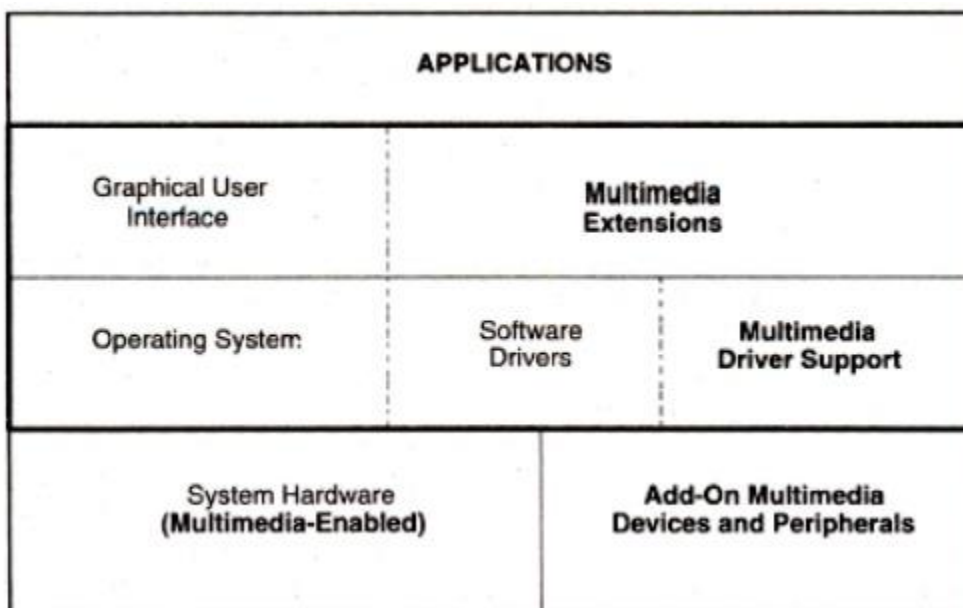


Multimedia Systems Architecture

Multimedia encompasses a large variety of technologies and integration of multiple architectures interacting in real time. All of these multimedia capabilities must integrate with the standard user interfaces such as Microsoft Windows.

The following figure describes the architecture of a multimedia workstation environment. In this diagram.



The right side shows the new architectural entities required for supporting multimedia applications.

For each special devices such as scanners, video cameras, VCRs and sound equipment-, a software device driver is need to provide the interface from an application to the device. The GUI require control extensions to support applications such as full motion video

High Resolution Graphics Display

The various graphics standards such as MCA, GGA and XGA have demonstrated the increasing demands for higher resolutions for GUIs.

Combined graphics and imaging applications require functionality at three levels. They are provided by three classes of single-monitor architecture.

(i) VGA mixing: In VGA mixing, the image acquisition memory serves as the display source memory, thereby fixing its position and size on screen:

(ii) VGA mixing with scaling: Use of scalar ICs allows sizing and positioning of images in pre-defined windows.

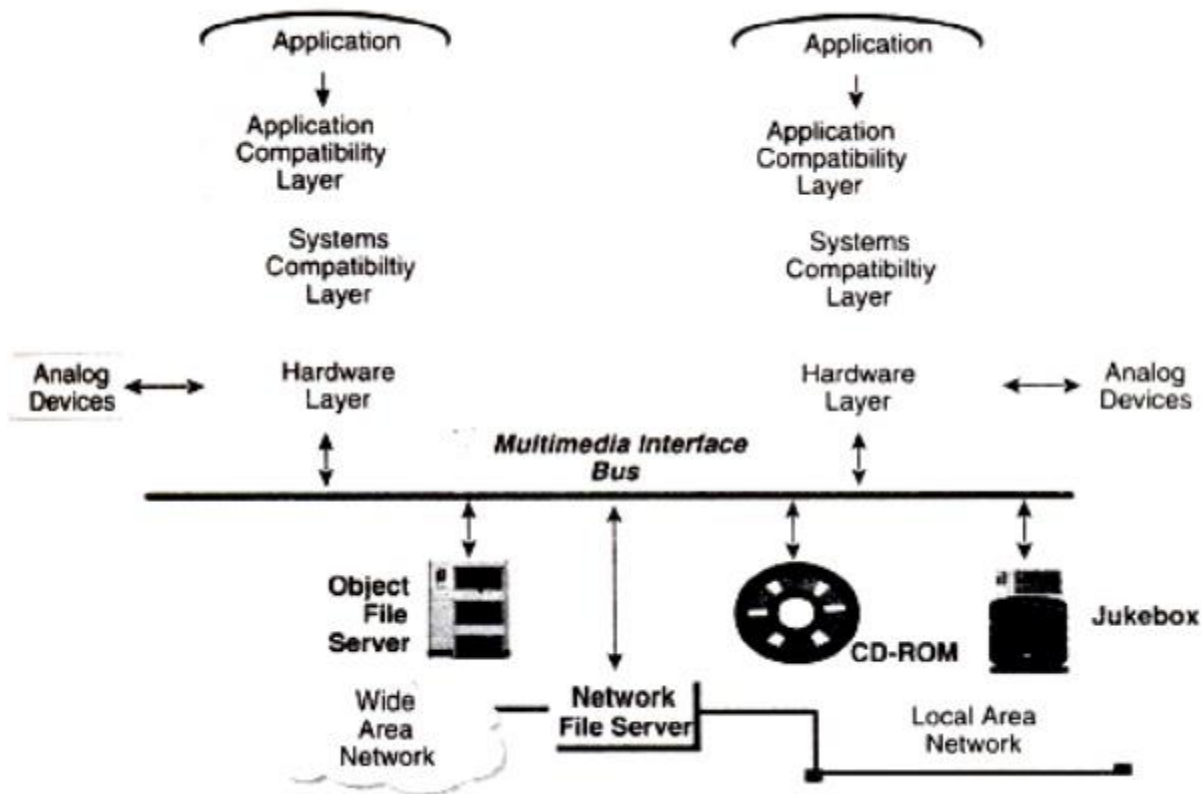
Resizing the window causes the things to be retrieved again.

(iii) Dual-buffered VGA/Mixing/Scaling: Double buffer schemes maintain the original images in a decompression buffer and the resized image in a display buffer.

The IMA Architectural Framework

The Interactive Multimedia Association has a task group to define the architectural framework for multimedia to provide interoperability. The task group has COncentrated on the desktops and the servers. Desktop focus is to define the interchange formats. This format allows multimedia objects to be displayed on any work station.

The architectural approach taken by IMA is based on defining interfaces to a multimedia interface bus. This bus would be the interface between systems and multimedia sources. It provides streaming I/O service"s, including filters and translators **Figure 3.4** describes the generalized architectural approach



Network Architecture for Multimedia Systems:

Multimedia systems need special networks. Because large volumes of images and video messages are being transmitted.

Asynchronous Transfer Mode technology (ATM) simplifies transfers across LANs and WANs.

Task based Multi level networking

Higher classes of service require more expensive components in the workstations as well as in the servers supporting the workstation applications.

Rather than impose this cost on all work stations, an alternate approach is to adjust the class of service to the specific requirement for the user. This approach is to adjust the class of services according to the type of data being handled at a time also.

We call this approach task-based multilevel networking.

High speed server to server Links

Duplication: It is the process of duplicating an object that the user can manipulate. There is no requirement for the duplicated object to remain synchronized with the source (or master) object. **Replication:** Replication is defined as the process of maintaining two or more copies of the same object in a network that periodically re-synchronize to provide the user faster and more reliable access to the data. Replication is a complex process.

Networking Standards: The two well-known networking standards are Ethernet and token ring. ATM and FDDI are the two technologies which we are going to discuss in detail.

ATM: ATM is a acronym for Asynchronous Transfer Mode. It's topology was originally designed for broadband applications in public networks.

ATM is a method of multiplexing and relaying (cell-switching) 53 byte cells. (48 bytes of user information and 5 bits of header information).

Cell Switching: It is a form of fast packet switching based on the use of cells. **Cells:** Short, fixed length packets are called cells.

ATM provides high capacity, low-latency switching fabric for data. It is independent of protocol and distances. ATM effectively manage a mix of data types, including

text data, voice, images and full motion video. ATM was proposed as a means of transmitting multimedia applications over asynchronous networks.

FDDI: FDDI is an acronym of Fiber Distributed Data Interface. This FDDI network is an excellent candidate to act as the hub in a network configuration, or as a backbone that interconnects different types of LANs.

FDDI presents a potential for standardization for high speed networks.

The ANSI standard for FDDI allows large-distance networking. It can be used as high-performance backbone networks to complement and extend current LANs.

DISTRIBUTED MULTIMEDIA SYSTEM

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 performs a number of tasks related to a specific business process. A business process consists of a 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 retrieves 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 control subapplication which allows a user to control playback or edit the component object.

Document store

A document store is necessary for application that requires storage of large 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 linked information, 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 or type, 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 providing 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 provides a directory of all multimedia objects on the server tracked by that element of the directory 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 ways.

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 of object classes, they transfer objects on demand on clients. They provide hierarchical storage for moving unused objects to optical disk libraries or optical tape libraries. They

provide system administration functions for backing up stored data. They provide the function of direct high-speed LAN and WAN server-to-server transport for copying multimedia objects.

Middleware in Distributed Workgroup Computing

The middleware is like interface between back-end database and front-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 model. 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 stored are known 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.

Synchronization in Multimedia Systems

Introduction

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,

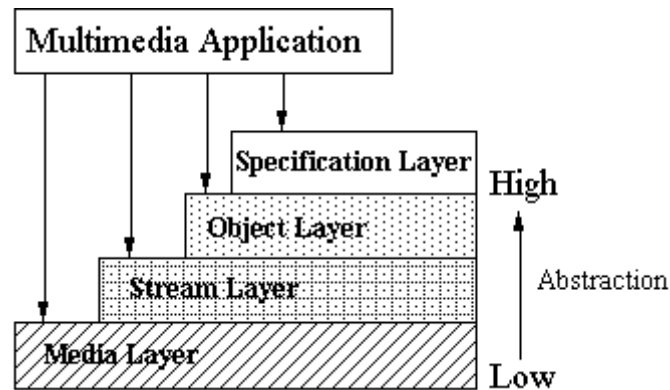
consequently synchronization support is typically found in the operating system, communication system, databases, multimedia documents, and the application. A general scheme might involve a layered approach to achieving synchronization. For example, a Computer-Supported Collaborative Workgroup (CSCW) 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).

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.

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.

QoS refers to the capability of a telecommunication system to provide better service to selected traffic over heterogeneous networks (technologies or domains). The primary goal of QoS is to provide priority, including dedicated bandwidth, controlled jitter and latency (required by some real-time and interactive traffic), and improved loss characteristics. Moreover, it is important to assure that providing priority for one or more flows does not cause the failure of other flows. On intuitive level, QoS represents a certain type of requirements to be guaranteed to the users (e.g., how fast data can be transferred, how much the receiver has to wait, how correct the received data is likely to be, how much data is likely to be lost, etc.).

QoS requirements for multimedia traffic have been covered by different standardization groups, like ITU, ETSI or 3GPP.

A **multimedia framework** is a software framework that handles media on a computer and through a network. A good multimedia framework offers an intuitive API and a modular architecture to easily add support for new audio, video and container formats and transmission protocols. It is meant to be used by applications such as media players and audio or video editors, but can also be used to build videoconferencing applications, media converters and other multimedia tools. Data is processed among modules automatically, it is unnecessary for app to pass buffers between connected modules one by one.

Multimedia Framework (MMF) Architecture

MMFramework is an open multimedia framework which may be used for dynamic creation of various multimedia applications and which could be extended by new multimedia devices. The proposed framework's architecture consists of six layers. Its definition results from decomposition of the system into components with well-defined interfaces and internal implementation dedicated to the given hardware usage or applied policy of the system control and management. Each layer consists of a collection of components which are characterized by similar functionality. The structure and goals of the layers are the following:

1. The first layer called **MMHardware and System Software Layer** consists of multimedia hardware and software provided by vendors. This layer is represented by a wide spectrum of devices such as: video cameras, computers, audio/video encoders/compressors, media servers, etc. These devices are usually equipped with proprietary control software.
2. The second layer - **MMHardware CORBA Server Layer** packs up the vendor-provided software by CORBA interfaces. This layer introduces a uniform abstraction defined by an interface. The main goal of introduction of this layer is to establish a common base for the system development.
3. The third layer - **A/V Streams Control Layer** is dedicated to multimedia streams creation, control, and destruction. This layer implements the [OMG specification](#) and provides software objects which expose functionality of the lower layer CORBA servers in standard form most suitable for audio and video streams control. It provides an abstraction of a stream encapsulated in the form of a CORBA object which represents its parameters and control operations. This layer provides also mechanisms for streams parameters negotiation between source and destination multimedia devices and provides streams addressing and QoS control.
4. The fourth layer - **Presentation Layer** resolves the problem of different data types used for the parameters of multimedia devices and streams representation. The main goal of this layer is to translate the parameters types from their actual values to CDF (Common Data Format).
5. The **Management and Access Control Layer** provides a uniform view of the MMF components' state and a set of functions for their manipulation and accessing (e.g involving security or providing statistics). Each component which is an object with its own interface and notification mechanism represents the state of a single connection or a device.

6. A top layer of the MMF architecture is called Application Layer. The entities of this layer are collection of user interfaces that provide access to control and visualisation of the system state in the most convenient(usually graphical) form. The objects defined on this level act as observers of the system components and combine them in the given application scenario. They may also perform the MMF clients' role actively changing the system's state by operations invocations on the devices and connections abstraction provided by the lower layer.
7. MMFramework has been constructed taking into account the distributed system scalability.