



Annotational Things

1.Note

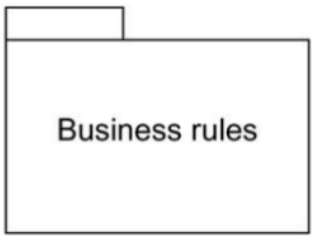
Figure 2-11 Notes



Grouping Things

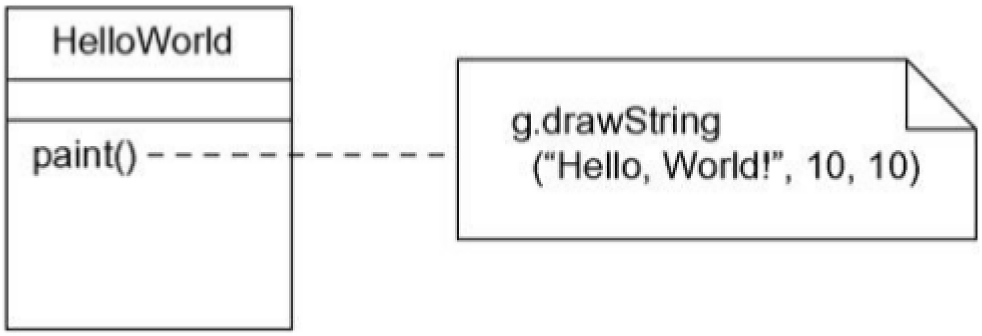
1. Package

Figure 2-10 Packages



Modeling diagrams of Hello Word Applet:

Figure 3-1 Key Abstractions for `HelloWorld`



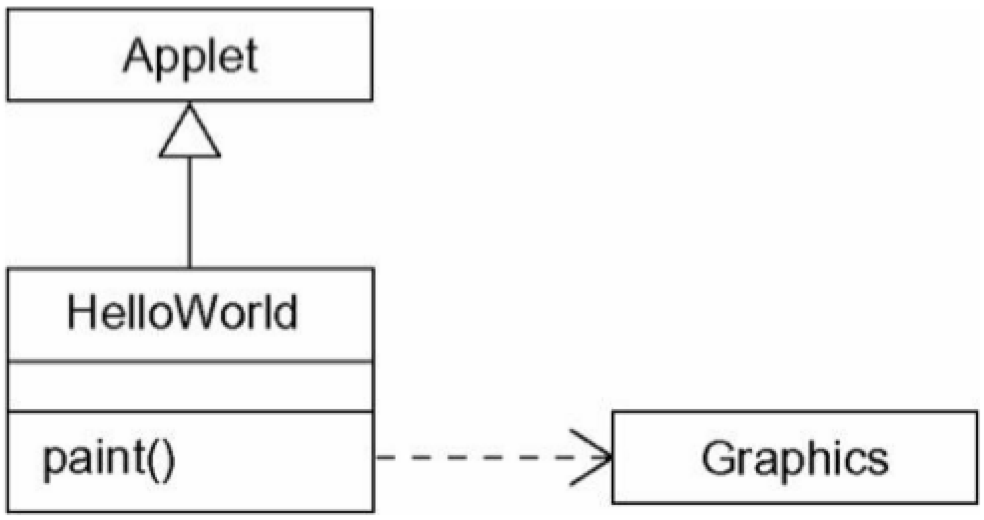


Figure 3-3 `HelloWorld` Inheritance Hierarchy

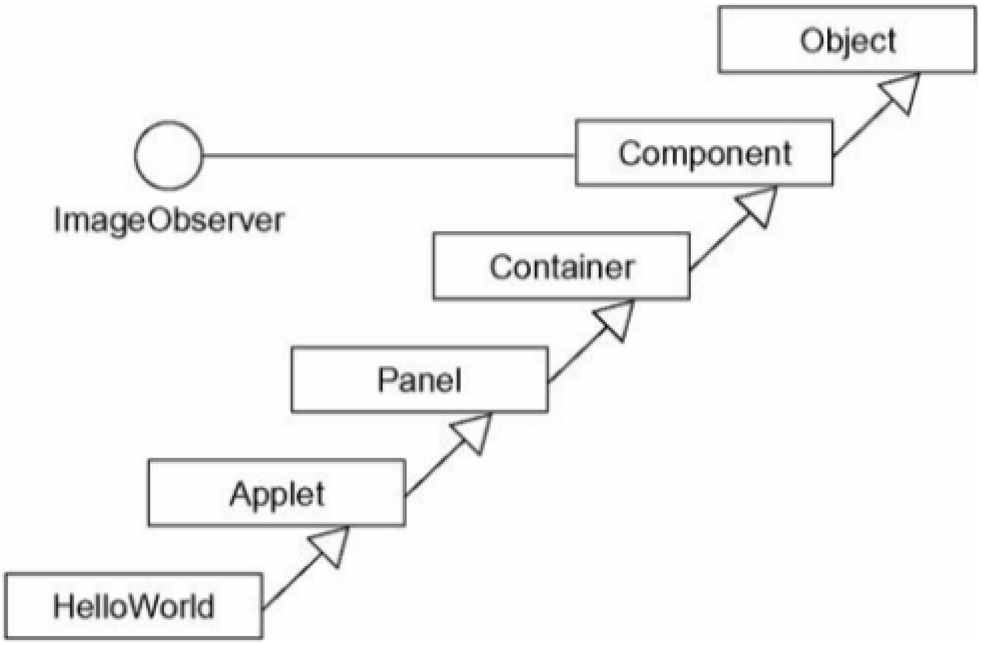


Figure 3-4 `HelloWorld` Packaging

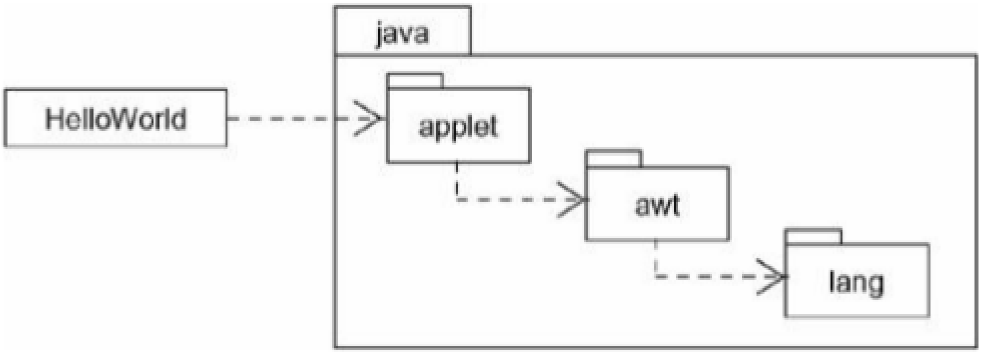


Figure 3-5 Painting Mechanism

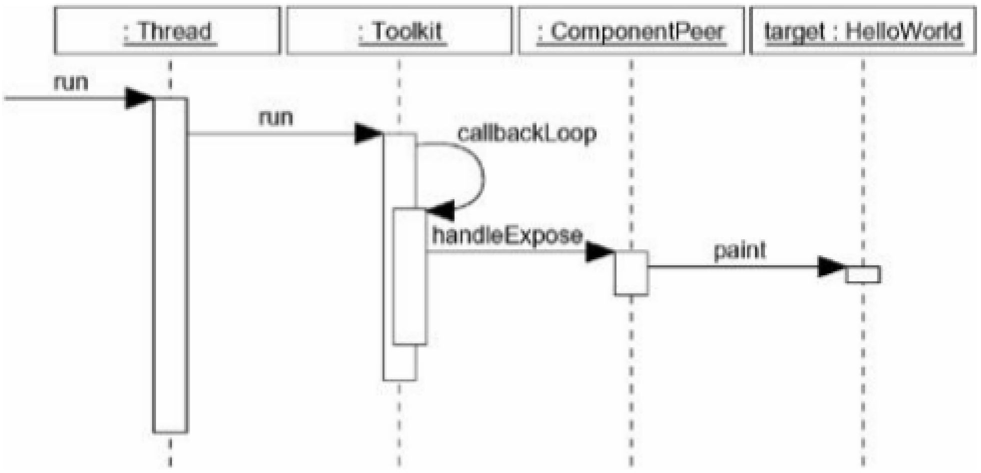
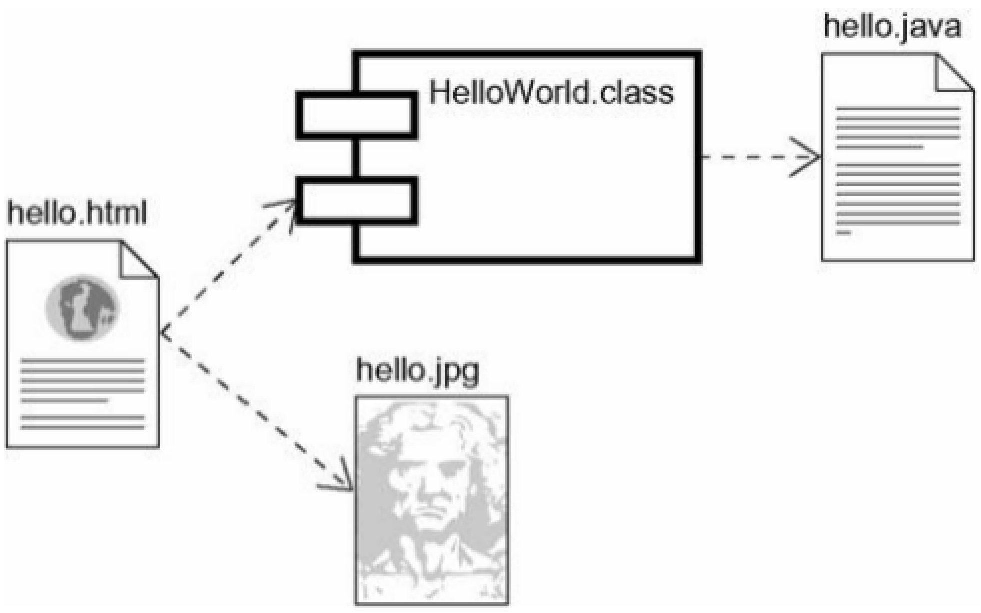


Figure 3-6 `HelloWorld` Components


hello.java
HelloWorld.class
hello.html
hello.jpg