

Group: 3D'S
Project: Media Server
Document: SDS v2
Date: February 18 2005

1. Introduction

1.1 Purpose of this document

To more formally describe the relationship between the various modules, their individual tasks, and how the user interfaces with the system.

1.2 Scope of the development project

We are developing a media server that will store media (DVD/CD content). This will only operate on a home network with windows machines. The amount of media stored is dependent on the network's free hard drive space.

There is a recommendation system to help indecisive viewers decide on what movie they wish to see or music to listen to.

Some benefits are that you can store the media on any computer on the network. Since you won't be using the actual disks they can be safely stored and media becomes readily accessible.

1.3 Definitions, acronyms, and abbreviations

Client Application: The system that the user interacts with to interact with the recommendation system, request media, and play back the streaming data.

Database Server: A server that runs a database containing information on movies, music files, and local media location.

IMDB: A website (<http://www.imdb.com>) that maintains a great database full of movie information.

Media: A movie or music data file.

Media Player: A program (implemented by a third party such as Windows Media Player) to play the media sent to it by the streaming server.

Recommendation System: A system that we will design, to make a good decision on what to recommend if a user would like the system to suggest a movie or an appropriate music playlist. This will analyze current preferences (i.e. I want to watch a comedy tonight) and the user's previous viewing practices to make a recommendation.

Streaming: The transfer of large amounts of data over a network. The receiving computer will then display the data, while it is coming in rather than saving it all.

Streaming Server: A server that streams media from the local hard drive to any other computer.

Session: A session consists of a user starting the client program, logging in, either picking some media, or having one recommended, and enjoying the media.

1.4 References

IMDB: <http://www.imdb.com>

Java: <http://java.sun.com/j2se/1.4.2/docs/api/>

Microsoft MFC Library:

http://msdn.microsoft.com/library/default.asp?url=/library/enus/vclib/html/_mfc_Class_Library_Reference_Introduction.asp

<Streaming stuff>:

<Media conversion tool>:

<Media Player>:

<Database server – MySQL>:

1.5 Overview of document

In section 2 we give a high level overview of our system and how a user interfaces with it. Section 3 will be completed in version 2 of this document. Section 4 describes the third party components we are using. Section 5 discusses the tradeoffs and design choices of our system.

2. System architecture description

2.1 Overview of modules / components

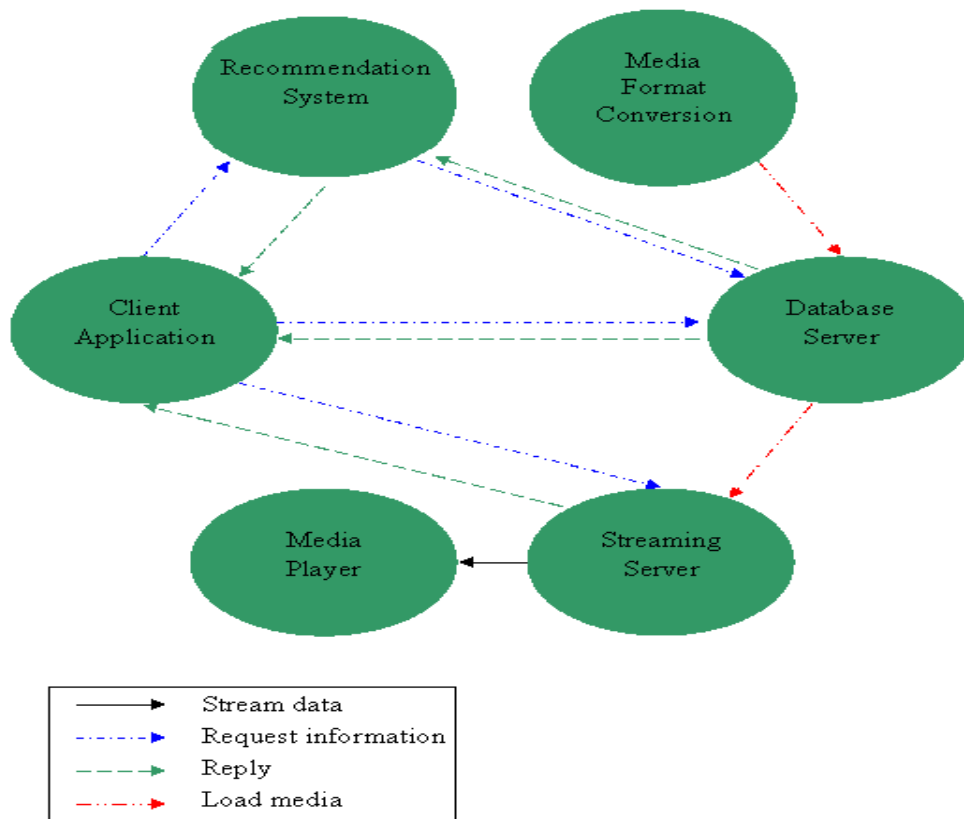
There are four major components to our system, plus a few other components provided by third parties. The user will interface with our system through our client application which has two options. First, a user can choose a movie to watch, or some music to listen to. Otherwise they can ask the advice of our recommendation system on what they should watch or listen to. The streaming server can run on any computer on the network with enough hard drive space to store media. This server will stream media to a users computer. A back end database server will store the locations of the media as well as the viewing history of the various users. The final component is an interface for adding media to the system, and if desired, converting it to a new file format.

2.2 Structure and relationships

User interacts with the client program to request media via either a wizard style interface or an explorer type interface.

User interacts with the database server to add media to the system via a wizard style interface.

This is a graphical representation of how our system interacts.



2.3 User interface issues

The user will start the client application and identify themselves through a login (so we know whose viewing history to reference and update). They can either choose to use the recommendation system, or to select media from the available list. Finally they can watch/listen to the media via Windows Media Player.

Little Billy comes home from school with his homework completed (he got bored in math class). His friends are all in detention for hacking into the school's grading system. Fortunately for Billy, he was getting a snack at the time. (He likes to eat Ho-Ho's while hacking.) Since his cohorts are all serving time, he no longer has a scape goat at his side to take the blame for his exploits on-line. This gives him a couple of hours to watch a movie before SuzyQ comes home from school and starts bossing him around. He logs in, and wonders what to watch. He sends in his preferences, "sci-fi," to the

recommendation system. He receives a list of movies to watch. These include *The Revenge of the Nerds*, *War of the Worlds*, *AI*, *I-Robot*, *Star Wars IV*, *Star Wars VI* and *Sneakers*. Various movies have been weeded out such as *Star Wars V* and *Minority Report*, because he has watched them recently. He chooses to watch *Star Wars VI* because he knows that if he does, he will be able to hone his own ability to use the force.

Two-and-a-half hours later, SuzyQ Teenager comes home after a rough day at school. She actually got caught ditching math class and making out with Brett – the Swedish model/foreign exchange student. She brings her friends with her to talk about the incident and after an hour of sharing the details of the Swedish Hunk's kissing abilities, they decide to watch a movie. At this point Little Billy becomes the servant of “The Brett Svenson Fan Club.” He is sent to make popcorn (“None of that microwave crap!”) and bring them cookies and milk. While they wait, they fire up the media server's recommendation system. They enter in “Brad Pitt” and “since 1995” and wait for the results. The recommendation system returns a list of movies that she hasn't watched more than once in the last week. This list includes *Ocean's Eleven*, *Meet Joe Black*, *Fight Club*, *Seven Years in Tibet*, and *Sinbad: Legend of the Seven Seas*. *Interview with a Vampire* was removed from the results because it was too old. They choose to watch *Meet Joe Black* because it is the longest and gives them more time to feast their eyes on his body.

That evening, Dad comes home all excited. He has purchased four new movies: *Mr. 3000*, *Sky Captain and the World of Tomorrow*, *Ray*, and *The Grudge*. He apologizes to his princess, but he has to get these loaded immediately. (The loading will use up some of the bandwidth, and may cause the loss of data to be a little more common which could cause possible glitches in the streaming of the movie). She complains enough and he collapses under her pouting. He decides he can do it later. After the “Pitt-fest” is over, he immediately begins to load the new movies into the system.

Later that night, after both of the kids are in bed, Mom and Dad go off to their room to enjoy a movie. Having just loaded four new movies, Dad wants to watch one of them. Mom however, has another movie in mind. She wants to watch one of the most romantic classics of all time, *An Affair to Remember*. They enjoy the rest of their evening together....

2.4 User Interface Spec

Located at http://www.cs.utah.edu/~arichard/3DS_UI.html

3. Detailed description of components

3.1 Component template description

Identification	Name associated with this component
Type	Type of component
Purpose	Brief description of purpose
Subcomponents	Classes/Functions that make up this component
Dependencies	Other components used by this one
Interfaces	How this system is to be interfaced with
Resources	Requirements, hardware or software
Processing	Pseudocode located in section 6
Data	Internal data representation

3.2 Client Application

Identification	Client Application running on clients computer
Type	Application/Process
Purpose	An intuitive interface to the entire system
Subcomponents	Login Screen, Choose Screen (choose between recommendation system or movie listing), Recommendation Screen, Select Screen
Dependencies	Network Interface (to access the database), Recommendation System (called as a function), Media Player
Interfaces	Interfaces with the client via GUI
Resources	Monitor, Keyboard, Mouse, Network
Processing	N/A
Data	Uses ADTs from database server (media and client classes)

3.3 Recommendation System

Identification	Recommendation System called by the client application process
Type	Group of functions
Purpose	Makes recommendations based on criteria entered by user and that user's viewing history.
Subcomponents	MakeRecommendation function
Dependencies	Network Interface (to access the database)
Interfaces	MakeRecommendation function called to invoke this system. LinkedList MakeRecommendation(QueryFields query);
Resources	Database, Network
Processing	See 6.1
Data	QueryFields class, ResultSet class

3.4 Media Format Conversion

Identification	Media Format Conversion running on the database server
Type	Application/Process
Purpose	Receives media and converts it to a form requested by the user. It also retrieves information from imdb to insert into the database.
Subcomponents	Upload function (uploads file to server), Retrieve function (retrieves info from imdb), Update function (updates database)
Dependencies	Third party apps: IMDB interface, MPlayer/Mencoder, Lame
Interfaces	Interfaces with user via GUI
Resources	Keyboard, Monitor, Mouse, Network, Database, DVD/CD drive
Processing	Located in 6.2
Data	MovieInfo class, Location class, Feature class, Writers class, Actors class

3.5 Database Server

Identification	Database Server
Type	Network Server
Purpose	Provides a network interface between MySQL and the rest of the system
Subcomponents	MatchData function (runs a query on the database based on callee specified criteria),
Dependencies	MySQL server, Network Interface
Interfaces	Network Interface
Resources	Hard drive space, Network, MySQL
Processing	Located in 6.3
Data	MovieInfo class, Location class, Feature class, Actor class, Writer class, User class

3.6 Streaming Server

Identification	Streaming Server
Type	Network Server
Purpose	Receives a request for a media file and streams it to the requesting client. Receives a request to store media and copies it to the hard drive
Subcomponents	N/A
Dependencies	Network
Interfaces	HTTP/1.1
Resources	Hard drive, Network
Processing	Located in 6.4
Data	N/A

3.7 Network Interface

Identification	Network Interface
Type	Utility classes/functions
Purpose	Classes used by other modules (i.e. client application, database server) to send/receive information over the network.
Subcomponents	N/A
Dependencies	N/A
Interfaces	Connection class
Resources	Network
Processing	Located in 6.5
Data	Message class

4.0 Reuse and relationships to other products

Mplayer/Mencoder is used to re-encode media into the desired stored format.
Lame as an mp3 encoder.
Windows Media Player is used to play streaming media. Other players can be used.
Moviedb written by imdb to access their database.
MySQL is the database we will use.
HttpInputStream is a java class written by Steven for another project that we will be using to read data off the network.

5.0 Design decisions and tradeoffs

Use this section to motivate any decisions that will help the reader understand the design that your team is using. This section can also capture good ideas that were abandoned and the reasons for leaving them out of the design.

6.0 Pseudocode for components

6.1 Recommendation System

```
Recommendation Component () {
    if(DVD) {
        Create and run Database query
        Match query and History components
        Return results
    } else {
        Select from user history
        return results
    }
}
```

6.2 Media Format Conversion

```
Media Conversion Component () {

    Display DVD/MUSIC option Button
    if(DVD){
        Get Title/Year
        if(!Year){
            list = query title (IMDB)
            output list
            Select item with year
        }
        object = query title/year (IMDB)
        Get format type
        Update database
        Upload file
    } else {
        if(< stage3){
            Not supported at this time
        } else {
            query CD
            get format type
            update database
            upload file
        }
    }
}
```

6.3 Database Server

```
MatchData (QueryFeilds data) {

    if(data doesn't specify any query criteria){
        return entire list
    } else {
        Create SQL query
        Run query
        Return list of results
    }
}
```

6.4 Streaming Server

```
if(GET){

    check for the existence of the requested media
    if(found){
        begin streaming it
    } else {
        send error message
    }.
} else if (PUT) {

    if (enough room) {

        download media
```

```
        } else {  
            send error message  
        }  
    }  
}
```

6.5 Network Interface

```
class Connection {  
  
    public Connection(String ipaddr);  
  
    public void close();  
    public Message send(String content, int type);  
    public Message send(File file);  
    public Message receive();  
    public static Connection listen();  
  
    public final int REQUEST;  
    public final int RESPONSE;  
}
```

```
Class Message {  
    public Message();  
    public String getRequest();  
    public void setRequest(String r);  
    public long getLength();  
    public void setLength(long l);  
    public int getResponseCode();  
    public void setResponseCode(int c);  
    public String getBody();  
    public void setBody(String b);  
}
```

6.6 Database Classes

<http://www.cs.utah.edu/~ddurrant/dbclasses.txt>

7.0 Appendices (if any)

None at this time