

Fall 2004 Capstone

Video Control System



Motorola Labs

Kabe VanderBaan

MSU Motorola Team

Brian Miedlar

Ferry Lili

Jae Sik Oh

Thien-Trang Vo

1 Introduction

In this project, Motorola Company suggests to develop a Video Control System with or without Pan-Tilt-Zoom (ptz-enabled) cameras from Sony and Canon brand based on extended VISCA protocol. The system will utilize a central database to register and upload configuration files (in XML format) for each camera, containing camera's location and configuration preference. The system operates in a server-client environment to control and view each camera. The clients download a specific Graphical User Interface (GUI) based on the requested type of camera and downloaded configuration files. However, the main emphasis of this project is on a clean client GUI for ptz-control and on the ability to locate all of the cameras in the system.

2 Problem

Develop a system where:

- a. There exist multiple video sources in a system (i.e., live cameras and stored video) that are accessible via one or more servers. Each server can have multiple cameras connected and can also transmit multiple stored videos (e.g., using a web server).
- b. Some cameras on each server may be capable of being controlled over a network (i.e., pan-tilt-zoom enabled).
- c. A client application can query for and locate each video source in the system using a central database.
- d. After locating the video source in the database, the client application downloads the appropriate GUI from the identified server.
- e. If the video source is a ptz-enabled camera, then the client must display a clean, usable GUI that can control the camera over an Internet Protocol (IP) network.
- f. If the video source is a stored video, then the client must display a GUI that allows for VCR like control (rewind, fast forward, play, pause, stop).
- g. If the video source is a camera that is not ptz-enabled, then a basic GUI is returned.

3 Solution

3.1 Architecture

The architecture of the project is a client-server environment with a central database over the network. Multiple servers (Video Server) may exist in the system; however, there is only a single Video Server application per server at all times. There is only one database that is shared among all control clients and servers in the system. Multiple clients can interact with each server and the database at the same time. Also, the system can have multiple cameras and they do not have to connect to the client directly.

3.2 Video Server

There should be a single Video Server application per server that can control one or more ptz-enabled cameras, and source one or more stored video files. The Video Server should be configured for each video source on the machine (i.e., live camera or stored video files). The Video Server will register each of the cameras and stored video files with a central database (i.e., Berkeley Database) so that clients can locate each video source. The Video Server should be able to enable and disable a video source (i.e., add and remove a video source from the database). Communications between the server and client will initially use Java RMI, but may be moved to another solution as the projects moves further (e.g., SOAP).

3.2.1 Video Server Configuration

Server preferences and video source definitions shall be stored in a configuration file (e.g., server.pref). The file will be stored in XML format and should be parsed and stored in a configuration object (e.g., ServerConfiguration.java). Initially, the configuration should be modified by directly editing the configuration file and later via the GUI on the server.

The following preferences should be included in the server's preference file:

<i>Preference</i>	<i>Description</i>
RMI REGISTRY ADDRESS	The IP address of the machine running the rmi registry.
RMI REGISTRY PORT	The port that the registry is listening for connections on.
RMI REMOTE_NAME	The name that the server will use to register with the registry
DATABASE HOST	The IP address of the machine running the database
DATABASE NAME	The name of the database.
DATABASE USERNAME	The username used to access the database
DATABASE PASSWORD	The password used to access the database

3.2.2 Video Server Command Line Options

The only command line option (besides any VM specific options) should be the location of the preferences file. This should be specified as “-p FILE” (without quotes), where FILE is replaced with the absolute or relative location of the preferences file (e.g., -p pref/server.pref).

3.2.3 VISCA Driver

The VISCA protocol was developed by Sony and is used to control devices over communication links (i.e., serial port). A VISCA driver that was developed by Stefan M. Strasser will be supplied by Motorola Company to control devices that use the VISCA protocol, i.e., Sony brand camera.

Currently, the library allows for a specially formatted XML file to be read in the lists of command name with the command syntax and the driver only supports Sony cameras. Therefore, the library must be extended and an XML file must be created to control Canon cameras.

3.3 Video Control Client

Multiple clients can exist in the system and each client can interact with one or more servers. The client will only control a single video source at a time; therefore, it will also only interact with a single Video Server at a time. The client will initially query the database for all of the video sources in the system. This may later be extended to allow for the system to query for a subset of video sources. After receiving the video sources, a user can select a certain video source and will request the appropriate UI from the server (i.e., `Server.getUI()`).

3.3.1 Video Control Client User Interface (UI)

The client should download the appropriate GUI, and should be capable of display any number of GUIs without any configuration.

3.3.2 Video Control Client Configuration

Client's preferences shall be stored in a configuration file (e.g., `client.pref`). the file will be stored in an XML format and should be parsed and stored in a configuration object (e.g., `ClientConfiguration.java`). Initially, the configuration should be modified by directly editing the configuration file and later via the GUI on the client.

The following preferences should be included in the client's preferences file:

<i>Preference</i>	<i>Description</i>
RMI REGISTRY ADDRESS	The IP address of the machine running the rmi registry.
RMI REGISTRY PORT	The port that the registry is listening for connections on.
RMI REMOTE_NAME	The name of the server's stub in the registry.

3.3.3 Video Control Client Command Line Options

The only command line option should be the location of the preference file. This should be specified as “-p FILE” (without quotes), where FILE is replaced with the absolute or relative location of the preference file.

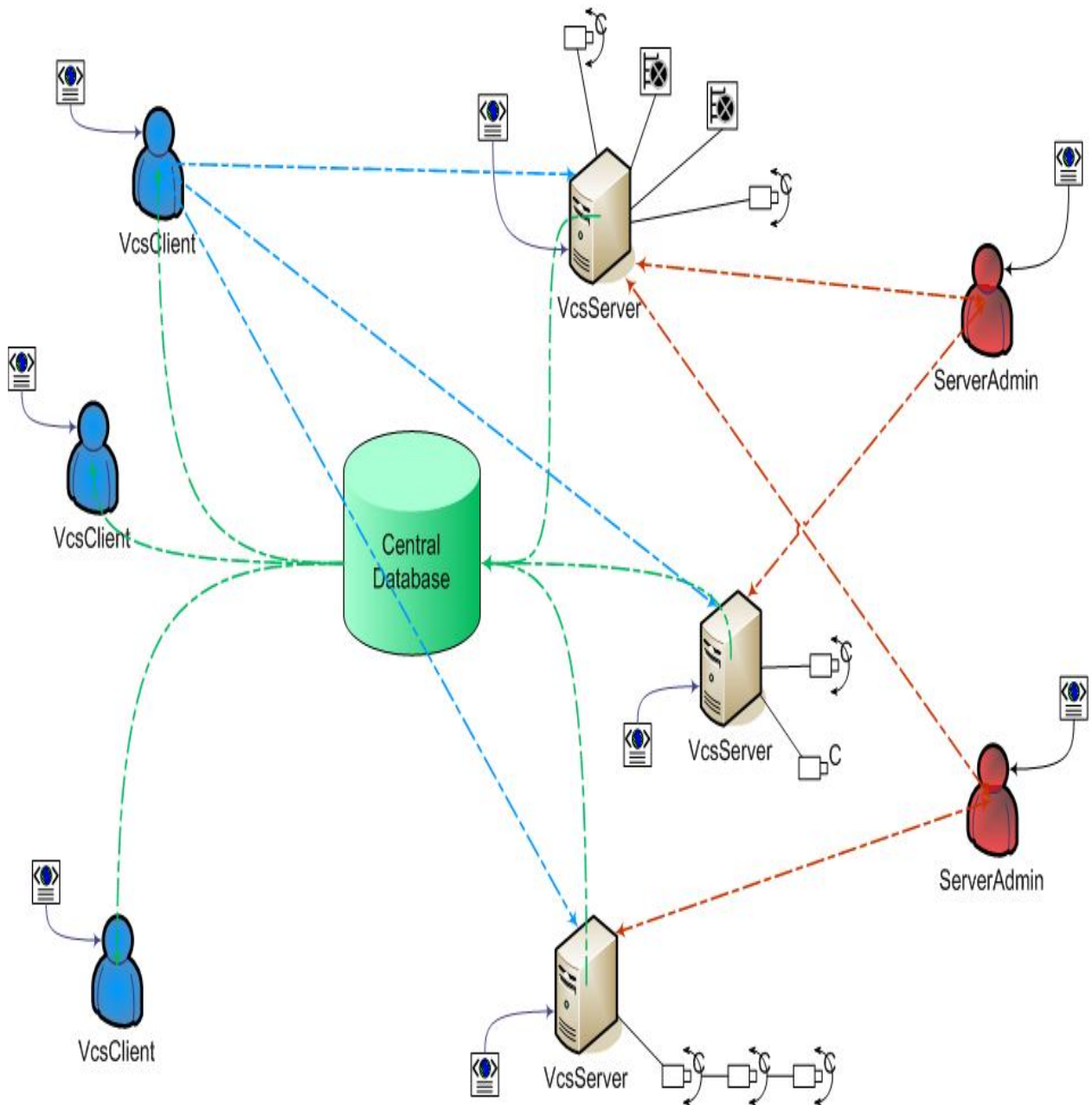
3.4 Database

The Berkeley Database will be used as the central Database in the system. The Motorola Company will supply the Berkeley Database with a java interface integrated. The Berkeley Database that will be supplied could be run on both Linux and Windows machine.

3.5 Admin Client

The admin client is a separate client that will implement the admin interface into the server. The admin client connects directly to video server. The admin client will allow for cameras to be added and removed from the database remotely over the network. In addition, the admin client can change the permission of any cameras in the server to be used or not by any client in the system.

4 Scenario



5 Flow of the Project

5.1 Minimum Requirements

The minimum requirements are the requirements that are expected to be met to be considered a success project. The minimum requirements correspond to Milestone 4 (See Section 5.2.4).

Video Server

- 1) The Video Server can read in a configuration file.
- 2) The Video Server is configured using a configuration file using a `-c` option on the command line.
- 3) Camera drivers are configured using the configuration file.
- 4) The Video Server's connection to the Berkeley Database's RMI registry and the name (e.g., `mediaSpace`) is configured using the configuration file.
- 5) The Video Server's connection to its RMI registry is configured using the configuration file.
- 6) The Video Server parses an XML file into and maintains a list (e.g., array, array list, hash map, etc) of camera and file objects.
- 7) The Video Server has a single VISCA driver and Canon driver.
- 8) The Video Server exposes implemented Canon and all VISCA commands.
- 9) The Video Server can add camera and file descriptions to the Berkeley Database.
- 10) The Video Server can remove camera and file descriptions from the Berkeley Database.
- 11) The Video Server can communicate with multiple cameras and files at the same time (e.g., using `COM1`, `COM2`).
- 12) The Video Server allows for Multiple VISCA cameras or Canon cameras to exist at the same time (i.e., either on different COM ports or daisy chained).
- 13) The Video Server allows for multiple clients to connect at the same time.

Video Control Client

- 1) The Video Control Client can read in a configuration file.
- 2) The Video Control Client is configured using a configuration file using a `-c` option on the command line.
- 3) The Video Control Client's connection to the Berkeley Database's RMI registry and the name (e.g., `mediaSpace`) is configured using the configuration file.
- 4) The Video Control Client can query the database to find all Video Sources using a button
- 5) The Video Control Client displays the list of available cameras and files in a selectable list.
- 6) The Video Control Client can select a single Camera or File to control.
- 7) The Video Control Client can use the RMI information in the selected Camera or File to connect to the Video Server.

- 8) The Video Control Client can control the appropriate camera using the name attribute in the description.
- 9) The Video Control Client will request and receive the appropriate UI from the Video Server.
- 10) The Video Control Client can send the implemented Canon and VISCA commands using the GUI.
- 11) The Video Control Client has a basic UI for a non-PTZ Camera Control.
- 12) The Video Control Client has a basic UI for a File Control (i.e., buttons for play, stop, rewind, forward).
- 13) The Video Control Client has a clean, final, and fully functional UI for the PTZ-Control.

Canon Control

- 1) The Canon Control is able to communicate with the camera and perform all commands (or at least the commands that MSU Motorola Team and Motorola Company agree upon).
- 2) The Canon Control is implemented cleanly as a PTZ “plugin” and shares code with the VISCA driver.

Admin Client

- 1) The Admin Client is configured using a configuration file.
- 2) The Admin Client is configured using a configuration file using a –c option on the command line.
- 3) The Admin Client’s connection to the Video Server’s RMI registry is configured using the configuration file.
- 4) The Admin Client allows for the Video Sources of a Video Server to be added or removed from the Berkeley Database.

5.2 Milestones

5.2.1 Milestone 1

DUE DATE: *Friday, September 17, 2004*

This milestone demonstrates the capability of having the client send a command from a UI and the server communicating with the existing UI.

Video Server

- 1) The Video Server can read in a configuration file.
- 2) The Video Server has Canon Control, VISCA Control, and File objects.
- 3) The Video Server can receive a “Turn camera on” and “Turn camera off” commands and turn a Sony camera on appropriately.

Video Control Client

- 1) The Video Control Client can read in a configuration file.
- 2) A basic UI exists for the PTZ-Control on the Video Control Client (e.g., Power On, Power Off, Left, Right, Up, and Down Arrow).
- 3) The Video Control Client can send a “Turn Power On” and “Turn Power Off” command to the server by clicking a button on the UI.

5.2.2 Milestone 2

DUE DATE: *Friday, October 1, 2004*

This milestone represents the first communication using the Canon driver. The server and client are configured using configuration files and the Berkeley Server is now integrated and being used.

Video Server

- 1) The Video Server can read in a configuration file.
- 2) The Video Server is configured using a configuration file using a `-c` option on the command line.
- 3) Camera drivers are configured using the configuration file.
- 4) The Video Server's connection to the Berkeley Database's RMI registry and the name (e.g., `mediaSpace`) is configured using the configuration file.
- 5) The Video Server's connection to its RMI registry is configured using the configuration file.
- 6) The Video Server parses an XML file into and maintains a list (e.g., array, array list, hash map, etc) of camera and file objects.
- 7) The Video Server has a single VISCA driver and Canon driver.
- 8) The Video Server can receive "Turn camera on" and "Turn camera off" commands and turn a Sony camera on appropriately.
- 9) The Video Server exposes implemented Canon commands.
- 10) The Video Server can add camera and file descriptions to the Berkeley Database.
- 11) The Video Server can add camera and file descriptions to the Berkeley Database.
- 12) The Video Server can communicate with multiple cameras and files at the same time (e.g., using `COM1`, `COM2`).
- 13) The Video Server allows for Multiple VISCA cameras or Canon cameras to exist at the same time (i.e., either on different COM ports or daisy chained).

Video Control Client

- 1) The Video Control Client can read in a configuration file.
- 2) The Video Control Client is configured using a configuration file using a `-c` option on the command line.
- 3) The Video Control Client's connection to the Berkeley Database's RMI registry and the name (e.g., `mediaSpace`) is configured using the configuration file.
- 4) A basic UI exists for the PTZ-Control on the Video Control Client (e.g., Power On, Power Off, Left, Right, Up, and Down arrow).
- 5) The Video Control Client can send a "Turn Power On" or "Turn Power Off" command to the server by clicking a button on the UI.
- 6) The Video Control Client can send the implemented Canon commands using the GUI.
- 7) The Video Control Client can query the database to find all Video Sources using a button.

- 8) The Video Control Client displays the list of available cameras and files in a selectable list.
- 9) The Video Control Client can select a single Camera or File to control.
- 10) The Video Control Client can use the RMI information in the selected Camera or File to connect to the Video Server.
- 11) The Video Control Client can control the appropriate camera using the name attribute in the description.

Canon Control

- 1) The Canon Control is able to communicate with the Canon camera and perform at least one of each type of command, such as Pedestal Control (Pan, Tilt), Camera Control (Power On, Power Off, Zoom, Focus), and System Control (Host/Remote Control, Display, Settings).

5.2.3 Milestone 3

DUE DATE: *Friday, October 22, 2004*

This milestone represents the first communication involving multiple clients and cameras at the same time. The Canon Control will perform full functional commands and display appropriate GUI.

Video Server

- 1) The Video Server can read in a configuration file.
- 2) The Video Server is configured using a configuration file using a `-c` option on the command line.
- 3) Camera drivers are configured using the configuration file.
- 4) The Video Server's connection to the Berkeley Database's RMI registry and the name (e.g., `mediaSpace`) is configured using the configuration file.
- 5) The Video Server's connection to its RMI registry is configured using the configuration file.
- 6) The Video Server parses an XML file into and maintains a list (e.g., array, array list, hash map, etc) of camera and file objects.
- 7) The Video Server has a single VISCA driver and Canon driver.
- 8) The Video Server exposes implemented Canon and all VISCA commands.
- 9) The Video Server can add camera and file descriptions to the Berkeley Database.
- 10) The Video Server can remove camera and file descriptions from the Berkeley Database.
- 11) The Video Server can communicate with multiple cameras and files at the same time (e.g., using `COM1`, `COM2`).
- 12) The Video Server allows for Multiple VISCA cameras or Canon cameras to exist at the same time (i.e., either on different COM ports or daisy chained).
- 13) The Video Server allows for multiple clients to connect at the same time.

Video Control Client

- 1) The Video Control Client can read in a configuration file.
- 2) The Video Control Client is configured using a configuration file using a `-c` option on the command line.
- 3) The Video Control Client's connection to the Berkeley Database's RMI registry and the name (e.g., `mediaSpace`) is configured using the configuration file.
- 4) The Video Control Client can query the database to find all Video Sources using a button
- 5) The Video Control Client displays the list of available cameras and files in a selectable list.
- 6) The Video Control Client can select a single Camera or File to control.
- 7) The Video Control Client can use the RMI information in the selected Camera or File to connect to the Video Server.
- 8) The Video Control Client can control the appropriate camera using the name attribute in the description.

- 9) The Video Control Client will request and receive the appropriate UI from the Video Server.
- 10) The Video Control Client can send the implemented Canon and VISCA commands using the GUI.
- 11) The Video Control Client has a basic UI for a non-PTZ Camera Control.
- 12) The Video Control Client has a basic UI for a File Control (i.e., buttons for play, stop, rewind, forward).

Canon Control

- 1) The Canon Control is able to communicate with the camera and perform all commands (or at least the commands that MSU Motorola Team and Motorola Company agree upon).

5.2.4 Milestone 4

DUE DATE: *November 5, 2004*

This milestone represents fully and clean functional GUI. Admin Client is now integrated and being used.

Video Server

- 1) The Video Server can read in a configuration file.
- 2) The Video Server is configured using a configuration file using a `-c` option on the command line.
- 3) Camera drivers are configured using the configuration file.
- 4) The Video Server's connection to the Berkeley Database's RMI registry and the name (e.g., `mediaSpace`) is configured using the configuration file.
- 5) The Video Server's connection to its RMI registry is configured using the configuration file.
- 6) The Video Server parses an XML file into and maintains a list (e.g., array, array list, hash map, etc) of camera and file objects.
- 7) The Video Server has a single VISCA driver and Canon driver.
- 8) The Video Server exposes implemented Canon and all VISCA commands.
- 9) The Video Server can add camera and file descriptions to the Berkeley Database.
- 10) The Video Server can remove camera and file descriptions from the Berkeley Database.
- 11) The Video Server can communicate with multiple cameras and files at the same time (e.g., using `COM1`, `COM2`).
- 12) The Video Server allows for Multiple VISCA cameras or Canon cameras to exist at the same time (i.e., either on different COM ports or daisy chained).
- 13) The Video Server allows for multiple clients to connect at the same time.

Video Control Client

- 1) The Video Control Client can read in a configuration file.
- 2) The Video Control Client is configured using a configuration file using a `-c` option on the command line.
- 3) The Video Control Client's connection to the Berkeley Database's RMI registry and the name (e.g., `mediaSpace`) is configured using the configuration file.
- 4) The Video Control Client can query the database to find all Video Sources using a button
- 5) The Video Control Client displays the list of available cameras and files in a selectable list.
- 6) The Video Control Client can select a single Camera or File to control.
- 7) The Video Control Client can use the RMI information in the selected Camera or File to connect to the Video Server.
- 8) The Video Control Client can control the appropriate camera using the name attribute in the description.

- 9) The Video Control Client will request and receive the appropriate UI from the Video Server.
- 10) The Video Control Client can send the implemented Canon and VISCA commands using the GUI.
- 11) The Video Control Client has a basic UI for a non-PTZ Camera Control.
- 12) The Video Control Client has a basic UI for a File Control (i.e., buttons for play, stop, rewind, forward).
- 13) The Video Control Client has a clean, final, and fully functional UI for the PTZ-Control.

Canon Control

- 1) The Canon Control is able to communicate with the camera and perform all commands (or at least the commands that MSU Motorola Team and Motorola Company agree upon).
- 2) The Canon Control is implemented cleanly as a PTZ “plugin” and shares code with the VISCA driver.

Admin Client

- 1) The Admin Client is configured using a configuration file.
- 2) The Admin Client is configured using a configuration file using a `-c` option on the command line.
- 3) The Admin Client’s connection to the Video Server’s RMI registry is configured using the configuration file.
- 4) The Admin Client allows for the Video Sources of a Video Server to be added or removed from the Berkeley Database.

5.2.5 Milestone 5

DUE DATE: *If the time still permit us*

This milestone represents the bonus features of the projects.

Video Server

- 1) The Video Server can read in a configuration file.
- 2) The Video Server is configured using a configuration file using a `-c` option on the command line.
- 3) Camera drivers are configured using the configuration file.
- 4) The Video Server's connection to the Berkeley Database's RMI registry and the name (e.g., `mediaSpace`) is configured using the configuration file.
- 5) The Video Server's connection to its RMI registry is configured using the configuration file.
- 6) The Video Server parses an XML file into and maintains a list (e.g., array, array list, hash map, etc) of camera and file objects.
- 7) The Video Server has a single VISCA driver and Canon driver.
- 8) The Video Server exposes implemented Canon and all VISCA commands.
- 9) The Video Server can add camera and file descriptions to the Berkeley Database.
- 10) The Video Server can remove camera and file descriptions from the Berkeley Database.
- 11) The Video Server can communicate with multiple cameras and files at the same time (e.g., using `COM1`, `COM2`).
- 12) The Video Server allows for Multiple VISCA cameras or Canon cameras to exist at the same time (i.e., either on different COM ports or daisy chained).
- 13) The Video Server allows for multiple clients to connect at the same time.

Video Control Client

- 1) The Video Control Client can read in a configuration file.
- 2) The Video Control Client is configured using a configuration file using a `-c` option on the command line.
- 3) The Video Control Client's connection to the Berkeley Database's RMI registry and the name (e.g., `mediaSpace`) is configured using the configuration file.
- 4) The Video Control Client can query the database to find all Video Sources using a button
- 5) The Video Control Client displays the list of available cameras and files in a selectable list.
- 6) The Video Control Client can select a single Camera or File to control.
- 7) The Video Control Client can use the RMI information in the selected Camera or File to connect to the Video Server.
- 8) The Video Control Client can control the appropriate camera using the name attribute in the description.
- 9) The Video Control Client will request and receive the appropriate UI from the Video Server.

- 10) The Video Control Client can send the implemented Canon and VISCA commands using the GUI.
- 11) The Video Control Client has a basic UI for a non-PTZ Camera Control.
- 12) The Video Control Client has a basic UI for a File Control (i.e., buttons for play, stop, rewind, forward).
- 13) The Video Control Client has a clean, final, and fully functional UI for the PTZ-Control.
- 14) The Video Control Client allows for the configuration file to be modified and saved using a GUI.
- 15) The Video Control Client can query the database to find some Video Sources (i.e., using various criteria).
- 16) The Video Control Client can download a stored Video Files.
- 17) The Video Control Client can play, pause, rewind, forward, etc. a stored Video Files.

Canon Control

- 1) The Canon Control is able to communicate with the camera and perform all commands (or at least the commands that MSU Motorola Team and Motorola Company agree upon).
- 2) The Canon Control is implemented cleanly as a PTZ “plugin” and shares code with the VISCA driver.

Admin Client

- 1) The Admin Client is configured using a configuration file.
- 2) The Admin Client remotely communicates with the Video Server.
- 3) The Admin Client allows for the Video Sources of a Video Server to be added or removed from the Berkeley Database.
- 4) The Admin Client remotely communicates with the Video Server.
- 5) The Admin Client allows for the configuration file to be modified and saved using the UI.
- 6) The Admin Client allows for the configuration file to be reloaded (i.e., objects removed from Berkeley Database and replaced with updated information).

5.3 Test Cases

This section describes the test cases for each application in the project and which milestone they correspond to. These are some of the cases that the system should handle gracefully. A failed test case does not imply that the application should exit; rather, it should be handled correctly.

5.3.1 Video Server

5.3.1.1 Configuration file

- 1) -c option in the command line argument does not point to the right location of the configuration file
- 2) -c option in the command line argument is missing
- 3) Configuration file's Root is not CONFIG tag
- 4) CONFIG does not have MACHINE child
- 5) MACHINE does not have NAME attribute
- 6) CONFIG does not have DATABASE child
- 7) DATABASE does not have HOST, PORT, or NAME attributes
- 8) DATABASE does not have RMI child
- 9) RMI does not have HOST, PORT, or NAME attributes
- 10) CONFIG does not have LOCATION child
- 11) LOCATION does not have BUILDING, FLOOR, or ROOM attributes
- 12) LOCATION has no text
- 13) CONFIG does not have PTZ tag
- 14) PTZ does not have TYPE, DEV, SOCKETS, or LOG attributes
- 15) PTZ TYPE is not either VISCA or CANON
- 16) PTZ DEV attribute points to a COM port that is in use or invalid
- 17) PTZ SOCKETS attribute is not greater than 0
- 18) PTZ COMMANDS does not point the right location
- 19) PTZ LOG attributes is invalid
- 20) CONFIG does not have any CAMERA children
- 21) CAMERA does not have NAME, MAKE, or MODEL attributes
- 22) CAMERA does not have CONTROL child
- 23) CONTROL does not have TYPE, DEV, or ID attributes
- 24) CONTROL DEV attributes points to a PTZ that does not exist
- 25) CONTROL ID attribute is not greater than 0 or no lower than 9
- 26) Combination of all CONTROL ID attributes is not sequential
- 27) CAMERA does not have DESCRIPTION child
- 28) DESCRIPTION has no text
- 29) CAMERA does not have LOCATION child
- 30) CAMERA does not have MEDIA child
- 31) MEDIA does not have TYPE, TRANSPORT, ADDR, or PORT attributes

- 32) MEDIA does not have FORMAT child
- 33) FORMAT does not have PTYPE attribute
- 34) CONFIG does not have any FILE children
- 35) FILE does not have NAME, PATH, HOST, or PORT attributes
- 36) FILE does not have DESCRIPTION child
- 37) DESCRIPTION has no text

5.3.1.2 Program Execution

- 1) Berkeley Server is not available initially
- 2) Berkeley Server becomes unavailable
- 3) Multiple VCS Clients try to access the same PTZ cameras within 20 seconds
- 4) VCS Client attempts to control camera or file that is not available
- 5) VCS Client attempts to control camera that does not have PTZ control
- 6) PTZ Camera is not in Host Control Mode
- 7) Application runs in the machine has no COM port or less than the defined COM ports in the Configuration file
- 8) Multiple servers are trying to access the same COM port

5.3.2 Video Client

5.3.2.1 Configuration file

- 1) -c option in the command line argument does not point to the right location of the configuration file
- 2) -c option in the command line argument is missing
- 3) Configuration file's Root is not CONFIG tag
- 4) CONFIG does not have DATABASE child
- 5) DATABASE does not have HOST, PORT, or NAME attributes
- 6) DATABASE does not have RMI child
- 7) RMI does not have HOST, PORT, or NAME attributes
- 8) CONFIG does not have LOCATION child
- 9) LOCATION does not have BUILDING, FLOOR, or ROOM attributes
- 10) LOCATION has no text

5.3.2.2 Program Execution

- 1) Camera/File becomes unavailable
- 2) Camera is configured for PTZ but does not actually have PTZ control
- 3) Berkeley Server is not available initially
- 4) Berkeley Server becomes unavailable
- 5) Camera is disconnected

5.3.3 Admin Client

5.3.3.1 Configuration file

- 1) -c option in the command line argument does not point to the right location of the configuration file
- 2) -c option in the command line argument is missing
- 3) Configuration file's Root is not CONFIG tag
- 4) CONFIG does not have SERVER child
- 5) SERVER does not have any RMI children
- 6) SERVER has more than one RMI child
- 7) RMI does not have HOST, PORT, or NAME attributes

5.3.3.2 Program Execution

- 1) Cannot connect to Video Server initially
- 2) A Video Server becomes unavailable after initial connection
- 3) Change status of Camera/File to unavailable while it is already unavailable
- 4) Change status of Camera/File to available while it is already available
- 5) On refresh, some or all Video Servers are not available

5.4 Tools

5.4.1 Eclipse IDE version 3.0.1

An integrated Development Environment (IDE) is not required but is highly recommended. However, any tools that you use (i.e., a GUI generator) **MUST** be able to integrate with the Eclipse IDE.

5.4.2 ANT

Ant is to be used for all compilation and scripting. The ant file that is created must be able to integrate into the Eclipse IDE.

5.4.3 Remote Method Invocation (RMI) Plugin

RMI requires a special compiler and some extra steps in order to be used. All the tools to use RMI are either included with Java, or are freely available. An RMI plugin has been developed for Eclipse that eases a lot of the headache associated with RMI. Therefore, the software will aid in the development. An RMI plugin will be supplied by Motorola Company.

5.5 Technologies

Ant

Eclipse

Berkeley Database

Java

Java RMI

Model View Controller

5.6 Useful Links

5.6.1 Eclipse 3.0.1

<http://www.eclipse.org>

5.6.2 Java Plugin for Eclipse (sun's java vm 1.4.2)

<http://java.sun.com/j2se/1.4.2/index.jsp>

5.6.3 RMI Plugin for Eclipse

<http://www.genady.net/mi/index.html>

5.6.4 Berkeley Database

<http://www.xml.com/pub/a/2003/05/07/bdb.html>

5.6.5 Java Networking

<http://www.cafeaulait.org/course/week12/index.html>

<http://www.cafeaulait.org/>

5.6.6 Model View Controller

<http://csis.pace.edu/~bergin/mvc/mvcgui.html>

<http://java.sun.com/blueprints/patterns/MVC-detailed.html>

<http://exciton.cs.oberlin.edu/JavaResources/DesignPatterns/MVC.htm>

6 Code Specifics

6.1 Files and Packages

- 1) Canon Protocol
- 2) VISCA Protocol
- 3) Video Control Client's GUI
- 4) Video Control Admin Client's GUI
- 5) Video Control Server's GUI
- 6) Video Control Client's Configuration file (XML format) stored in the Database for camera reference
- 7) Video Control Client's RMI Interface
- 8) Video Control Server's RMI Interface
- 9) Video Control Admin Client's RMI Interface
- 10) Video Control Server's drivers protocol
- 11) Video Control Server's Configuration file register to the Database
- 12) Video Control Admin Client's Configuration file register to the Database
- 13) Database Interface to handle registering and uploading video source configuration files

6.2 Interfaces

This section describes the interfaces that exist on the server. The interfaces described here may change as the project evolves. Each interface should be implemented as an object on the server.

6.2.1 VideoSourceControlIF

The interface that the Client uses to send control messages to the Video Server.

6.2.2 VideoServerAdminIF

The interface that either a local or remote client can use (Note: this client is different than the Control Client) to enable or disable a video source.

6.2.3 VideoServerQueryIF

The interface that either a local or remote client uses to query the Video Server.

7 Contact Information

7.1 Motorola Labs

7.1.1 Kabe VanderBaan
Senior Research Engineer
Networks and Infrastructure Research Lab
Motorola Labs
Work: (847)576-1832
E-mail: kabev@motorola.com

7.2 Michigan State University Motorola Team

7.2.1	Brian Miedlar	miedlarb@msu.edu
7.2.2	Ferry Lili	liliferr@msu.edu
7.2.3	Jae Sik Oh	ohjae1@msu.edu
7.2.4	Thien-Trang Vo	vothien@msu.edu