Dr.. Hanash disinfectant

Modeling language

consolidated

(UML)

- **language): (UML**

It is an abbreviation for (Unified Modeling Language). In Arabic,

it is called (Unified Modeling Language).

- **use language): (UML**

In the work of planning programming projects before starting the work of the software project.

- **language components): (UML**

1. Geometric shapes.
2. Symbols such as (square, rectangle, straight line, arrows, hexagon).

Note: Each of the symbols symbolizes something in the software project

- **salient features of language: (UML**

1. Unified Modeling Language is not associated with software production methodology or methods.

2. The Unified Modeling Language provides a set of engineering best practices expertise.

3. The Unified Modeling Language is not a methodology for building or designing and developing software.

UMLIt consists of four basic layers

The Unified Modeling Language

1. User Objects Layer

It is the general surface layer used by those working with the UML, and it consists of nine main diagrams plus objects and utilities

It is the most clear and descriptive layer. The user here is the user of the language and not the end user of the software or software product.

The first layer includes the following nine schemes:

* Outline of cases to use.

* Category scheme.

* Activity chart.

* Status chart.

Component diagram.

* Object outline.

* Collaboration scheme.

* Sequence diagram.

Distribution chart.

2. The second layer: called the Model Layer

And it is in the first stage of the analysis, as it contains the concepts of the subject of the analysis, such as understanding the system, and this layer is used by the system analyst during his work before the system idea matures or becomes clear with a description less than the first user layer.

3.The third layer: the meta model layer

This layer is concerned with concepts related to the Unified Modeling Language, such as the concept of class, phenomenon, data type, abstraction, patterns, and other language concepts. It is a layer that describes what is happening in the model and consists of three main packages:

* Basic package.

* Behavioral Elements Pack.

* Form management package

4. Layer 4: Meta meta model layer

Layer is not of interest to most systems analyst

Interested in the Unified Modeling Writing Language, and of interest to developers of Unified Modeling Language tools

Like programs that automate charts and draw them.

Schematics of the modeling language**UML DIAGRAMS**

consolidated

1. Use Case Diagram:

Shows the relationship between actors and use cases.

2. Class Diagram:

Class structure models and their contents use design elements such as classes, packages, and objects.

It also displays relationships such as containment, inheritance, associations, and others.

3. Interaction Diagrams:

- **sequence diagram (Sequence Diagram**

Displays the timeline of the objects participating in the interaction. This consists of the vertical dimension (time) and the horizon dimension (different objects).

- **collaboration chart (Collaboration Diagram**

Shows structured interaction around objects and their relationships with each other. Numbers are used to show the sequence of messages.

4 . State Diagram:

It presents the sequence of situations through which the object of an interaction has gone through in its life in the process of responding to previously received stimuli, along with its responses and actions.

5. Activity Diagram:

View the characteristic diagrams of the state diagram, where most of the states are action states and most of the transitions are triggered by the termination of actions in the states source. This scheme focuses on flows driven from internal processing.

6. Physical Diagrams:

- **composite diagram (Component Diagram**

Displays the high level structure of the code package itself. Depending on the components that emerged, including source code components, binary code components, and executable components.

- **deployment planning (Diagram Deployment**

Displays the configuration of run-time processing and software components, processes, and objects that run in them. Software component instances provide run-time aspects of codeunits.