RSU Journal of Biology and Applied Sciences

ISSN:

RSUJBAS

About us

Rivers State University Journal of Biology and Applied Sciences (RSUJBAS) publication is a quarterly, open access, international journal for all academic research in science discipline. Microbiology, Botany, Zoology, Environmental Biology, Chemistry, Physics, Mathematics, Computer Science, Biochemistry, Medical Laboratory Sciences and other applied science related areas. RSUJBAS is a platform set for elites to influence, contribute and communicate to the global environment through their various academic researches. We synergistically engage our noble efforts to contribute to the knowledge development, discoveries and innovations in all fields of study. In RSUJBAS we publish research papers on current academic issues with standard scientific reviews. RSUJBAS publishes original research articles, review articles, case studies, short communications, survey report, comparative studies and many more.

Aim and scope

Rivers State University Journal of Biology and Applied Sciences aims to publish high quality papers that communicate fundamental and contemporary discoveries both theoretical and practical. Most importantly RSUJBAS seeks to establish a platform for communicating emerging trends in various discipline such as Microbiology, Botany, Zoology, Environmental Biology, Chemistry, Physics, Mathematics, Computer Science, Biochemistry, Medical Laboratory Sciences and other applied science related areas.

Description:

- Area of concentration: All science academic disciplines
- Frequency of publishing: Quarterly
- Mode of publishing: both online and print publication
- Language of publication: English
- Double Blinded Review Process
- Zero Level Plagiarism Tolerance

Why publish with us

Low Article Processing Charge (ACP) to promote the research work
Easy and Rapid review process
Instant publication upon acceptance
Dedicated editorial and review team for fast review process
RSUJBAS provides hard copies of publication every quarterly

Editorial Board

All Correspondence to
Dr. Sam Wenedu (Editor-in-Chief)
Department of Microbiology, Rivers State University
Editor.jbas@yahoo.com

Or

OLUCHI DICKSON
Publication Manager
Dicksonoluchi87@gmail.com

Consulting Editors

Guideline for Manuscripts

Manuscripts should be typewritten on an A4 sheet having B1.5= line spacing throughout the text. The margins should be 2B54cm (1 inch) in all sides and page number should be consecutively on the bottom of the page. The manuscript should be written in Times New Roman using '12' font size.

For original research paper, the manuscript should be arranged in the following order: Title page, Abstract, Keywords, Introduction, Materials and Methods, Results, Discussion, Acknowledgment, References, Tables with legends, figures with legends and supplementary materials

The title page should contain the title, the name(s) of the author(s), the name(s) and address(es) of the institution(s) where the work was carried out, including a valid e-mail address from the corresponding author along with telephone numbers. The title of the manuscript should be specific and concise but sufficiently informative.

The Abstract should not exceed 250 words and it should contain brief summary of the findings including brief introduction, methodology, results, and conclusions,

The keywords should have a minimum of five and maximum of seven words.

The introduction should provide a clear statement of the problem and indicates aim of the study citing relevant literature to support background statements.

The Materials and Methods should include the methods and methodology of the research.

The Results should be presented in the form of tables or figures. It should be presented with clarity and precision. Statements used to present results should be written in the past tense. Detailed interpretation of data should not be included in the results but should be put into the Discussion section.

The Discussion should interpret the results clearly and concisely, and should integrate the research findings of this and past studies on this topic. Highlight the significant/unique findings of the research under conclusion.

The acknowledgments of people, grants or funds should be brief.

Contents

List of Contributors

Kingsley-Opara, Ngozi
Research Scholar, Department of Computer Science, Ignatius Ajuru University of Education, Rivers State, Nigeria.
Advanced Database Management

Prof. Asagba, Prince Oghenekaro.
Visiting Scholar, Department of Computer Science, University of Port-Harcourt, Rivers State Nigeria.
Email:asagba.prince@uniport.edu.ng

Gabriel B.C., Gabriel M. N, P. O. Asagba
School of Graduate Studies Ignatius Ajuru University Of Education (IAUE), Rumuolumeni, Port Harcourt, Rivers State, Nigeria.
Department Of Computer Science.
gabrielbariyira@gmail.com, meegabz@gmail.com

WAIDOR, Tamaramiebi Keith[1]& ASAGBA, Prince Oghenekaro[2]
Department of Computer Science,
Faculty of Natural and Applied Sciences
Ignatius Ajuru University of Education, Port Harcourt
zalimaxxx@gmail.com

[2]Department of Computer Sciences,
University of Port Harcourt, Port Harcourt, Nigeria
Prince.asagba@uniport.edu.ng

Fiberesima, Alalibo Ralph
Visiting Scholar, Department of Computer Science,
University of Port-Harcourt, Rivers State Nigeria.
fiberesima.a.r@outlook.com;

Asagba, Prince Oghenekaro
Visiting Scholar, Department of Computer Science,
University of Port-Harcourt, Rivers State Nigeria.
asagba.prince@uniport.edu.ng

Kingsley-Opara, Ngozi
Research Scholar, Department of Computer Science,
Ignatius Ajuru University of Education, Rivers State, Nigeria.
Email: ngozikopara@gmail.com

Prof. Asagba, Prince Oghenekaro.
Visiting Scholar, Department of Computer Science,
University of Port-Harcourt, Rivers State Nigeria.

Requirements Analysis: Multimedia Data and Databases

Fiberesima, Alalibo Ralph
Visiting Scholar, Department of Computer Science, University of Port-Harcourt, Rivers State Nigeria.
fiberesima.a.r@outlook.com;

Asagba, Prince Oghenekaro.
Visiting Scholar, Department of Computer Science, University of Port-Harcourt, Rivers State Nigeria.
Email:asagba.prince@uniport.edu.ng

Abstract
Multimedia database system technologies provide an overview of architectures for multimedia database management. In this report we discuss different dimensions in the following aspects: firstly, we give an explanation of requirements engineering and analyses in the context of software engineering; this covers such topics as requirements gathering testing and SOLID principles. Next, there is an introduction to requirements analysis for multimedia data management. Finally, we discuss multimedia database management architectures, models, and functions, which include query processing, metadata management, and storage management.
There is too much to write about multimedia data management and an effort has been made to provide as much information as possible. This is because, in the fields of e-commerce, news, television programs, conversation and social media, multimedia databases exist and the amount of information they store can only increase.

We discuss various multimedia database management technologies, which include architectures, data modeling, query processing, metadata management, storage management, and data distribution. Essentially, we examine the functions of database systems and discuss the impact of managing multimedia data. Recent progress in multimedia database systems management, and associated commercial database system products are also discussed.

Keywords: *Multimedia data management. Multimedia data security. Multimedia Data storage.*
Multimedia database architecture. Requirements engineering, Requirements analysis, Secure distributed multimedia data management

Introduction
Multimedia databases are a very popular topic of research. Their use in multimedia, geographical information systems, and digital libraries demand special requirements with respect to their database schema and particular instances (Ramakrishnan & Gehrke, 2000) as they may need schema different from traditional columns and rows. Furthermore, multimedia databases rapidly accumulate data in their respective domains of use and the challenge of managing the data is the focal point of multimedia database research especially when it concerns multimedia information collation and retrieval.

Requirements Analysis in Software engineering has as its primary purpose and design the service of the user. This involves the combination of practice and theory. These incorporate quantitative thinking, an evaluation of alternatives in the compilation of requirements for any software project and management of risk for the project.

The need to understand the customer who may be the primary user by a practitioner of software engineering lays an emphasis on domain analysis, requirements gathering and

requirements validation, use case analysis and usability is the priority of the software engineer.
Software engineering is based on the development on SOLID principles and reusable technology and encompasses the following features:
Single-responsibility principle, whereby a class should only have a single responsibility, that is, only changes to one part of the software's specification should be able to affect the specification of the class.

Open–closed principle whereby software entities should not be modified, but may be extended.
Liskov substitution principle whereby objects can should be replaced with those of sub classes without affecting the correctness of that program.
Interface segregation principle whereby specific interfaces are to be preferred to a single all-purpose interface.

Dependency inversion principle where abstractions are to be preferred over concretions
Finally, it is necessary for emphasis to be placed on an understanding of design principles in conjunction with a thorough grasp of suitable technology for a successful project. Visual models are one method of software engineering, and many modelling tools and diagrams are available, one popular variant of which is UML.

Software metrics are an integral part of project implementation and should be incorporated as a means for practitioners to think quantitatively, while emphasis also should be made on the use of iterative and agile development using requirements analysis, design and implementation.
Requirements engineering also involves software design, quality assurance, and iterative changes especially with advanced features, and may inculcate test-first development, and effective use of templates for each document specification. Risk management is important in all software engineering activities. These include cost evaluation, and risk evaluation, avoidance of unnecessary or unworthy work, and the evolution of plans with more access to information.
The use of techniques such as interviewing, brainstorming, use case analysis and prototyping for effective gathering result in a document which:

ensures the anticipated benefit of each requirement outweighs its cost, each featured requirement contributes to the solution of the problem, each requirement is unambiguous, consistent with its complementary requirements, contributes to a system, is affordable in line with the resources available, is verifiable, uniquely identifiable, does not represent too much of a constraint on the design of the system, ensures the proper organization of the requirements document, gives a clear rationale for requirements which need careful analysis or may prove to be controversial and obtains an agreement from all stakeholders for the requirements document.

Requirements engineering also includes the principles of testing and involves the following:
Firstly, testing should be treated like detective work, be effective and efficient with suspicion towards every new feature, be divided into equivalence classes based on inputs, and should be based on whether inputs cause variance in the resultant code for execution.
Secondly, tests done in each equivalence class together with boundary classes should meet standards for unit and integration testing.

Finally, testing should be for all categories of defects: algorithmic defects, timing and co-ordination defects, defects in handling unusual situations, and defects in documentation

The relationship between data, information and knowledge

In knowledge management the relationship between knowledge, information and data is important, and often misunderstood. This misunderstanding often leads to problems in information system design. Knowledge should neither be confused with data nor information, though it bears a relationship to both, while the differences between these terms are often a matter of degree.

The confusion about the meaning of data, information, and knowledge: how they differ, has resulted in enormous expenditure on technological initiatives with the resultant inability to deliver on the expenditure for an effective Return on Investment (ROI).

Elimination of the present confusion occurs if a better understanding of the ways in which information and knowledge are both similar and different is had, leading to a resultant costly and dangerous mistake in database software and computing. The widespread but largely unconscious assumption that information is equal to knowledge is wrong.

The idea that knowledge is more than information is intuitive and has led many authors to make distinctions between raw data, information and knowledge. At first, these concepts look almost obvious to common sense, and yet they have been a constant source of confusion.

Data can be described as symbols which have not yet been interpreted, information is the application of logic to data with meaning, and knowledge is what enables people assign meaning and thereby generate information. Another description allows data to be simple observations of states of the world, information is data endowed with relevance and purpose, while knowledge is valuable information. Yet another description holds information as meaningless symbolic statements, which become meaningful knowledge when interpreted. Finally, information consists of facts or data that is organized to describe a particular situation or condition whereas knowledge consists of truths and beliefs, perspectives and concepts, judgments and expectations, methodologies and know-how based on the available information.

All these underlying models of knowledge as a “higher form of information” hold the general idea that information has to be extracted from its raw materials, with meaning added to them.

Multimedia Data Formats

Multimedia data should be stored and transmitted in compressed form because of the large number of bytes required to represent it. For image data, the most widely used format is JPEG, named after the standards body that created it, the Joint Picture Experts Group. Video data may also be stored by encoding each frame of video in JPEG format, but such an encoding is wasteful, since successive frames of a video are often nearly the same.

The MPEG series of standards for encoding video and audio data developed by the Moving Picture Experts Group has developed these encodings to exploit commonalities among a sequence of frames to achieve a greater degree of compression. The MPEG-1 standard stores a minute of 30-frame-per-second video and audio in approximately 12.5 megabytes (compared to approximately 75 megabytes for video in only JPEG). However, loss of quality is to be expected for MPEG-1 encoding techniques, introduces some loss of video quality, resulting in output of a level roughly comparable to that of VHS videotape.

The MPEG-2 standard, designed for digital broadcast systems and digital video disks (DVDs); results in a negligible loss of video quality. MPEG-2 encoding techniques compress 1 minute of video and audio to approximately 17 megabytes. MPEG-4 provides techniques for further compression of video, with variable bandwidth to support delivery of video data over networks with a wide range of bandwidths. Several competing standards are used for audio encoding, including MP3, which stands for MPEG-1 Layer 3, RealAudio, Windows Media Audio, and other formats.

Continuous-Media Data

Video and audio data are the most important types of continuous-media data. Continuous-media systems are characterized by their real-time information-delivery requirements:
The speed of data delivery must be sufficient so that no gaps in the audio or video result.
The rate of data delivery must be such that there is no overflow of system buffers.
There must be synchronization among distinct data streams, and example being when the video of a person singing is not synchronous with the audio of the song being sung.

To supply data predictably at the right time to a large number of consumers of the data, the fetching of data from disk must be carefully coordinated. This involves data being fetched in periodic cycles, say of n seconds, n seconds worth of data being fetched for each consumer and stored in memory buffers, while the data fetched in the previous cycle is being sent to the consumers from the memory buffers. The cycle period is a compromise whereby short period sutilize less memory but requires more disk-arm movement, leading to a waste of resources, while a long period reduces disk-arm movement but result in more memory utilization and may result in a delay in data delivery. On the arrival of a new request, admission control comes into play: That is, the system checks if the request can be satisfied with the available resources (in each period); if so, it is admitted; otherwise, it is rejected.

Several vendors offer video-on-demand servers with current systems being based on file systems, because existing database systems do not provide the real-time response that these applications need. The basic architecture of a video-on-demand system comprises:
Video server: Multimedia data are stored on several disks (usually in a RAID configuration). Systems containing a large volume of data may use tertiary storage for less frequently accessed data.
Terminals: People view multimedia data through various devices, collectively referred to as terminals. Examples are personal computers and televisions attached to a small, inexpensive computer called a set-top box.
Network: Transmission of multimedia data from a server to multiple terminals requires a high-capacity network.

Video-on-demand service over cable networks is available in many places today, and has become ubiquitous, because of their usage in cable and broadcast television today.
Similarity-Based Retrieval
In many multimedia applications, data are described only approximately in the database.
The notion of similarity in database systems is often subjective and user specific. However, similarity testing is often more successful than speech or handwriting recognition, because the input can be compared to data already in the system and, thus, the set of choices available to the system is limited.
Several algorithms exist for finding the best matches to a given input by similarity testing. Some systems, including a dial-by-name, voice-activated telephone system have been deployed commercially.

Examples of similarity-based retrieval include:
Pictorial data: Two pictures or images that are slightly different as represented in the database may be considered the same by a user. For instance, a database may store trademark designs. When a new trademark is to be registered, the system may need first to identify all similar trademarks that were registered previously.

Audio data: Speech-based user interfaces are being developed that allow the user to give a command or identify a data item by speaking. The input from the user must then be tested for similarity to those commands or data items stored in the system.
Handwritten data: Handwritten input can be used to identify a handwritten data item or command stored in the database. Here again, similarity testing is required.

Multimedia systems have a variety of information sources – including text, voice, image, audio, animation, and video – which are delivered synchronously or asynchronously through multiple devices. The important characteristic of such systems is the combination of different media into a single unit one single unit, all controlled by a computer. Normally, multimedia systems require the management and delivery of extremely large bodies of data at very high rates using high speed Content Delivery Networks (CDN) and may require the delivery within real-time constraints. A major constraint exists for every multimedia system which is how to synchronize the various data types from distributed data sources for multimedia presentations. Moreover, considerable semantic heterogeneity may exist among the users of multimedia data due to the differences in their perceived interpretation or intended use of the data as the data is often spread across geographical regions or time zones and as such must not be stale when received. Semantic heterogeneity has been a difficult problem in conventional databases.

Conventional relational models lack facilities for the management of spatio-temporal tables and schema and do not cover all features required for multimedia database retrieval. The limitations of the relational model are also obvious when semantic modeling of time-dependent multimedia data (video or audio) is considered. From an integration point of view, this means that conventional data modeling techniques lack the ability to manage the composition of multimedia objects in a heterogeneous multimedia database environment. This is because, for a large number of multimedia applications, it may be required to integrate the various types of data semantically. In traditional database management systems (DBMSs), such as relational database systems, only textual and numerical data is stored and managed in the database and there is no need to consider the synchronicity among media. Retrieving data, based on simple comparisons of text or numerical values, is therefore no longer adequate for the multimedia data. The relational data model also has the drawback of losing semantics, which can cause erroneous interpretation of multimedia data. Finally, the relational data model has limited capabilities in modeling the structural and behavioral properties of real-world objects since the complex objects are not directly modeled and the behavioral properties of the objects are not explicitly specified in terms of meaningful operations and applicable knowledge rules.

Since a BLOB is treated as a single entity in its entirety, although some relational DBMSs have started the support for the access to multimedia objects in the forms of pointers to binary

large objects (BLOBs), they are incapable of interactively accessing various portions of objects.
The definition of a composite object which is an object consisting of other objects provides the capability to handle the structural complexity of the data it contains. In object-oriented database systems, the object-oriented data models offer a number of powerful features such as inheritance, polymorphism, encapsulation and abstraction mechanisms, in addition to the inclusion of image data formats. The object-oriented data models have been proposed as a data model that provides a system with better facilities for managing the multimedia data. This is because the object-orientation encapsulates data with a set of operations that are applicable to the data removing the need to worry about the heterogeneity of operations caused by different types of data for the purpose of manipulating data. However, the core of the object-oriented data models lacks facilities for the management of spatio-temporal relations though it provides a system with operational transparency and enables the definition of the part-of-relationship among objects which may often take on an arbitrary structure. In other words, the object-relational DBMS still is not designed to support multimedia information. Therefore, multimedia extension is needed to handle the mismatch between multimedia data and the conventional object-oriented database management systems. The purpose of the design and development of a multimedia database management system (MDBMS) is to efficiently organize, store, manage, and retrieve multimedia information from the underlying multimedia databases. Due to the heterogeneous nature of multimedia data and their characteristics, it is very unlikely that the design of an MDBMS can follow the footsteps of the design of a traditional DBMS since the MDBMS must have considerably more capabilities than conventional information-management systems.

Some of the important characteristics of multimedia objects are listed as follows

1. Multimedia objects are complex and therefore less than completely captured in an MDBMS.
2. Multimedia objects are audiovisual in nature and are amenable to multiple interpretations.
3. Multimedia objects are context sensitive.
4. Queries looking for multimedia objects are likely to use subjective descriptors that are at best fuzzy in their interpretation.
5. Multimedia objects may be included in fuzzy classes.

Hence, it is suggested that MDBMSs be developed as libraries from which only the minimum required functionalities can be compiled to meet the needs of an application.
Furthermore, in the authors define ten requirements for modeling multimedia data. These ten requirements are:

1. Incomplete information should be specified.
2. If an individual document definition extends beyond its type/class, it should be specified.
3. Integration of data from various databases should be uniformly done.
4. Information should be structured and described appropriately.
5. Internal modeling and external objects should be distinguished when presented.
6. There is the need for data sharing to be enabled among multiple documents.
7. There should be version controlling across databases for document uniformity.
8. Operations allowable by the database should be specified.
9. There is a need to handle concurrent access control operations like transactions and locks.

10. Context-free and context-sensitive references should be handled in a semantic manner.

The multimedia environment ideally consists of a multimedia presentation system and a multimedia database system and so an ideal multimedia environment not only displays media streams to users but allows two-way communication between users and the multimedia system. The absence of a database system and only a presentation system results in a system like a VCR or a TV as there is no feedback from the user. This is because users cannot specify queries for information from the system. The information may be relative to text data as well as image or video content. The combination of a multimedia presentation and multimedia database system allows users specify queries for useful information they wish to receive.

The architectures being examined to design and develop a multimedia database management system (MM-DBMS)fall under different categories, and various types.

One architecture type involves integrating multimedia data with the database system. This system has two approaches. The first one, known as the loose coupling approach, is where the multimedia data is managed by the file system, while the database system manages the metadata. In the other, also known as the tight coupling approach, the multimedia data is managed by the database system. In the loose coupling approach, the DBMS is used to manage only the metadata, and a multimedia file manager is used to manage the multimedia data. Then there is a module for integrating the DBMS and the multimedia file manager. The advantage of the loose coupling approach is that one can use various multimedia file systems to manage the multimedia data.

Another type of architecture is schema architecture. Here, the three-schema and n-tier architectures apply for a multimedia database system.
A third type of architecture is functional architecture, which describes the functions of a multimedia database system.

A fourth type of architecture is whether a multimedia database system extends a traditional database system. This is what we call system architecture.
A fifth type of architecture is a distributed architecture, where a multimedia database is distributed.

Finally, multimedia databases may be heterogeneous in nature and need to be integrated. The architecture for integrating heterogeneous databases is known as interoperable architecture.
Functional Architectures

Functions of a multimedia database system include data representation, browsing and editing, quality of service processing, real-time scheduling, metadata management, storage management, distribution, query/update processing and security/integrity management.
A more detailed description of functional architecture has major modules as follows: the presentation layer which presents various media types to the user, the query manager which performs query processing, the storage manager which accesses the multimedia database and the metadata manager which manages the metadata.

In the early to mid-1990s, there was much debate as to which models would be appropriate for multimedia databases. The competing models were object and relational models. After much discussion, there is the feeling that both objects as well as relationships are needed to capture the complex data types and the relationships between them. In addition, rule-based models are useful to specify the timing constraints between objects. Therefore, a combination of objects, relations, and rules is needed.

Multimedia data has several features which have to be supported. First of all, there has to be some way to capture the complex data types and all the relationships between the data. Various temporal constructs such as play-before, play-after, and play-together must also be captured.

Furthermore. an appropriate data model is critical for representing a multimedia database. Relational, object-oriented, and object-relational data models have been examined to represent multimedia data. In an object-oriented data model, each object corresponds to an object in the object model and the attributes of each object may be represented as object variables which should include time interval, frames, and content description.

There should also be a means of creating extensions to link to existing data models to support complex relationships for multimedia data. These extensions/relationships may include temporal relationships between objects.

As mentioned earlier, object models are better than relational models for multimedia databases because object-oriented models can represent complex data types. However, both types of models have to be extended to capture the temporal constructs and other special features. Both data models are associated with a query language. The language should support the constructs needed to manipulate the multimedia database if constructed using either a relational or object-relational model. For example, one may need a query to play a range of frames of a video script. Languages such as SQL, while being standard, are being extended for MM-DBMSs.

Metadata means many things to many people, within our context it means a data dictionary or schema. Initially, a three-schema architecture was proposed for centralized databases. Then, for distributed databases, the schema architecture was extended to five layers or an n-tier. Then came the Internet and the Web, and metadata has expanded to include information about usage patterns, policies, and procedures as well as administration information.

For multimedia databases, metadata plays a key role. Metadata may include information about the multimedia databases, such as text, image, audio, video data and annotations for text, images, audio, and video.

Metadata may also include combinations of data types such as audio and video or text and video. Metadata could also be characterized by its semantic meaning. An example, in the case of text metadata, being content-dependent metadata including information that depends on the content, i.e., keywords. Content-independent metadata may include information such as, “This is a fruit tree”. Metadata may be extracted or independent of multimedia data but ideally should have the same semantics with multimedia data.

Metadata for text should include information about the text. Content-independent metadata for text should include the type of text, the number of pages of text, formatting, chapters and paragraphs. Content-dependent metadata may include meaning and keywords.

SGML (standard generalized markup language), which became a standard in the 1980s has contributed a lot to development for text data.

SGML allows for textual data to be tagged for easy extraction of metadata. Also possible is the extraction of keywords from the tags and the development of appropriate tags is essential to this process. SGML has a subset called extensible markup language or XML With the advent of the World Wide Web, XML heavily utilized inmodeling metadata is an important part of the world wide web. XML tags can be placed in a relational database, or an object model to represent the metadata. Special XML based database models may even be developed for information retrieval since those systems may be utilized for text data and the text linked to multimedia data in turn. . For example, metadata for text may be placed in relational databases, and then those relational databases can be mined.

Text may also be annotated; large volumes of text may be annotated so that only essential information is included in the annotations. Especially with Web-based systems, one can follow the links and get the annotations for the particular text being reviewed. Annotations can be compared to footnotes. Annotations can also be regarded as a type of metadata.

Metadata for Multimedia

Metadata for images may include text data describing the images, or the metadata may be stored in relational databases describing various properties of the images.

For example, consider a picture illustrating an ocean with palm trees and beach homes. The text metadata for this picture may be, “the picture A describes a farm in a forest with palm trees.”

As in the case of text, metadata for images may be described as annotations.

For example, when one browses the Web, annotations about the images may be available when one clicks on the images. That is, images may be used to describe the metadata. Images may be tagged and subsequently metadata extracted from the tags. The extracted metadata may be analyzed to understand the images. This is a development known as Image mining.

Image metadata may also be content-independent or content-dependent. An example of content-dependent image metadata is, “the image is empty”. Content-independent metadata may include information such as resolution and the number of pixels.

Audio data is essentially speech data. While the speech itself is audio data and is stored in audio databases, the text that describes this audio such as, “Audio data frames 1 to 200 contain speech on audio data” is the metadata.

Audio metadata may not only be text but may also include audio clips, for example, samples of speech. In that case, the samples of speech are the metadata.

Video data, like audio data, is continuous media, unlike text and images which are still or non-continuous data. With respect to metadata, video and audio data are handled in a similar manner.

Video metadata may not just be text but can be images as well as video frames. Images may be extracted from video clips and stored as part of the metadata and later used for video retrieval. An example would be, “find the portion of the video where P is throwing a ball toQ.” The metadata may have images of P and Q. Parts of the video can also be used as the metadata.

Typically, the modules for managing metadata include extracting, querying, updating, indexing, and maintaining the security and integrity of the metadata. Metadata should interface the database system as well the user. In fact, metadata should be usable by the database system user.

In addition, there may be complex metadata, and therefore query, update, and indexing strategies may be more complex in multimedia systems.

Multimedia Query Processing
All multimedia DBMS must support the basic DBMS functions. These include data representation, data manipulation (which includes query processing, editing, browsing, filtering, transaction management, and update processing), metadata management, storage management, and maintaining security and integrity.

All of the database functions for an MM-DBMS are more complex because the data may either be structured or unstructured for multimedia databases. Furthermore, handling various data types such as audio and video is complex. An MM-DBMS must also support real-time processing to synchronize multimedia data types such as audio and video. Quality of service is an important aspect for an MM-DBMS. Special user interfaces and multi-modal interfaces are also needed to support different media.

Object Editing
In a multimedia DBMS, two objects may be merged to form a third object. One object can project an object to form a smaller object and objects may be merged based on time intervals, while an object may also be projected based on time intervals. Objects may also be updated in whole or in part. Browsing multimedia data is essentially carried out by a hypermedia database management system.
Filtering is the process of removing unnecessary material from data. This occurs quite often in video data where inappropriate material is removed from a video clip. This means that the filtered data is displayed.

Transaction management also is possible in an MM-DBMS, with the associated features of transaction management, concurrency control and recovery. Here we specify the transaction models, the presence of any special concurrency control and recovery mechanisms?
Update processing is usually considered part of query processing or transaction management. Update processing is essentially updating the multimedia data and is often a single user update.
Query operation is usually the most commonly used function in a database system.

First, It should be possible for users to query the database and obtain answers to their queries. There are several aspects to query processing. First, a good query language is needed. Languages such as SQL (structured query language) are popular for relational databases and are being extended for other types of databases.

The second aspect is the availability of techniques for query processing. Numerous algorithms have been proposed for query processing in general. Also, different strategies are possible to execute a particular query, but when the costs for the various strategies are computed, and the one with the least cost is usually selected for processing, this process is called query optimization.
In general, rules used in the transformation process include the factoring of common sub-expressions and pushing selections and projections down in the query tree as much as possible. Cost is generally determined by disk access. The goal is to minimize disk access in processing a query. When users pose a query using a language, the constructs of the language have to be transformed into the constructs understood by the database system. This process is called query transformation. Query transformation is carried out in stages based on the various schemas. For example, a query based on external schema is transformed into a query on conceptual schema. This is then transformed into a query on physical schema. If selections

and projections are performed before the joins, then the cost of the joins can be considerably reduced.

The modules in query processing include the user interface manager which accepts queries, parses the queries, and then gives them to the query transformer. The query transformer and query optimizer communicate with each other to produce an execution strategy. The database is accessed through the storage manager. The response manager gives responses to the user. All of the modules for query processing are needed for multimedia databases. That is, the query transformer will take the query request and transform it into requests that can be understood by the system. The query optimizer will then optimize the query.

The query may be "find all the objects that have a common map of America in their content and display them with all objects with a map of Africa." Then, with the object relational representation, one could examine the contents of the objects and join those objects where their content has a map of Africa.

In a query language for multimedia database systems, the user poses the query in a language that is appropriate for an MM-DBMS. The user interface manager then performs various functions and gives the query to the query processor as responses. The assembled responses are then presented to the user. The complexity of multimedia data means there is the need for interfaces for the different media types. For example, the interface for text may be quite different than the interface for video. Nevertheless, a common language to express queries is highly desirable.
Since SQL is a standard query language for databases, various extensions to SQL are being proposed for multimedia databases also. A query language should support all these constructs.

SQL for Multimedia Queries
Various languages to manipulate relations have been proposed. Notable among these languages is the ANSI (American National Standards Institute) standard SQL. This language is used to access and manipulate data in relational databases. SQL is widely accepted among database management system vendors and users. It supports schema definition, retrieval, data manipulation, schema manipulation, transaction management, integrity, and security. Other languages include the relational calculus first proposed in the Ingres project at University of California at Berkeley though recent vendors now incorporate this in their MM-DBMS products making it indistinguishable from SQL in their offerings.

SQL is a data manipulation and data definition language and is used to define the schemas and tables as well as manipulate the data for retrieval from the MM-DBMS. The operations for data definition include create, delete, and insert. The operations for data manipulation include statements of the form, "select x from table where some condition for x is specified." This select statement is used to retrieve data using just one relation or multiple relations.
With multimedia data, one may use SQL for selecting elements from an object-relation. For example, to retrieve the content portion of the multimedia object described in the previous section, an SQL statement would look as follows:

Select X from Table where Content = ???
Here, X is the data, table is the relation name and content is the string ???. However, with image and video data, we need to extend the language to support image and video content. That is, we need a way to specify the image, such as the U.S. map, or video, such as a

particular film script. One possibility is to express the content with a string and then store the information that is the content in the metadata. That is, the string is the name of the frame, while the actual content may be the film. But there has to be a way to differentiate ??? and its content. Therefore, we need additional constructs, such as content in XXX. That is, the query may be of the following form:

Select from Table
WHERE CONTENT = IN ???
We also need constructs to play-together, play-after and play-before. That is, to play video and audio data together, we may need a statement of the following form:
PLAY A AND B
WHERE A =
SELECT - - -
AND B =
SELECT - - -

That is, we need to play A and B together. To get A and B, we need select statements. The above are just examples. A lot of work is going on to not only extend SQL but also to develop new multimedia query languages.
Note that SQL/MM contains the extensions that the standards groups have proposed for multimedia databases.
User Interface Issues

In a MM-DBMS, a good user interface is needed to process the queries. These queries could be expressed via SQL, forms, pictures, and graphics. An ideal user interface should support multi-modal interfaces. Of recent research into user interfaces and database management is proceeding almost independently and visualization tools have been integrated with database management systems. For multi-media database management, a variety of interfaces should include interfaces for communicating with video, audio, and text databases. In addition, interfaces to support SQL extensions for multimedia data as well as ODMG standards should be provided.
Multimedia Storage Management

Efficient storage management techniques for retrieving data are needed in a MM-DBMS so that users can get their responses in a timely manner. This is because as a user interacts with the multimedia system through queries, it is critical that the queries are processed efficiently.
In an example, a single video clip may consume gigabytes of data. This means that special techniques are needed for the efficient access of multimedia data as standard techniques for relational database management are not sufficient for multimedia databases.
Access methods and indexing are also closely related to storage management. This is because efficient access may need the data to be indexed.

Access Methods and Indexing
By using an appropriate indexing mechanism, the query processing algorithms may not have to search the entire database. Instead, the data to be retrieved could be accessed directly. Consequently, the retrieval algorithms are more efficient. The storage manager is responsible for accessing the database. To improve the efficiency of query and update algorithms, appropriate access methods and index strategies have to be enforced. That is, in generating strategies for executing query and update requests, the access methods and index strategies

that is used need to be taken into consideration. The access methods used to access the database depend on the indexing methods. Therefore, creating and maintaining appropriate index files is a major issue in database management systems.

Developing special index methods and access strategies for multimedia data types are the major issues in storage management for multimedia databases because of the following:
Content-based data access is important for many multimedia applications.
Efficient storage is tied to the development of indexes for the annotations. Since the annotations describe information about the multimedia data, if one can access the annotations, one can get the multimedia data.

Storage Methods
In MM-DBMS systems there are two main storage methods, the single method and the multiple method. In the single disk storage method, all of the multimedia data are stored on one disk, that is, the multimedia database has text, image, video, and audio data.
In the single storage method, the data may be stored contiguously (side-by-side) to save space, or it may be scattered randomly. In the multiple disk storage method, objects are distributed across disks. Text data, image data, video data and audio data are stored on separate disks. In another type of multiple disk storage method, the multimedia data are distributed. Another form of multiple storage is disk striping, where the multimedia objects are stored on more than one disk. That is, objects of different type formats are distributed across multiple disks.
For each of the techniques for disk storage, efficient access methods and index strategies are necessary to efficiently access the data. For the case of the single disk strategy, all of the objects are on one disk. This means that the system does not need to access multiple disks to answer a query, but the system needs to locate each object within the disk for display. If audio data is stored on one disk and video data on another, scheduling techniques are needed for synchronized display of the data. In the case of disk striping, for instance on redundant array of inexpensive disk (RAID) systems, the system needs to scan multiple disks to compose the object.

Materials and methods
The present study was carried out to know the theory and practice regarding the use of multimedia databases. The method of Literature Review (LR) is used as a tool to construct a base of further research questions in multimedia databases which are formulated from the identification of factors revealed via analysis of the literature in the aforementioned topics

Results
The results of the research can be summarized in a table listing the requirements for multimedia databases.

Characteristic	Yes/No
Rows and columns	No
Additional schema	Yes
Support for only SQL	No
Support for SQL with extensions	Yes
Open closed principle	Yes
Single Responsibility Principle	Yes
Very Large Data	Yes
Liskov Substitution Principle	Yes
Interface Segregation Principle	Yes

Dependency Inversion Principle	Yes
Inputs cause variance	Yes
Heterogeneous multimedia	Yes
Similarity based retrieval	Yes
spatio-temporal tables and schema	Yes
Support for BLOB	Yes
Feedback mechanism for interaction	Yes
Multi architectural design	Yes
Metadata support	Yes

Table 1: Requirements analysis for multimedia databases

Discussion

The various types of architectures for multimedia database systems include loose/tight integration architecture, schema architecture, system architecture, functional architecture, distributed architecture, interoperability architecture, and hypermedia architecture. These architectures for database systems have an impact on multimedia data management on these architectures. We have also examined various aspects including architectures for integrated heterogeneous databases, architectures based on object request brokers, client-server-based architectures, three-tier architectures, and component-based architectures. There is the understanding that component-based architectures will gain popularity as more and more applications deal with complex data such as voice, text, video, and images. There is no ideal architecture for a multimedia database system. The architecture selected will depend on the user needs and the application in question. Therefore, before building a multimedia database system, it is important for one to carry out an architecture study and select the most appropriate architecture.

Data and information modeling for multimedia databases and applications is also critical. An overview of data models includes object-oriented data models, object-relational data models, and hypersemantic data models. Information modeling is also critical as good data model is critical for multimedia applications. The data model should not only capture the complex data types but also the temporal relationships between the data objects.

Metadata plays a key role in multimedia databases. Metadata may include information about a multimedia database, including text, image, audio, and video data. This information could be of the form, “Frames 2000 to 3000 contain information about the professor’s speech.” The data here is the president’s speech. Metadata may also contain annotations for text, images, audio, and video.

Metadata exists for text, images, audio, and video. Both content-dependent as well as content-independent metadata exist. We also discussed annotations for multimedia data, and how to extract metadata from the data for text, video, audio, and images.

Five disparate topics pertaining to metadata exist. One dealt with metadata for combinations of data types, the second with ontologies, the third with annotations, the fourth with pedigree, and the fifth with XML for multimedia data. As mentioned earlier, XML is an increasingly important development for specifying multimedia documents. Also discussed are issues regarding metadata management, specifically, issues on querying the metadata, carrying out transactions on metadata, and distributing the metadata. Often, for multimedia databases, the metadata can be quite large, therefore, efficient techniques are needed for managing the metadata.

Multimedia query processing. First, data manipulation issues, in particular, browsing, editing, transaction processing, querying, and updating have been discussed. We then discussed query processing at some length.
Examples were given of performing joins in multimedia databases, and we then discussed query language issues for multimedia database systems.
Much of the work on multimedia database systems has focused on query processing and data modeling, the need for a good data model and query language as well as strategies for query processing.

Finally, we describe access methods, indexing, and storage strategies for multimedia databases. As mentioned earlier, a lot of work remains to be done for efficient query processing, and some key issues addressed. We began with a discussion of indexing for multimedia databases and explain how we can index data with keywords, or we can use multimedia data such as images for indexing. Next, we discussed caching as well as synchronizing the output/display
of multimedia data. We discussed storage strategies for multimedia data, i.e., single disk storage as well as multiple disk storage.
Multimedia databases consume a lot of space. Therefore, access techniques developed for relational and even object database systems are not sufficient for multimedia database systems. Special techniques are needed not only for text, images, video, and audio data, but also for combinations of data types.

References

Abate, A. F., Nappi, M., & Sebillo, M. (2004). *Proceedings of the workshop on mdic 2004 : multimedia databases and image communication : Salerno, Italy, 22 June, 2004.* River Edge, Nj ; London: World Scientific.

Agrawal, S., Chakrabarti, K., Chaudhuri, S., Ganti, V., Konig, A. C & Xin, D. (2009). Exploiting web search engines to search structured databases. *In Proceedings of the 18th international conference on World wide web (WWW '09)*. Association for Computing Machinery, New York, NY, USA, 501–510. DOI:https://doi.org/10.1145/1526709.1526777

Armstrong, W.W. (1974). Dependency Structures of Data Base Relationships. IFIP Congress.

Brian Berenbach, & Al, E. (2009). Software & systems requirements engineering : in practice. Editorial: New York: Mcgraw-Hill.

Connoly, T., & Begg, C. (2015). Database Systems: A Practical Approach to Design, Implementation, and Management, 6th ed., Boston MA.

Gilbert, S & Lynch, N. (2002). Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. *SIGACT News 33*, (2), pp. 51–59. DOI:https://doi.org/10.1145/564585.564601

Gray, J. (1988). The transaction concept: virtues and limitations. Readings in database systems. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Hema Priya, P., & RangaRaj, R. (2013). An Improved Search Engine by Semantic Web using Ontology, *International Journal of Science and Research (IJSR),2* (12), pp. 403 - 408. Retrieved from https://www.ijsr.net/search_index_results_paperid.php?id=02013583

Hoang, V. T., Spognardi, A. Tiezzi, F., Petrocchi, M.. &Rocco, D. (2015). Domain-specific queries and Web search personalization: some investigations. *Electronic Proceedings in Theoretical Computer Science.188*. DOI:10.4204/EPTCS.188.6.

Lethbridge, Timothy C., Robert Lagani. (2005). Object-Oriented Software Engineering Practical Software Development using UML and Java . NewYork: McGraw-Hill.

Liaw, S., & Huang, H.. (2006). Information retrieval from the World Wide Web: a user-focused approach based on individual experience with search engines. *Computers in Human Behaviour, 22* (3),501-517.

Oppel, A. (2010). Databases Demystified, 2nd ed. McGraw-Hill Professional, Boston, MA
Ramakrishnan, R., & Gehrke, J. (2000). Database Management Systems, 2nd ed. McGraw-Hill Higher Education, Boston, MA

Ramakrishnan, R., & Gehrke, J. (2002). Database Management Systems, 3rd ed. McGraw-Hill Higher Education, Boston, MA

Resources for developers, by developers. (2020, May 20). Retrieved from https://developer.mozilla.org/en-US/

Rob, P., & Coronel, C. (2007). Database Systems: Design, Implementation, and Management, 8th ed. Course Technology, Boston, MA

Sadalage, P. J., & Fowler, M. (2013). NoSQL Distilled. Addison-Wesley Professional, Boston, MA

Silberschatz, A., Korth, H. F., & Sudarshan, S. (2011). Database system concepts. New York: McGraw-Hill.

Stewart, J. M., Chapple, M., & Gibson, D. (2015). CISSP (ISC)2 Certified Information Systems Security Professional Official Study Guide, 7th ed. John Wiley & Sons, Indianapolis, IN

Thuraisingham, B. (2001). *Managing and mining multimedia databases*. Boca Raton, FL: CRC Press.

Tiwari, S. (2011). Professional NoSQL. John Wiley & Sons, Indianapolis, IN

Williams, H. E., & Lane, D. (2004). Web Database Applications with PHP and MySQL, 2nd ed. Sebastopol, CA

Wirth, W., Bocking, T., Karnowski, V., & Pape, T. (2007). Heuristic and systematic use of search engines. *Journal of Computer-Mediated Communication, 12* (2), 778-800.

Zhou, L. (2015). Personalized Web Search.ArXiv, abs/1502.01057. Retrieved June, 10, 2020 from the ArXiv database