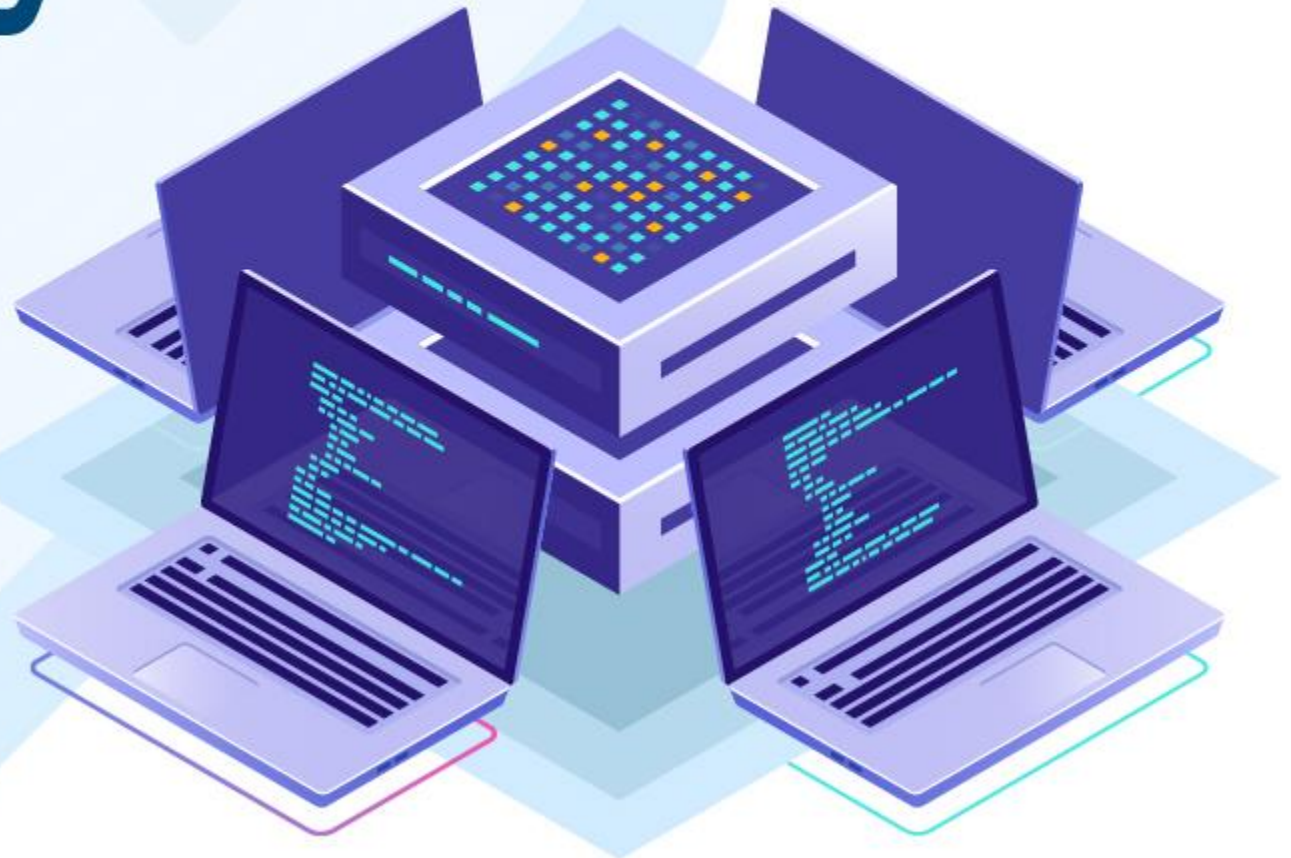


Information Technology

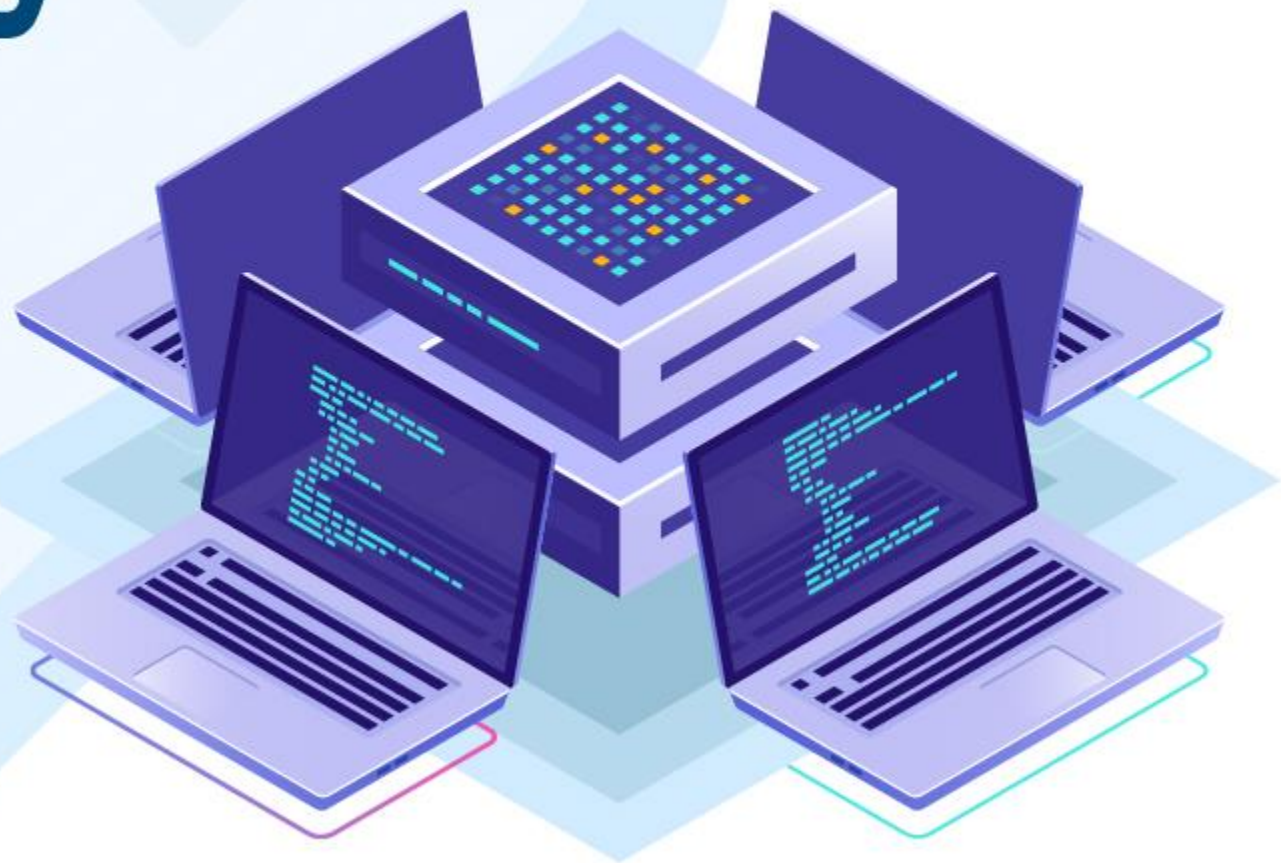
Spring 23-24
Semester 4

- » Web Technology
- » Dr. Rodaina Abdelsalam
rabdelsalam@eelu.edu.eg



Information Technology

- » Introduction to
- » internet concepts



EELU

الجامعة المصرية للتعليم الإلكتروني
THE EGYPTIAN E-LEARNING UNIVERSITY

Outline

- Modeling Web Applications
 1. Content Modeling
 2. Hypertext Modeling
 3. Presentation Modeling

Hypertext Modeling: description

- The hypertext is one of the most important properties to be taken into account when modeling Web applications.
- Thus, the hypertext structure has to be designed carefully.
- This can be achieved by using suitable access structures, that is, navigation options, to avoid the risk of users getting lost and putting them under excessive cognitive stress.
- The objective of hypertext modeling – also known as navigation modeling – is to specify the navigability through the content of a Web application, i.e., the navigation paths available to the users.

Hypertext Structure Modeling Concepts

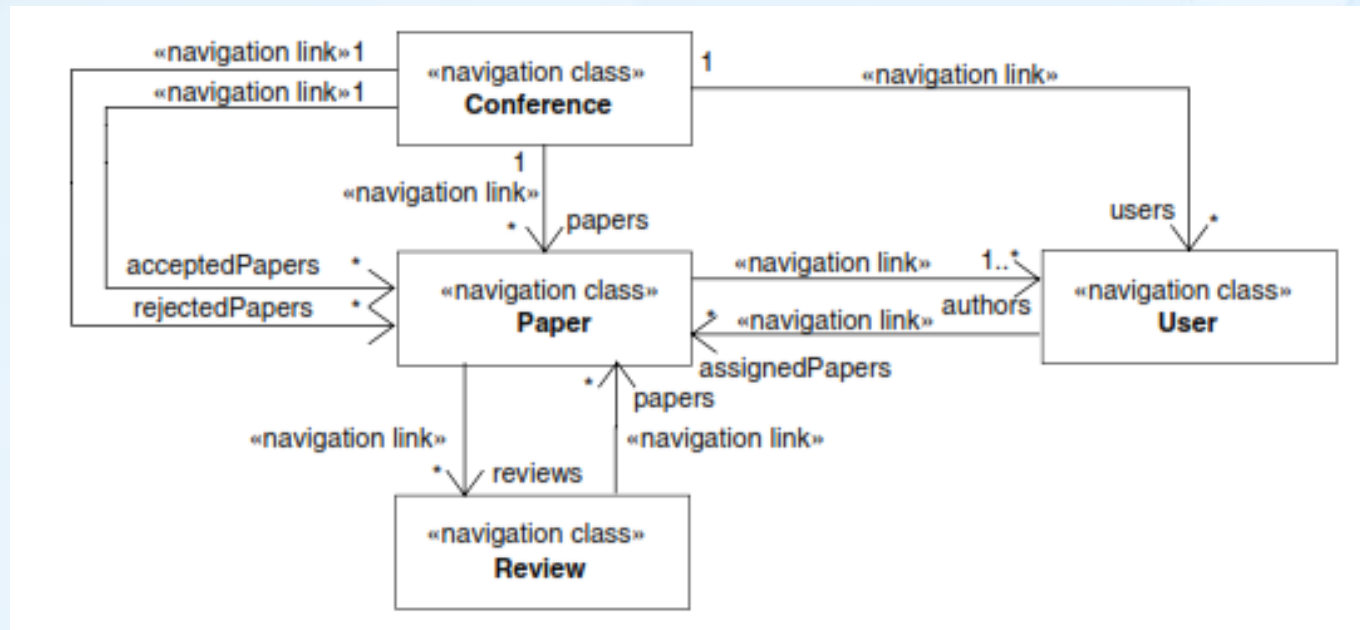
- Hypertext structure modeling is based on the concepts of hypertext, i.e., on nodes (also called pages or documents) and links between these nodes.
- The starting point used for the creation of a hypertext structure model is usually the content model which contains the classes and objects to be made available as nodes in the hypertext.
- Often the hypertext structure model is specified as a view on the content model and is therefore sometimes also called the **navigational view**. Thereby a node is specified as a view on the content model selecting one or more objects from the content.

Hypertext Structure Modeling Concepts

- Some methods even define transformation rules to derive links on the basis of relationships on the content level. Additional links can be added by explicit design decision.
- Other methods model the hypertext structure independently of the content model. For example the OOHDM (Object-Oriented Hypermedia Design Method) offers an approach to model scenarios, where the hypertext structure model can be built directly from the navigational requirements identified by these scenarios.

Hypertext Structure Modeling Concepts

- In the reviewing system example hypertext views are required for the following user roles: author, reviewer, and PC chair.
- The figure shows the hypertext structure model for the PC chair's view.
- A PC chair can view all submitted papers. In addition, the PC chair can access the list of accepted or rejected papers, and the reviewer profiles.
- The figure shows how the UML stereotype <<navigation class>> is used to mark classes representing nodes in the hypertext structure model to distinguish them from content classes.
- Links are modeled by directed associations with the stereotype <<navigation link>>.



(PC (Program committee)
= reviewer + administrative
work)

Hypertext modeling

- Regarding to the distribution of nodes or the hypertext level over pages on the presentation level, WebML specifies the following types of links:
 - **Intra-page links** are used, when a source and the destination of a link belong to the same page.
 - Example, when a link allows the user to directly navigate to the summary of a paper, which is displayed further down on the page.
 - **Inter-page links** are used when the source and the destination are on different pages.
 - Example, when detailed information about the authors and their papers are on different pages.

Hypertext modeling

- The following types of links are Based on the functional requirements of Web applications:
 - **Navigation links** *are used to navigate between nodes.*
 - *e.g., links between papers and their authors.*
 - **Process links** *point to the start node of a process.*
 - *e.g., to the beginning of the review submission.*
 - **External links** *point to a node not directly belonging to the application.*
 - *e.g., to the formatting guidelines established by the publisher of the conference proceedings, which are not directly stored in the reviewing system.*

Object Oriented Hypermedia modeling

The OO-H (Object-Oriented Hypermedia) modeling method defines five types of links as follows:

- **I-links or (internal links).**
 - They point to nodes inside the boundaries of a given navigational requirement, for example, internal links to review details of one of the reviewers.
- **T-links or (traversal links.)**
 - They point to nodes covering other navigational requirements, for example, from an author to his or her papers.
- **R-links or (requirement links).**
 - They point to a start of a navigational path, for example, to add a new review.
- **X-links or (external links).**
 - They point to external nodes, for example, to external formatting guidelines.
- **S-links or (service links).**
 - They point to services, for example, to an external search engine.

Outline

- Modeling Web Applications
 1. Content Modeling
 2. Hypertext Modeling
 3. Presentation Modeling

Presentation Modeling

- Similar to traditional Software Engineering, presentation modeling deals with the user interface and thus with the look and feel of a Web application. In contrast to traditional applications, the central element of the presentation in Web applications is the page as a visualization unit.
- Presentation modeling is aimed at designing the structure and behavior of the user interface to ensure that interaction with the Web application is simple and self-explanatory. In addition, the communication and representation task of the Web application are taken into account.

Presentation Modeling -User Interface

- Presentation modeling generates a two-fold result:
 - First: Structure of the page
 - First, the Presentation modeling produces a uniform presentation concept by modeling recurring elements on the pages, e.g., headers and footers.
 - It should ideally show the composition of each page and the design of the fields, texts, images, forms, etc., that are included in these pages.
 - Second: behavior-oriented aspects of the user interface.
 - Second, in addition to the structure of the pages, the presentation model describes the behavior-oriented aspects of the user interface, e.g., which button to click to activate a function of the application logic.
- Due to the wide variety of navigation options and the inherent risk of getting lost, care should be taken to give users appropriate orientation help on the presentation level. This can be achieved, for example, by displaying the current navigation path, or pages visited during the active session.

Presentation Modeling -User Interface

First: Structure of the page:

Model elements are described on three hierarchical levels:

- **A presentation page.**

- A presentation page describes a page presented to the user; it can be composed of different presentation units.

- **A presentation unit.**

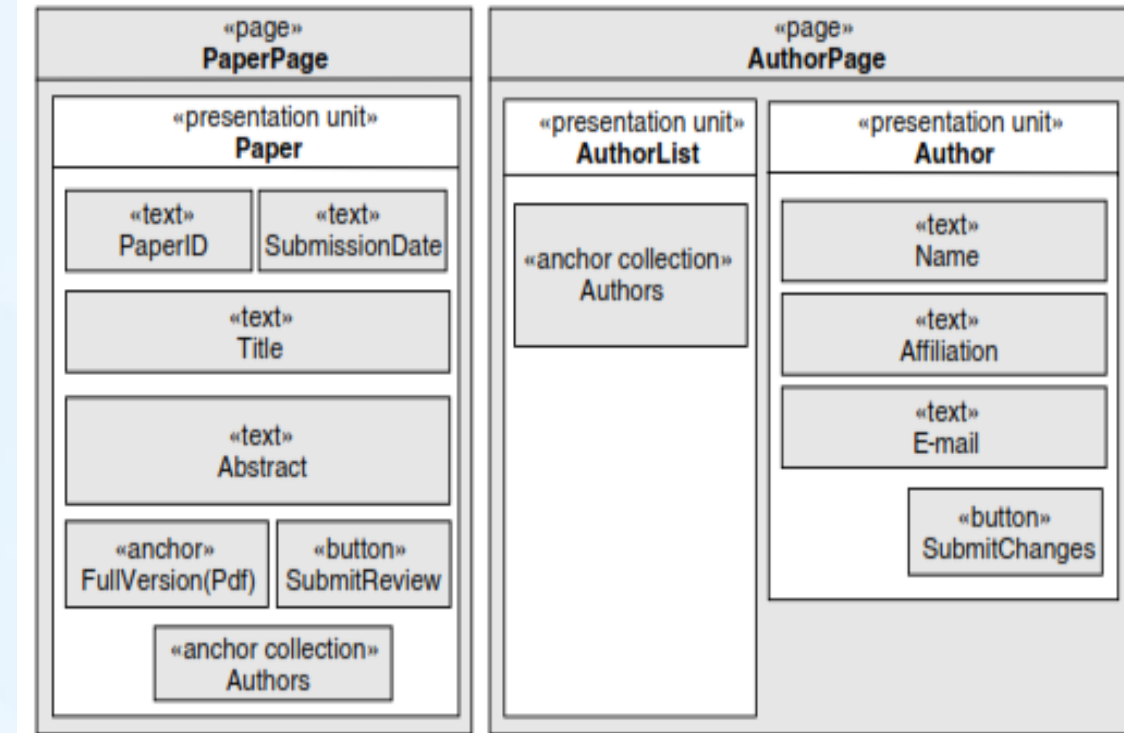
- A presentation unit serves to group related user interface elements, representing a logical fragment of the page.
- It presents a node from the hypertext model.

- **A presentation element.**

- A presentation element is the basic building block of the presentation model.
- Presentation elements represent a node's set of information, and they can include text, images, audio, and so on.

Presentation Modeling -User Interface

- We can visualize the composition of presentation pages on the basis of a nested UML class diagram representation known as “composition”, as shown in this Figure.
- This example uses the stereotype classes <<page>> and <<presentation unit>> to depict presentation pages and presentation units.
- The Figure shows two presentation pages of our reviewing system.
 - A paper is positioned on the page called “PaperPage” with the appropriate fields as well as a link to the paper’s full version and a link to display the paper’s authors.
 - Moreover, the user can press a button to add a new review. The page “AuthorPage” has two presentation units, i.e., the list of all authors and each author’s detailed information.

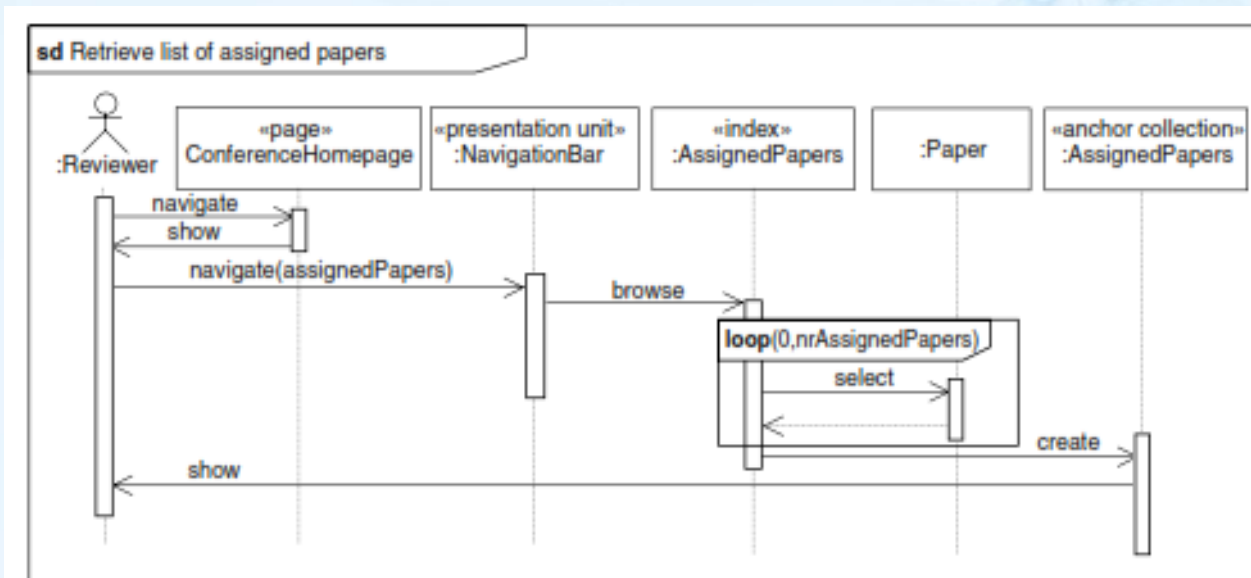


Presentation Modeling -User Interface

- **Second: Behavior of the page:**
- Behavioral aspects of the user interface, such as a reviewer's interaction to navigate to the papers assigned to him or her for reviewing, can be modeled by means of behavior diagrams.
- In general, a user's interaction with the Web application does not only involve the presentation level; it is also forwarded to the hypertext level and the content level, depending on the type of interaction.

Presentation Modeling -User Interface

- **Second: Behavior of the page:**
- The figure is a simplified sequence diagram that a reviewer activates the navigation to the index of assigned papers by using the navigation bar from within the conference home page.
- This information is, in turn, composed of the relevant papers on the content level. The list allows the user to select a paper out of the list of assigned papers. The user can then navigate to select one paper, which will be displayed in the details view.



Outline

Web Application Architectures

Web Application Architectures

The web architecture has three main important properties:

1. Architecture describes structure.

- The architecture of a software system consists of its structures, the decomposition into components, and their interfaces and relationships. It describes both the static and the dynamic aspects of that software system, so that it can be considered a building design for software.

2. Architecture forms the transition from analysis to implementation.

- The architecture try to break the functional requirements and quality requirements down into software components and their relationships and interfaces in an iterative approach.

3. Architecture makes a system understandable.

- Structuring software systems and breaking them down into different models allows better management of the complexity of software systems, and the systems become easier to be understood.

Functional Requirements for system architecture

- The architecture of an application is primarily influenced by **functional requirements**, i.e., the services provided by a system, and **quality considerations** such as scalability or performance.
- Apart from these requirements, architectures are further influenced by **technical constraints**, such as the used **system software** (e.g., the operating system), the **middleware** (e.g., a CORBA implementation), **legacy systems to be integrated**, **standards used**, development rules (e.g., coding guidelines), or **distribution aspects** (e.g., the distribution over different locations of a company).
- Moreover, the **software architect's experiences** play a considerable role in the definition of architecture.

The Architectures for Web Application Environments

- **Layering Aspect**

- Layering means that software systems are structured in several tiers to implement the principle of "separation of concerns" within a software system.
- Web-based applications run on a web application server and access data on an enterprise information system. The components of web-based applications are spread across multiple tiers, or layers.
- In general, the user interface is on the first tier, the application programs are on the middle tier, and the data sources that are available to the application programs are on the enterprise information system tier.
- Developing web-based applications across a multi-tiered architecture is referred to as **server-side programming**.
- Many frameworks in the field of distributed systems and Web applications are primarily structured by the layering aspect.

The Architectures for Web Application Environments

- **Data Aspect**

- Data can be either structured or Non-structured,
 - **Structured data** follow a defined scheme like tables in a relational database or XML structures in a document.
 - **Non-structured data** are multimedia contents, example, images, audio, and video, which typically do not follow an explicit scheme.
- Which is more difficult to process?
 - The automatic processing of non- structured data is difficult, as they don't follow an explicit scheme.

The Architectures for Web Application Environments

Middleware:

- Because businesses, institutions, and technologies change continually, the software systems that serve them must be able to accommodate such changes.
- Following a merger, the addition of a service, or the expansion of available services, a business can find difficulty to afford the recreation of its information systems. It is at this most critical point that it needs to integrate new components or to scale existing ones as efficiently as possible.

The Architectures for Web Application Environments

Middleware:

- The easiest way to integrate heterogeneous components is not to recreate them as homogeneous elements but to provide a layer that allows them to communicate despite their differences.
- This layer, called **middleware** which allows software components (applications, enterprise java beans, servlets, and other components) that have been developed independently and that run on different networked platforms to interact with one another.

The Architectures for Web Application Environments

Middleware:

- Middleware can be grouped into the following categories:
 - **Remote Procedure Call or RPC-based middleware**, which allows procedures in one application to call procedures in remote applications as if they were local calls. The middleware implements a linking mechanism that locates remote procedures and makes these transparently available to a caller. Traditionally, this type of middleware handled procedure-based programs; it now also includes object-based components.
 - **Object Request Broker or ORB-based middleware**, which enables an application's objects to be distributed and shared across heterogeneous networks.
 - **Message Oriented Middleware or MOM-based middleware**, which allows distributed applications to communicate and exchange data by sending and receiving messages.

The Architectures for Web Application Environments

The Architectures for Web Application Environments also take the distribution of data and messages into account such as :

- Distributed Object Middleware (DOM).
- Virtual Shared Memory (VSM).
- Message Oriented Middleware (MOM).
- Peer to Peer (P2P).
- Service Oriented Middleware (SOM).

The Architectures for Web Application Environments

Distributed Object Middleware (DOM):

- DOM is the software modules that are designed to work together, but reside either in multiple computers connected via a network or in different processes inside the same computer.
- This type of infrastructure involve location transparency where remote objects appear the same as local objects
- It is based on the Remote Procedure Call (RPC) mechanism.

The Architectures for Web Application Environments

Virtual Shared Memory (VSM):

- Virtual shared memory (VSM) is a technique through which multiple processors within a distributed computing architecture are provided with an abstract shared memory. The idea is to implement a coherent shared memory on a network of processors without physically shared memory.
- This data can be virtually "anywhere" in the system.

The Architectures for Web Application Environments

Message Oriented Middleware (MOM):

- MOM systems offer functionalities for asynchronous transmission of messages.
- Asynchronous communication means messages that are sent to the receiver regardless of its status, e.g., the receiver may not be available, when the message is sent, i.e., he or she may be offline.
- MOM ensures that messages are delivered, nevertheless.
- MOM makes use of messaging provider to mediate messaging operations.
- The basic elements of a MOM system are clients, messages, and the MOM provider, which includes an API (Application Programming Interface) and administrative tools.
 - **API** (Application Programming Interface) is a software intermediary that allows two applications to talk to each other.
- The MOM provider uses different architectures to route and deliver messages: it can use a centralized message server or it can distribute routing and delivery functions to each client machine. Some MOM products combine these two approaches.

The Architectures for Web Application Environments

Peer to Peer (P2P):

- Peer to peer architecture is a type of computer networking architecture in which there is no division or distinction of abilities among the various nodes of a network. Every section has the exact same responsibilities and can perform the same set of actions.
- **Among the Applications of Peer to Peer Architecture**
 - Easy file sharing
 - Efficient instant messaging
 - Smooth voice communication
 - Secure search and communication network

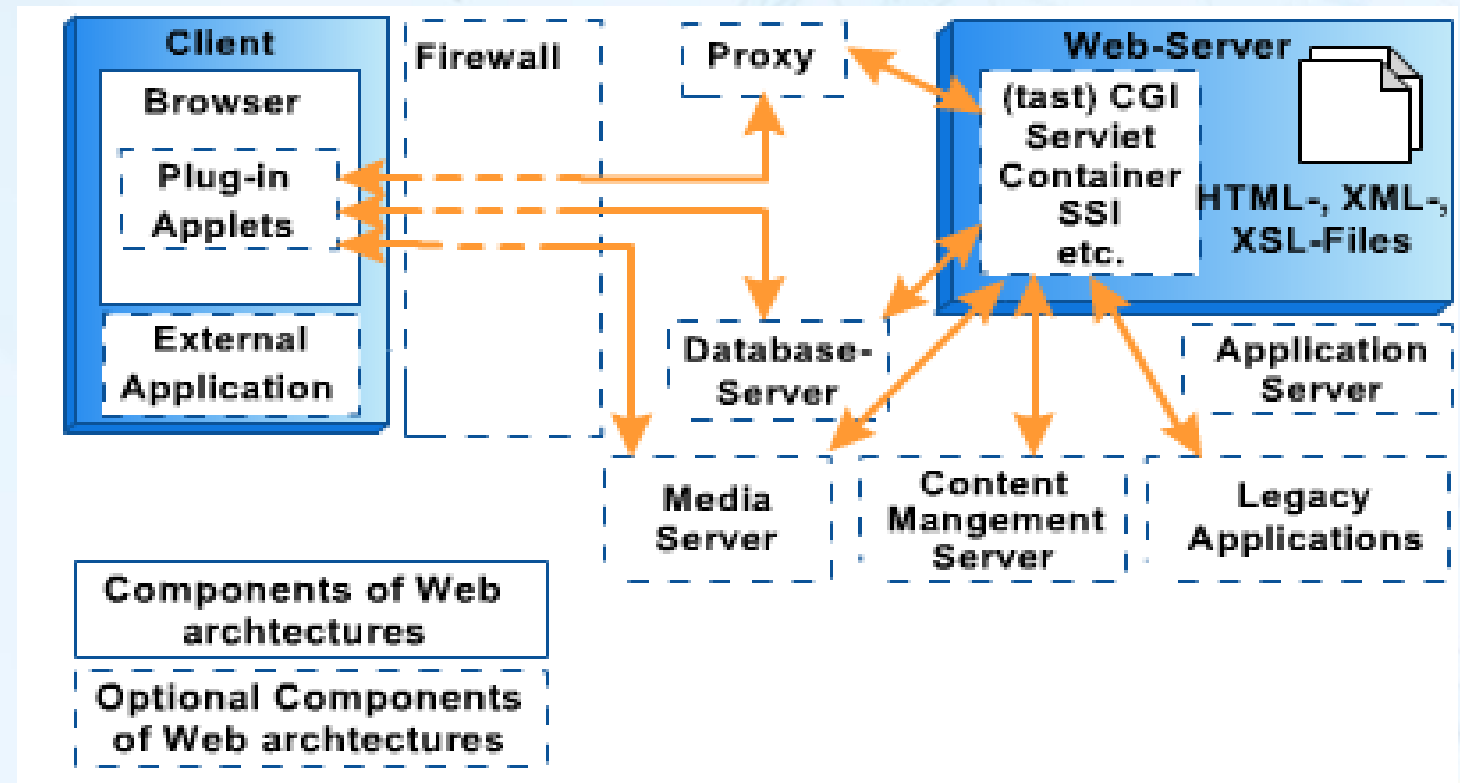
The Architectures for Web Application Environments

- **Service Oriented Middleware (SOM):**

- Service Oriented Architecture (SOA) is a relationship between the service provider, broker and user. Designing middleware platforms for this architecture really helps in achieving interoperability, loose coupling and high efficiency especially for applications running in heterogeneous environments. It uses a defined interface to make a service available for other systems/services.
- The job of an SOM is to serve the functional properties in addition to supporting the non-functional properties which include scalability, service reliability, service flexibility and Quality of Service (QoS) assurance.

Web architecture component

- The following list briefly describes the basic components of Web architectures and their relationships:
 - Client
 - Web server
 - Firewall
 - Proxy
 - Database server
 - Media server
 - Content management server
 - Application server



Web architecture component

- **Client:**

- User agent that is controlled by a user to operate the Web application.

- **Web server:**

- A Web server is a piece of software that supports various Web protocols like HTTP, and HTTPS, etc., to process client requests.

Web architecture component

- **Firewall:**

- A piece of software regulating the communication between insecure networks (e.g., the Internet) and secure networks (e.g., corporate LANs). This communication is filtered by access rules.

- **Proxy:**

- A proxy is typically used to temporarily store Web pages in a cache.
- However, proxies can also assume other functionalities, e.g., adapting the contents for users (customization), or user tracking.

Web architecture component

- **Database server:**

- This server normally supplies an organization's production data in structured form, e.g., in tables.

- **Media server:**

- This component is primarily used for content streaming of non-structured bulk data (e.g., audio or video).

Web architecture component

- **Content management server:**

- Similar to a database server, a content management server holds contents to serve an application.
- These contents are normally available in the form of semi-structured data, e.g., XML documents.

- **Application server:**

- An application server holds the functionality required by several applications, e.g., workflow or customization.

True /False

In Content model in web applications, the structure of the information will remain unchanged, even if the information itself changes frequently.	T
Analysis models will typically feature a mix of conceptual and specification perspectives.	T
The starting point used for the creation of a hypertext structure model is usually the presentation model	F
The OOHDM (Object-Oriented Hypermedia Design Method)models the hypertext structure independently of the content model.	T

MCQ

- In UML class notation, the mandatory information is/are
 - a. Class Name
 - b. Class Attributes
 - c. Class Operations
 - d. Both class name and class attributes
- In OO-H (Object-Oriented Hypermedia) modeling, T-links (traversal links) point to:
 - a. nodes inside the boundaries of a given navigational requirement
 - b. a start of a navigational path
 - c. nodes covering other navigational requirements
 - d. external nodes



EELU

الجامعة المصرية للتعليم الإلكتروني الأهلية
THE EGYPTIAN E-LEARNING UNIVERSITY

THANK YOU FOR WATCHING

QUESTIONS?

