**MULTIMEDIA AND ANIMATION**

**BRANCH: CS/IT**

**YEAR:FIRST**

**SEM:SECOND**

**MULTIMEDIA AND ANIMATION**

**BY NAVNEET SOLANKI (LECTURER -IT)AND PRASHANT KUMAR(LECTURER--CS)**

 UNIT-2

 **Multimedia Systems Architecture**



• While focusing on multimedia system architecture, the most important point under consideration is on multimedia applications.

• Applications can be training app, conferencing app, messaging app, education app.

• After that multimedia application interface with windows or with presentation manager.

• API’s support wide range of applications for publishing and interacting with applications.

• Multimedia applications such as educational, online training electronic messaging, and video conferencing kind of applications we are using at very first level of architecture.

• Multimedia architecture is of three types:

• Multimedia workstation architecture

• IMA architectural framework

• Network Architecture for multimedia systems

 **DISTRIBUTED MULTIMEDIA**

**A distributed system** is the collection of autonomous computers that are connected using a communication network and they communicate with each other by passing messages. The different processors have their own local memory. They use a distribution middleware. They help in sharing different resources and capabilities to provide users with a single and integrated coherent network.

Distributed computing is a field of computer science that studies distributed systems and the computer program that runs in a distributed system is called a distributed program.

A distributed system requires concurrent Components, communication network and a synchronization mechanism. A distributed system allows resource sharing, including software by systems connected to the network.

**Examples of distributed systems / applications of distributed computing :**

Intranets, Internet, WWW, email.

Telecommunication networks: Telephone networks and Cellular networks.

Network of branch office computers -Information system to handle automatic processing of orders,

Real-time process control: Aircraft control systems,

Electronic banking,

Airline reservation systems,

Sensor networks,

Mobile and Pervasive Computing systems.

**DISTRIBUTED MULTIMEDIA SYSTEMS**

If the multimedia systems are supported by multiuser system, then we call those multimedia systems as distributed multimedia systems.

A multi user system designed to support multimedia applications for a large number of users consists of a number of system components. A typical multimedia application environment consists of the following components:

1.     Application software.

2.     Container object store.

3.     Image and still video store.

4.     Audio and video component store.

5.     Object directory service agent.

6.     component service agent.

7.     User interface and service agent.

8.     Networks (LAN and WAN).

**Application Software**

The application software perfom1s a number of tasks related to a specific business process. A business process consists ofa series of actions that may be performed by one or more users.

The basic tasks combined to form an application include the following:

**(1) Object Selection**- The user selects a database record or a hypermedia document from a file system,database management system, or document server.

(**2) Object Retrieval**- The application ret:ieves the base object.

(**3) Object Component Display** - Some document components are displayed automatically when the user moves the pointer to the field or button associated with the multimedia object.

**(4)** **User Initiated Display**- Some document components require user action before playback/display.

**(5)** **Object Display Management and Editing**: Component selection may invoke a component controlsubapplication which allows a user to control playback or edit the component object.

**Document store**

A document store is necessary for application that requires storage oflarge volume of documents. The following describes some characteristics of document stores.

**1.**  **Primary Document Storage**: A file systems or database that contains primary document objects(container objects). Other attached or embedded documents and multimedia objects may be stored in the document server along with the container object.

**2.** **Linked Object Storage**: Embedded components, such as text and formatting information, and linkedinformation, and linked components, such as pointers to image, audio, and video. Components contained in a document, may be stored on separate servers.

**3.** **Linked Object Management**: Link information contains the name of the component, service class ortype, general attributes such as size, duration of play for isochronous objects and hardware, and software requirements for rendering.

**Image and still video store**

An image and still video is a database system optimized for storage of images. Most systems employ optical disk libraries. Optical disk libraries consist of multiple optical disk platters that are played back by automatically loading the appropriate platter in the drive under device driver control.

The characteristics of image and still video stores are as follows:

(i) Compressed information (ii) Multi-image documents

(iii)Related annotations (iv) Large volumes

(v)Migration between high-volume such as an optical disk library and high-speed media such as magnetic cache storages(vi) Shared access: The server software managing the server has to be able to manage the different requirements.

**Audio and video Full motion video store**

Audio and Video objects are isochronous. The following lists some characteristIcs of audio and full-motion video object stores:

(i) Large-capacity file system: A compressed video object can be as large as six to ten M bytes for one minute of video playback.Temporary or permanent Storage: Video objects may be stored temporarily on client workstations, servers PFoviding disk caches, and multiple audio or video object servers. Migration to high volume/lower-cost media. Playback isochronocity: Playing back a video object requires consistent speed without breaks. Multiple shared access objects being played back in a stream mode must be accessible by other users.

**Object Directory Service Agent**

The directory service agent is a distributed service that providea directory of all multimedia objects on the server tracked by that element of the directoryy service agent.

The following describes various services provided  by a directory service Agent.

(1)Directory Service: It lists all multimedia objects by class and server location.

(2)   Object Assignment: The directory service agent assigns unique identification to each multimedia object.

(3)Object Status Management: The directory service must track the current usage status of each object.

(4)Directory Service Domains: The directory service should be modular to allow setting up domains constructed around groups of servers that form the core operating environment for a group of users.

(5)  Directory Service Server Elements: Each multimedia object server must have directory service element that reside on either server or some other resources.

(6)Network Access: The directory service agent must be accessible from any workstation on the network.

**Component Service Agent**

A service is provided to the multimedia used workstation by each multimedia component. This service consists of retrieving objects, managing playback of objects, storing objects, and so on. The characteristics of services provided by each multimedia component are object creating service, playback service, component object service agent, service agents on servers and multifaceted services means (multifaceted services component objects may exist in several forms, such as compressed Or uncompressed).

**User Interface Service Agent**

It resides on each user workstation. It provides direct services to the application software for the management of the multimedia object display windows, creation and storage of multimedia objects, and scaling and frame shedding for rendering of multimedia objects.

The services provided by user interface service agents are windows management, object creation and capture, object display and playback, services on workstations and using display software. The user interface service agent is the client side of the service agents. The user interface agent manages all redirection since objects are located by a look-up mechanism in the directory service agent

**Distributed client server operation**

The agents so far we have discussed combine to form a distributed client-server system for multimedia applications. Multimedia applications require functionality beyond the traditional client server architecture.

Most client-server systems were designed to connect a client across a network to a server that provided database functions. In this case, the client-server link was firmly established over the network. There was only one copy of the object on the specified server. With the development of distributed work group computing, the picture has changed for the clients and servers. Actually in this case, there is a provision of custom views in large databases. The advantage of several custom views is the decoupling between the physical data and user.

The physical organization of the data can be changed without affecting the conceptual schema by changing the distributed data dictionary and the distributed data repository.

**Clients in Distributed Work Group Computing**

Clients in distributed workgroup computing are the end users with workstations running multimedia applications. The client systems interact with the data servers in any of the following *w3fs.*

1.     Request specific textual data.

2.     Request specific multimedia objects embedded or linked in retrieved container objects.

3.     Require activation of a rendering server application to display/ playback multimedia objects.

4.     Create and store multimedia-objects on servers.

Request directory information. on locations of objects on servers

**Servers in Distributed Workgroup Computing**

Servers are storing data objects. They provide storage for a variety f object classes, they transfer objects on demand on clients. They rovide hierarchical storage for moving unused objects to optical\_ isk lirbaries or optical tape libraries. They provide system dministration functions for backing up stored data. They provide le function of direct high-speed LAN and WAN server-to-server ~ansport for copying multimedia objects.

**Middleware in Distributed Workgroup Computing**

The middleware is like interface between back-end database and font-end clients.The primary role of middleware is to link back end database to front end clients in a highly flexible and loosely connected network nodel. Middleware provides the glue for dynamically redirecting client requests to appropriate servers that are on-line.

Multimedia Object Servers The resources where information objects are storedareknown as servers. Other users (clients) can share the information stored in these resources through the network.

**Types of Multimedia Servers**

Each object type of multimedia systems would have its own dedicated server optimized for the type of data maintained in the object. A network would consist of some combination of the following types of servers.

**(1) Data-processing servers RDBMSs and ODBMSs. (2) Document database servers.**

**(3) Document imaging and still-video servers. (4) Audio and voice mail servers.**

**(5) Full motion video server.**

Data base processing servers are traditional database servers that contain alphanumeric data. In a relational database, data fields are stored in columns in a table. In an object-oriented database these fields become attributes ofthe object. The database serves the purpose of organizing the data and providing rapid indexed access to it. The DBMS can interpret the contents of any column or attribute for performing a search.

**Mass Storage for Multimedia Servers**

**RAID(Redundant Arrays of Inexpensive Disks)**

In terms of redundancy,RAID provides a more cost effective solution than disk mirroring.

RAID is a means of increasing disk redundancy, RAID systems use multiple and potentially slower disks to achieve the same task as a single expensive large capacity and high transfer rate disk.

In RAID high transfer rates are achieved by performing operations in parallel on multiple disks. There are different levels of RAID available, namely disk striping(level 0), disk mirroring(level 1, Bit interleaving of date(level 2), Byte interleaving (level 3), sector interleaving(level 4), and block interleaving(level 5)RAID technology is faster than rewritable optical disk and high data volumes can be achieved with RAID. RAID technology provides high performance for disk reads for almost all types of applications.

**Write Once Read Many Optical Drives: (WORM)**

WORM Optical drives provide very high volumes of storage for very low cost. Some important characteristics of WORM optical disks are:

Optical drives tend to be slower than magnetic drives by a factor of three to four.         .

WORM drives can write once only; typically 5-10% of disk capacity m left free to provide for changes to existing information.

They are useful for recording informations that would not change very much. They are virtually indestructible in normal office use and have long shelf lives.

They an be used in optical disk libraries (Juke boxes). A Juke box may provide anywhere from 50-100 disk platters with two or more drives.

These characteristics make optical disks ideal candidates for on-line document images (which change very little once scanned and do not have an isochronous requirement) and archived data.

**Rewritable Optical Disks**:

Rewritable optical drives are produced by using the technologies like magneto-optical. It has the advantage ofrewritability over the WORM where rewritable is not possible. It can be used as primary or secondary media for storage of large objects, which are then archieved. (Placed where documents are preserved) on WORM disks.

If it is used as primary media, it should be accompanied by highspeed magnetic disk cache. This is to achieve acceptable video performance.

**Optical Disk Libraries**:

Optical disk libraries are nothing but juke boxes. Work disks and rewritables can be used in optical disk libraries to achieve very high volumes of near-lines storage. Optical disk libraries range from desk top juke boxes with one 5' 1/4" drive and I O-slot optical disk stack for upto lOG Bytes of stroage of large libraries using as many as four 12" drives with an 80-s10t optical disk stack for upto terabytes of storage. The disadvantage of optical disk library is the time taken for a platter to be loaded into a drive and span to operating speed.

**Network Topologies for Multimedia Object Servers**

A number of network topologies are available Network topology is the geometric arrangement of nodes and cable links in a network. We still study three different approaches to setting up multimedia servers.

(i) **Centralized Multimedia Server**: A centralized multimedia object server performs as a central store for multimedia objects. All user requests for multimedia objects are forwarded by the applications to the centralized server and are played back from this server. The centralized server may serve a particular site of the corporation or the entire enterprise. Every multimedia object has a unique identity across the enterprise and can be accessed from any workstation. The multimedia object identifier is referenced in every data that embeds or links to it.

**Dedicated Multimedia Servers**: This is the approach where a video server is on a separatededicatedsegmentIn this approach, when a workstation dumps a large video, the other servers on the networks are not affected. Provides high performance for all local operations. The isochronocity of the objects is handled quite well in a dedicated mode.

Disadvantage of this approach is that the level of duplication of objects.

**Distributed multimedia servers:**

In this approach multimedia object servers are distributed in such a manner that they are placed in starategic locations on different LANs.They are replicated on a programmed basis to provide balanced serviceto all users.

**Multiserver Network Topologies**

To distribute the full functionality of multimedia network wide there are vareity of network topologies available. ' The primary topologies are Traditional LANs (Ethernet or Token Ring Extended LANs (Using network switching hubs bridges and routers). ' High speed LANs (ATM and FDDI II). WANs (Including LANs, dial-up links-including ISDN T1 and T3 lines-etc.). ' I

**Traditional LANS (Ethernet or Token Ring) Ethernet:**

**Ethernet**: It is a Local Area Network hardware, communication, and cabling standard originallydeveloped by Xerox corporation that link up to 1024 nodes in a bus network. It is ahigh speed standard using a baseband (single-channel) communication technique. It provides for a raw data transfer rate of 10 Mbps, with actual throughput in the range of 2-3 Mbps. It support a number of sessions ina mixof live video, audio electronic mail and so on.

**Token Ring**: It is a Local Area Network architecture that combines token passing with a hybridstar/ring topology. It was developed by IBM. Token Ring Network uses a multistation Access unit at its hub ..

**ATM (Asynchronous Transfer Mode)**

It is a network architecture that divides messages into fixed size units (called cells) of small size and that establishes a switched connection between the originating and receiving stations.

A TM appears to be a potential technology for multimedia systems for connecting object servers and user workstations. ATM is actually a good candidate for two reasons: as a hub and spoke teclmology, it adapts very well to the wiring closest paradign; and it allows workstations to operate at speeds defined by the workstation. Figure 5.12 below illustrates LAN topology using an A TM Switching System.

**FDDI II (Fiber Distributed Data Interface II)**

It is a standard for creating highspeed computer networks that employ fiber-optic cable. FOOl II operates exactly like token ring, . with one difference: FOOl employs two wires through all the hosts in a network.

FOOl II is a single media LAN and its full bandwidth supports all users.

FOOl II appears to be a very useful high-speed technology for connecting servers on an additional separate network and providing the dedicated high bandwidth necessary for rapid transfer and replication of information objects. Figure 5.13 shows a multiievel network based

**WANS (Wide Area Network)**

This includes LANs, dial up ISDN, T1 (1.544 Mbits/sec) and T3 (45.3 Mbits/sec) lines and regular telephone dial-up lines. The two big issues here are:

:.W ANs may have a mix of networking and communication protocols.

:. WAN has a variety of speeds at which various parts of it where it communicates. Protocol Layering: Layering helps to isolate the network from the . application. Layering of protocols started with the release ofthe ISO model.

**Distributed Multimedia Databases:**

A multimedia database consists of a member of different types of multimedia objects. These may include relational database records, object-oriented databases with objects for alphanumeric attributes, and s:orage servers for multimedia objects such &s images, still video, audio, and full-motion video. It is feasible to include an image or a video object as a binary large object (BLOB) in a relational database.

It is also feasible to include such an object as an attribute in an object.

**Database Organization for Multimedia Applications**

Optical disk storage technology has reduced the cost of multimedia document storage by a significant factor. Distributed architectures have opened the way for a variety of applications distributed around a network accessing the safe database in an independent manner. The following discussion addresses some key issues of the data organization for multimedia systems.

**Data Independence:**Flexible access to a variety of distributed databases for one or more applicationsrequires that the data be independent from the application so that future applications can access the data without constraints related to a previous application. Important features of data independent design are:

1.     Storage design is independent of specific applications.

2.     Explicit data definitions are independent of application programs.

3.     Users need not know data formats or physical storage structures .

4.     Integrity assurance is independent of application programs.

5.     Recovery is independent of application programs .

**Common Distributed Database Architecture:**Employment of Common Distributed databasearchitectured is presented by the insulation of data from an application and distributed application access.

Key features of this architecture are:

1.The ability for multiple independent data structures to co-exist in the system (multiple server classes).

2.Uniform distributed access by clients.

3.Single point for recovery of each database server. 4.Convenient data re-organization to suit requirements. 5.Tunability and creation of object classes. 6.Expandibility.

Mnltiple Data Servers: A database server is a dedicated resource on a network accessible to a number of apphcations, When a large number of users need to access the same resources, problem arises

This problem is solved by setting up multiple data servers that have copies of the same resources,

**Transaction management for Multimedia Systems**

It is defined as the sequence of events that starts when a user makes a request to create, render, edit, or print a hypermedia document. The transaction is complete when the user releases the hypermedia document and stores back any edited versions or discards the copy in memory or local storage.

Use of object classes provides an excellent way for managing and tracking hypermedia documents, Given that all components of a hypermedia document can be referenced within an object as attributes, we can find a solution for the three-dimensional transaction management problem also in the concept of objects.

Andleigh and Gretzinger expand on the basic concepts developed for the object request broker (ORB) by the Object Management Group (OMG) and combine it with their transaction management approach.

**Managing Hypermedia Records as Objects**

Hypermedia records or documents are complex objects that contain multimedia information objects within them, A hypermedia document can be stored in a document data base, as a BLOB in a relational database, or in an object-oriented data base. A Hyper media document may contain multimedia objects embedded in it as special fields.

**Object linking and embedding:**OLE provides an object oriented framework for compounddocuments. When a user double cliks or click on an icon for an embedded object, the application that created the object starts, and allows the user to view andor the object .

**Managing Distributed Objects:**We see the nature of communication between servers and themanaging of distributed objects.

**Interserver communications:**Object replication , object distribution, object recompilation andobject management and network resources are some of the design requirements that play a role in defining interserver The following lists the types of communications that one server may make to another server:

1.     Obtain a token from an object name server for creating a new multimedia object; the object is not accessible by others users until complete and released.

2.     Search the object class directory for the current locations of that object and the least expensive route for accessing it.

3.     Perform a shared read lock on the object to ensure that it is n archived or purged while it is being retrieved.

4.     Replicate a copy of the object; update the object name server directory.

5.     Copy an object for non-persistent use.

6.     Test and set an exclusive lock on an object for editing purposes'

7.     create new versions.

8.     Pause the retrieval of an object to support a user action or to pace the retrieval to the speed supported by the network.

9.     A Sound server architecture is necessary for providing these services in a fully distributed environment.

**Object Server Architecture**

**Figure**describes an object server architecture that can support multimedia applications for a largenumber of users.



The architecture describes the logical distribution of functions. The following lists the key elements of this architecture:

Multimedia Applications -Common Object Management API.Object Request Broker. Object Name Server -Object Directory Manager -object Server -Object Manager.Network Manager . Object Data Store. Any multimedia application designed to operate on the common object management API can function in this architecture

The common object management API is a programming interface definition that provides a library of functions the applications can call.

The common Broker Architecture API provides a uniform interface to all applications and a standardized method for managing all information objects in a corporate network.

A common Object Request Broker Architecture (CORBA) h been defined by a Object Management Group. An object request broker performs the following functions:

**(i)**  **Object recompilation.**

**(ii)** **Playback control.**

**(iii)** **Format conversions.**

The object name server provides an object directory service. The object directory manager may exist in a distributed form within an object server. The object directory manager updates the object directory when changes take place.The object server is a logical subsystem in the network responsIble for storing and retrieving objects on demand. The object manager consists of a number of object classes that performs a number of specialized services. They are: Object retrieval. (ii) Replication(Ui) Migration. (iv) Transaction and Lock Management. (v) User Preferen'ce. (vi) Versioning. (vii) System Administration. (ix) Archival. (x) Purging. (xi) Class-Specific functions.

**Identification method:**Objects can be distinguished from one another in many potential ways.Identification of objects in a persistent state is different from non-persistent objects. At the highest level, persistent objects are distinguished by the class of objects. Andleigh and Gretzinger defined a rule for unique object identification as follows:

**ROLE:**An object must have an identifier that is unique in a time dimension as well as with locationsuch that it cannot be modified by any programmed action. An alternative approach is to divide the network into domains and have a name server in each domain be responsible for assigning new object IDs for all objects created in that domain. An object identification algorithm can be made unique by combining several of the following components.

.:. Network domain name. --Address and server ID of the name server node.--A time stamp of creating time.-- An object class identifier.

**Object Directory services**

A multimedia object directory manager is the name server for all multimedia objects in a LAN. It has an entry for every multimedia object on all servers on the LAN, or in a domain if a LAN or WAN is subdivided into domains. The object directory manager manages changes to the object directory resulting from object manager actions.

**Multimedia Object Retrieval**

The multimedia object manager performs the functions of managing all requests from the multimedia applications for retrieving existing multimedia objects *01'* storing new or edited multimedia objects created by the user. In systems actively designed using an object request broker, this request is channeled through the object request broker. Data structure maintained by the multimedia object manager:

**Database Replication Techniques**In the simplest fonn of data management, the databases are set up asduplicates of the databases. Database duplication ensures that the multiple copies are identical.

There is an approach to allow each copy of the database to be modified as needed and to synchronize them by comparing them and copying the changes to all other database copies on a very frequent basis, this process is called replication.

**Types of Database Replication**: There are eight types of modes available. They are: Round Robinreplication.2.Manual replication. (Hi) Scheduled replication. (iv) Immediate replication. V)Replication-on-demand. Vi) Predictive replication. Vii) Replication references. Viii)No replication. **Object**

**Migration Schemes**

**Optimizing Object Storage**A number of techniques are available for optimizing data storage formultimedia objects. Let us consider the three design approaches

**1.**  **Optimizing Servers by Object Type:**

The mechanism for optimizing storage is to dedicate a server to a particular type of object. The object server may ne designed to provide specialized services for specific object classes related to rendering

2.     **Automatic Load Balancing:**It can be achieved by programming the replication algorithm tomonitor use counts for each copy of a replicated object.

**3.**     **Versioned Object Storage:**

The storage problem will be more complex if multiple vcrsions need to be stored. Hence, we should follow the technique which is based on saving changes rather than storing whole new objects. New versions of the object can be complex objects,

Synchronization in multimedia

**Synchronization in multimedia** systems refers to temporal relationships between **media objects** in the **multimedia** systems. ... **Synchronization** between **media objects** comprises relationships between time-dependent **media objects** as well as time-independent **media objects**.

**Synchronization in multimedia** systems refers to temporal relationships between **media objects** in the **multimedia** systems. ... **Synchronization** between **media objects** comprises relationships between time-dependent **media objects** as well as time-independent **media objects**.

Synchronization in multimedia systems refers to temporal relationships between media objects in the multimedia systems. In future multimedia systems (based, e.g., on MPEG-4) synchronization may also refer to spatial and content relationships, as well as temporal.
Synchronization between media objects comprises relationships between time-dependent media objects as well as time-independent media objects. Synchronization may need to occur at different levels in a multimedia system, consequently synchronization support is typically found in the operating system, communication system, databases, multimedia documents, and the application. A general scheme might involve a layered approach to achieving synchronization. For example, a Computer-Supported Collaborative Workgroup (CSCW) session might involved a multi-party video conferencing session with audio, and a shared whiteboard. Parties may make reference to objects on the shared whiteboard, using a pointer, to support what they are saying (e.g., saying "This area here..." while indicating the area with a pointer). Here, video and audio are continuous media objects which are highly periodic, whereas the shared whiteboard is a discrete media stream, as changes to it are highly irregular (the content, including the position of the pointer, depends on which participant has control of the object and when they make changes to it). The media streams must be highly synchronized, so that speech remains lip synchronized, and the whiteboard updates are synchronized with audio references to them.
The operating system and lower levels of the communication system are responsible for ensuring that jitter on individual streams does not occur during presentation of the video, audio, and whiteboard streams (intramedia synchronization). At a higher level, the runtime support for the synchronization of multiple multimedia media streams must ensure that the various media streams remain synchronized with respect to each other (intermedia synchronization). Finally, the application(s) are responsible for ensuring synchronicity between application-level events (usually initiated by the users). For example, if the application at the source does not capture timing dependencies between a user waving the pointer over part of the object in the whiteboard and the supporting audio stream, then it will be impossible for the application at the sink to know that the whiteboard and audio events need to be synchronized.
The temporal relations between media objects must be specified during capture of the media objects, if the goal of the presentation is to present media in the same way that they were originially captured. Synchronization information of events in an animation sequence or a slide show is usually specified by the designer, using, for example, a time-axis.

**A Reference Model for Multimedia Synchronization**

A reference model is needed to understand the requirements of multimedia synchronization, identify and structure runtime mechanisms that can support these requirements, identify interfaces between runtime mechanisms, and compare solutions for multimedia synchronization systems.
Figure 11.1 shows a reference model for multimedia synchronization systems. Each layer implements synchronization mechanisms which are provided by an appropriate interface. These interfaces can be used to specify or enforce the temporal relationships. Each interface can be used by the application directly, or by the next higher layer to implement an interface. Higher layers offer higher programming and QoS abstractions.



**Media Layer**

An application operates on a single continuous media stream, which is treated as a sequence of LDUs. Networking components must be taken into account. Provides access to files and devices.

**Stream Layer**

The stream layer operates on continuous media streams as well as groups of media streams. In a group, all streams are presented in parallel by using mechanisms for interstream synchronization. QoS parameters will specify intrastream and interstream synchronization requirements.
Continuous media is seen as a data flow with implicit time constraints; individual LDUs are not visible. An application using the stream layer is responsible for starting, stopping and grouping the streams, and for the definition of the required QoS in terms of timing parameters supported by the stream layer. It is also responsible for the synchronization with time-independent media objects. Tasks include resource reservation and LDU process scheduling.

**Object Layer**

The object layer operates on all media streams and hides the differences between continuous and discrete media. An application that interacts with this layer will be presented with a view of a complete, synchronized presentation. This layer takes a complete synchronization specification as its input and is responsible for the correct schedule of the overall presentation.

**Specification Layer**

This layer contains applications and tools that are allowed to create synchronization specifications (e.g., authoring tools, multimedia document editors).
The specification layer is also responsible for mapping user-required QoS parameters to the qualities offered at the object layer interface.
Synchronization specifications can be:

* Interval-based: specifications of the temporal relations between the time intervals of the presentation of media objects
* Axes-based: allows presentation events to be synchronized according to shared axes, e.g., a global timer
* Control flow-based: at specified points in presentations, they are synchronized
* Event-based: Events in the presentation trigger presentation actions

**Synchronization in a Distributed Environment**

Synchronization in a distributed environment is complex, because there may be more than one source of multimedia data, and more than one sink consuming it. The synchronization information for the various media stream may also reside at different sources.

**Transport of the synchronization specification**

The sink needs to have the synchronization information available to correctly display an object. There are three main approaches to delivering the synchronization information to the sink:

* Delivery of the synchronization information before the start of the presentation
* Use of an additional synchronization channel
* Multiplexed data streams

If the multimedia presentation is live and multiple parties are involved, then none of the approaches above is suitable for delivering synchronization information to the sink(s) in a timely fashion. Figure 11.2 shows typical communication patterns.



Of particular interest here, is that if multiple sinks are involved, then they will receive identical data. It would be inefficient if the data were replicated at the source for separate transmission to each of the sinks. It would also be inefficient if the same operation was carried out at different sinks. *Multicasting* or broadcasting of streams is the responsibility of the stream layer, whereas efficient planning of operation execution in the different communication patterns is a responsibility of the object layer.

**Multi-Step Synchronization**

In a distributed environment, synchronization is typically a multi-step process, during which the synchronization must be maintained so as to enable the sink to perform the final synchronization. The synchronization steps are:

* during object acquisition, e.g., during frame digitization
* during retrieval, e.g., synchronized access to frames of a stored video
* during delivery of the LDUs to the network
* during the transport of the LDUs, e.g., using isochronous protocols
* at the sink, e.g., synchronized delivery to the output devices
* within the output device

With many different points at which synchronization must occur decisions must be made about how to implement it. A first decision is the selection of the type of transport for the synchronization specification. In runtime, decisions must be taken concerning the location of synchronization operations, keeping clocks in synchrony (if used to provide common timing information), and the handling of multicast and broadcast messages. Coherent planning of the steps in the synchronization process, together with the necessary operations of the objects, e.g., decompression, must also be done. In addition, presentation manipulation operations demand additional replanning at runtime.

**Synchronization Specification**

A synchronization specification should comprise:

* Intra-object synchronization specifications for the media objects of the presentation
* QoS descriptions for intra-object specifications
* Inter-object synchronization specifications for media objects of the presentation
* QoS descriptions for inter-object synchronization

In addition, the form, or alternate forms, of a multimedia object may be described. For example, a text could be presented as text on the screen or as a generated audio sequence. In the case of live synchronizations, the temporal relations are implicitly defined during capture. QoS requirements are specified before the start of the capture. In the case of synthetic synchronization, the specification must be created explicitly.

Orchestration

**Orchestration or Meta-Scheduling**

Each resource manager includes a scheduling function which orders the current requests for servicingso as to meet the required performance bounds. For example, a continuous media file system schedulesstorage system access operations, and the network layer schedules traffic to the transport layer. Anapplication requires the coordinated operation of these scheduling functions if end-to-end performance bounds are to be met. An approach to coordinating resource scheduling of the various systems is to adda layer between the application and the resource managers for orchestration or meta-scheduling.

QOS Architecture

Quality of service (QOS) is used in the OSI reference model to allow service users to communicatewith network service regarding data transmission requirements. In OSI, QOS is specified using anumber of parameters which can be grouped into three sets: single transmission, multiple transmission,and connection mode. QOS parameters include transit delay,Figure real-time multimedia system architecture: (a) the orchestration layer as a middle layer betweenthe application and multimedia system services, (b) the orchestration layer as a service for providingsynchronization for a distributed computing system, and (c) the orchestration function, with verticalarrows indicating control paths and horizontal arrows indicating data paths.**A**

FRAMEWORK FOR MULTIMEDIA SYSTEMS

The framework presented here provides an overall picture of the development of distributedmultimedia systems from which a system architecture can be developed. The framework highlights thedominant feature of multimedia systems: the integration of multimedia computing andcommunications, including traditional telecommunications and telephony functions.Low-cost multimedia technology is evolving to provide richer information processing andcommunications systems. These systems, though tightly interrelated, have distinct physical facilities,logical models, and functionality. Multimedia information systems extend the processing, storage, andretrieval capabilities of existing information systems by introducing new media data types, includingimage, audio, and video. These new data types offer perceptually richer and more accessiblerepresentations for many kinds of information. Multimedia communication systems extend existingPoint-to-Point connectivity by permitting synchronized multipoint group communications.Additionally, the communication media include time-dependent visual forms as well as computer