International Journal of Computational Intelligence and Informatics, Vol. 2: No. 1, April - June 2012

A Network-based Remote Controlling and Monitoring System with More Security and Platform-free Features(NRCM-SP)

Chiranji Lal Chowdhary
School of Information Technology and Engineering
VIT University
Vellore, India
chiranji.lal@vit.ac.in

Chandra Mouli P.V.S.S.R.
School of Computing Science and Engineering
VIT University
Vellore, India
chandramouli@vit.ac.in

Abstract- **With increasing use of internet technologies in general life, it's becoming more challenging to solve problems like, remote monitoring and controlling and other related operations. It is not easy to access huge data from many network systems to a single network system. If it is widely distributed then it demands a more strong system. To overcome such problems, a more reliable and secured platform-free remote controller is required which is having the ability to monitor the system. Many researchers are working for this to apply internet technology for developing general and expendable system architecture. In this paper a novel Network-based Remote Controlling and Monitoring System with More Security and Platform-free Features (NRCM-SP) is proposed. The major contribution in this work is to use the network base for the purpose of real-time remote monitoring and controlling of processing equipment. (*Abstract*)**

Keywords: Platform free, Network-base, Remote Controlling and Monitoring system, One-to-One Remote Control.

I. INTRODUCTION

With the development in computer science, engineering and technology, work related to controlling and monitoring was increased. Users need a well-defined secured system with ability to control huge and widely spread data from a single node point. Many existing systems [1-8] are based on the protection of regular equipments, special devices, and data. Now people are more interested in controlling home devices from remote. Various approaches are proposed to help needy people to access their equipments at home. Some of such equipments are air-conditioners, heaters, lights, and safety and security devices, are needed to be controlled from remote [14]. The internet use for the remote control provides easy access on very low cost. Some development on the architecture issues in many-to-many control is included in current research topics. As controlled devices are considered as service and therefore service-oriented architecture forms the basic framework for remote control. Many of the currently offered designs [15-20] are giving importance of integrating web resources for remote control. So, some significance is given to establishing a remote control services development environment.

The first step in remote control laboratory [11] was about the basic virtual instruments design, used for real-time experiments in a remotely accessed environment. This approach [11-12] was flexible and responsible to the client side as the remote user compiles and executes the controller locally. Some web-based remote control services [4, 10] simultaneously provide controlling and many remote control services through which any person was able to control, by using a simple web browser or any pre-registered device through the Internet. Next work was done on a remote monitoring and control architecture, that consists of a Web-services-based monitoring and control gateway [5] which describes Ethernet-ready I/O modules for safety detection modules, Web cameras, and networks. It will not only provide general functions of remote monitoring and control but also possess the mechanism of actively detecting appliance's abnormal power consumption and ensuring appliance safety. Some others worked on Internet-based remote health and activity of daily living (ADL) monitoring system [9] for the elders using the integrated medical mechatronics techniques. An original solution to perform the remote control of measurement instrumentation [13], with the aim of realizing the experimental section of an E-Learning portal of Electrical and Electronic Measurement courses, based on Web Services implemented in Visual Basic.Net.

Guo et al. [3] proposed an Internet based telemedicine system for the tele-monitoring of the medical data and the virtual instruments. The virtual instruments such as electrocardiogram (ECG), electromyogram (EMG) and electroencephalogram (EEG) were implemented using the LabView (National Instruments, Austin, TX, USA) modules. The proposed system was capable of extracting the valuable diagnostic information using embedded advanced biomedical signal processing algorithms. The system had been tested, and it used the Java applets and pc Any where software to realize the medical data and signal transmissions via the 300 kpbs cable modem.

An earlier system [9] was composed of the main controller, remote controller, host computer, physiological signal acquisition device, sensor sets, and video camera. In that, the main controller was designed as a communicable single processor based controller. It was responsible for the signal collections from the sensor sets and physiological signal collection devices, home automation and security system. Additionally, the main controller can communicate with the host computer using the serial communication and the portable remote controller using the wireless RF data receiver.

A simple registration and message forwarding system will not be sufficient for the purpose of accessing a widely distributed and huge data from many network systems to a single network system. With the obvious securities, now-a-days some kind of conflict resolution is necessary if different users are commanding a machine to different things at the same time [9]. Different conflict resolution users will need to be explored as the integrity and safety of the controlled machine will need to be maintained. In short, industry standards should be used for message passing to provide a service platform that is hardware and software independent [2]. This proposed is an extension of the earlier work done in [21]. In this proposed system, network-based system handles different type of devices with different kinds of components and also can handle large number of users with simultaneous multiple user access, if required in future.

The need for controlling and monitoring remote systems is increasing day by day with more security and minimum constraints on processing and the use of software and hardware. Some objectives to build a platform-free implementation of proposed architecture are:

1. Web based diagnostic of remote devices.
2. Remote data viewing and data feeding.
3. Reduces hard disk constraints.
4. Accessing and monitoring the usage of remote systems and its devices.
5. Patient’s condition monitoring by doctor.
6. Monitoring remote experiments.
7. Using software installed in remote system.
8. Using services which are there in remote systems. For ex- Internet.
9. Multi-node viewing can be done of the remote device.
10. Can access and monitor multiple remote systems’ data simultaneously.
11. Compatible with different operating systems.

In the proposed system, it is assumed to have server and client access to the network either through *wifi* or cable. In that system, it is to listen by server, to the port on which client want to connect. As Java is a platform independent language so the system will also be the same. The speed of the network must be relatively faster for quick and smooth image transfer.

The rest of the paper is organized as follows. In Section 2, the system architecture of proposed system is planned. In Section 3, different test cases are noted. Testing results are in section 4. The discussion on the testing results is constructed in Section 5. Section 5 concludes the work with future work.

II. PROPOSED SYSTEM ARCHITECTURE

The architecture of the remote control service system (Fig. 1) consists of three layers, Data Exchange Layer (DEL), Service Thread Layer (STL) and Software Interface Layer (SIL). Data Exchange Layer implements the conversation of different protocols, commands and signals. DEL also takes care of standardization of object and message. Service Threads Queue provides communication disruption handling and different service management for multiple user access. Intelligent service also can be provided. Software Interface Layer manipulates the controlled hardware, for example, new computer and operate system, such as Window XP, does not allow user to visit underlying hardware interfaces directly by using static RMI, the hardware DLL and OCI drive files are needed to implement a dynamic communication for the underlying hardware.

Many researchers wrote about the potential of using the web for the purpose of real-time remote monitoring control of processing equipment. The architecture of internet-based remote control and monitoring was studied, and then explanation of web based remote control service is given [3].

The design of the network-based monitoring system component generally includes four mechanisms:

1. Actively monitoring and controlling the working of applications.
2. Detecting abnormal behavior of applications.
3. Recording data into the database, and
4. Promptly delivering messages to the person in charge.

Network-based remote control focused on real-time remote control, by using CGI, COM, COBA technologies to implement a static Remote Method Invoke (RMI). Another typical remote control and monitoring is web-based remote control, by using XML, SOAP and IIS technologies to transmit web pages requests and responses within an executing program, client can invokes a web service by sending an XML message by using IE to send control command and browse the feedback results.

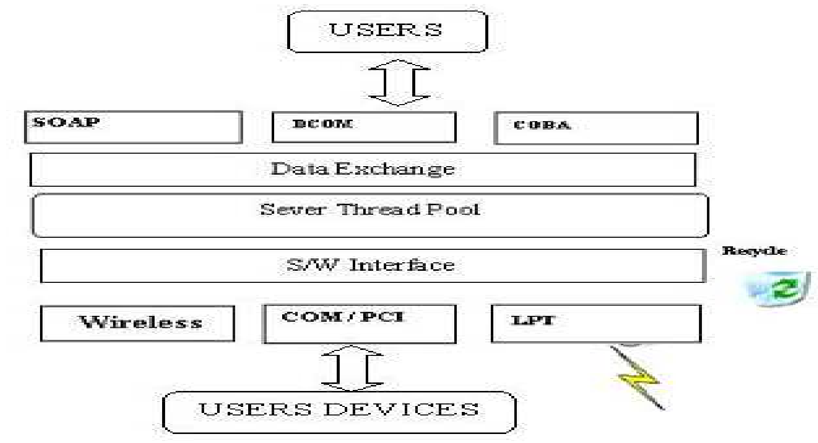

Figure 1. Architecture of network-based remote control

Fig. 2 shows architecture of one-to-one remote control. It consists of client and server systems, a network for connection between the two and an authentication graphical user interface for secure and easy usage of the system. In place of architecture of one-to-one remote control, it is assumed that the architecture of many-to-one remote control system exists with more than one wireless user device connected to a single remote controller.

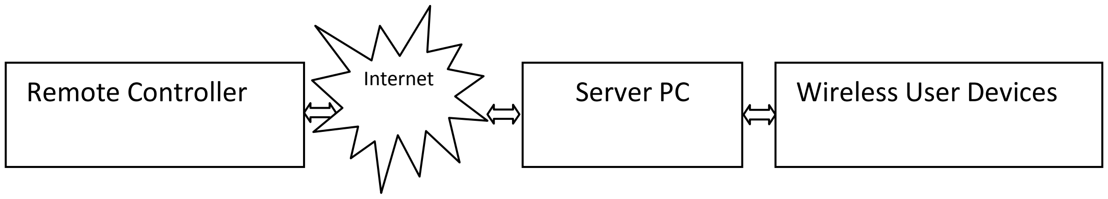

Figure 2. Architecture of one-to-one Remote control

The proposed remote control and monitoring system was a remote-based client-server system. The server components were running on the machine, and controlled remotely from another machine as shown in Fig. 3.

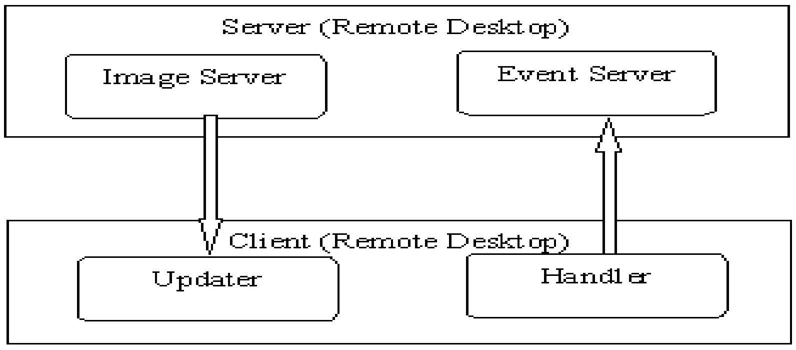

Figure 3. Remote Control and Monitoring

Two parts of remote desktop server are: Image Server and Event Server. Image server was a thread that will periodically take screen shots of the desktop and send it to the client over the network. Other part, i.e. event server, is an entity that will reserve the keyboard and mouse events sent by the client and emulate the same on the server.

The client (remote desktop) of the remote control and monitoring system was also divided into image updater and desktop event handler. This component will be run on a machine from where the user wants to control the remote machine. The image updater gets the images of the server’s desktop and displays it to the user and the desktop events handler is an entity that records the keyboard and mouse actions performed by the user on the displayed image(of the server’s desktop) and send these events to the server.

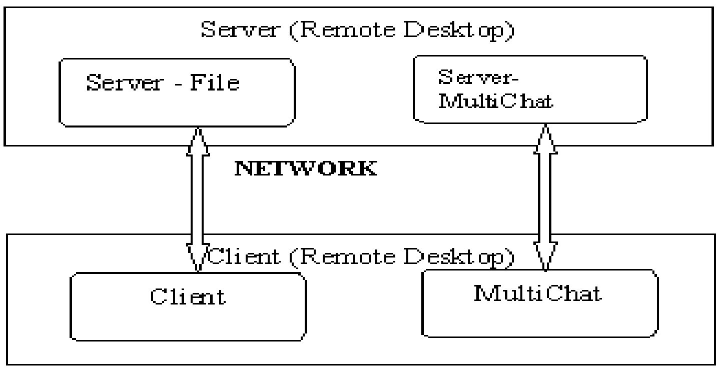

Figure 4. File Transfer and MultiChat

Fig. 4 describes the file transfer and multi-chatting between client and server.

III. TEST CASES

3.1 1. Test case name – Interface Testing
2. Objective/purpose – To test proper building of the interface (GUI)
3. Test case ID – **TC1**
4. Test Case description - To find out whether GUI is being built or not after running the function.
5. Steps to be executed - Run the function BuildGUIframe() without any input.
6. Expected result - GUI built.
7. Actual result - GUI built.
8. Pass/Fail - Pass if Expected result is same as Actual Result.
9. Bug Number - 0
10. Test result – Pass

3.2 1. Test case name – Authentication and response Testing
2. Objective/purpose – To test username and password.
3. Test case ID – **TC2**
4. Test Case description – To test whether options for choosing server comes on entering correct username and password.
5. Steps to be executed – Enter username and password in authentication page.
6. Expected result – Authenticated on giving correct username and password.
7. Actual result - Authenticated on giving correct username and password.
8. Pass/Fail criteria - Pass if Expected result is same as Actual Result.
9. Bug Number - 0
10. Test result – Pass

3.3 1. Test case name - Connection establishment of file server and receiving file-path from file client Testing
2. Objective/purpose – To test file server and file client.
3. Test case ID – **TC3**
4. Test Case description - To test whether file server is receiving file name from file client.
5. Steps to be executed – Start File Server and enter file path at file client side and send to file server on connection establishment.
6. Expected result – File name received at client side.
7. Actual result - File name received at client side.
8. Pass/Fail criteria - Pass if Expected result is same as Actual Result.
9. Bug Number - 0
10. Test result – Pass

3.4 1. Test case name - File server sending file to client testing.
2. Objective/purpose – To test the file received at client end.
3. Test case ID – **TC4**
4. Test Case description - To test whether the file received at client end is the same as what was to be

received.
5. Steps to be executed – Run server function for sending file and client function for receiving file.
6. Expected result – File sent by server and same file being received by client.
7. Actual result - sent by server and same file being received by client..
8. Pass/Fail criteria - Pass if Expected result is same as Actual Result.
9. Bug Number - 0
10. Test result – Pass

3.5 1. Test case name - Multi-chat server sending messages to clients testing.
2. Objective/purpose – To test the chat message received at client end.
3. Test case ID – **TC5**
4. Test Case description - To test whether the message received at client end is the same as what was to be received.
5. Steps to be executed – Run server function for sending chat messages and client function for receiving chat messages.
6. Expected result – Message sent by server and same chat message being received by client.
7. Actual result - sent by server and same message being received by client.
8. Pass/Fail criteria - Pass if Expected result is same as Actual Result.
9. Bug Number - 0
10. Test result – Pass

3.6 1. Test case name - Multi-chat client sending messages to server testing.
2. Objective/purpose – To test the chat message received at server end.
3. Test case ID – **TC6**
4. Test Case description - To test whether the message received at server end is the same as what was to be received.
5. Steps to be executed – Run client function for sending chat messages and server function for receiving chat messages.
6. Expected result – Message sent by server and same chat message being received by client.
7. Actual result – message sent by client and same message being received by server.
8. Pass/Fail criteria - Pass if Expected result is same as Actual Result.
9. Bug Number - 0
10. Test result – Pass

3.7 1. Test case name - Monitoring server robot action queue testing
2. Objective/purpose – To test Robot Action Queue.
3. Test case ID – **TC7**
4. Test Case description - To test whether reader thread is reading the actions and addJob() is adding jobs to the queue or not.
5. Steps to be executed – Do some action like clicking or opening a file.
6. Expected result – Reader thread is reading the actions and addJob() is adding jobs to the queue.
7. Actual result - Reader thread is reading the actions and addJob() is adding jobs to the queue.
8. Pass/Fail criteria - Pass if Expected result is same as Actual Result.
9. Bug Number - 0
10. Test result – Pass

3.8 1. Test case name - Monitoring server ScreenShot Thread() testing
2. Objective/purpose – To test ScreenShot Thread.
3. Test case ID – **TC8**
4. Test Case description - To test whether ScreenShot Thread is taking screen shots and compressing the image before sending.
5. Steps to be executed – Run ScreenShot thread function.
6. Expected result – ScreenShot Thread is taking screen shots and compressing the image.
7. Actual result - ScreenShot Thread is taking screen shots and compressing the image.
8. Pass/Fail criteria - Pass if Expected result is same as Actual Result.

9. Bug Number - 0
10. Test result – Pass

3.9 1. Test case name - Monitoring server WriterThread() testing
2. Objective/purpose – To test WriterThread function.
3. Test case ID – **TC9**
4. Test Case description - To test whether WriterThread is writing/sending jobs to the client after taking it from Robot Action Queue or not.
5. Steps to be executed – Run WriterThread function.
6. Expected result – WriterThread is writing/sending jobs to the client after taking it from Robot Action Queue.
7. Actual result - WriterThread is writing/sending jobs to the client after taking it from Robot Action Queue.
8. Pass/Fail criteria - Pass if Expected result is same as Actual Result.
9. Bug Number - 0
10. Test result – Pass

3.10 1. Test case name - Monitoring client Read Action Objects() & Execute() Testing.
2. Objective/purpose – To test Read Action Objects() & Execute() functions.
3. Test case ID – **TC10**
4. Test Case description - To test whether Read Action Objects() is receiving/reading jobs from the server and after that passing it to the Execute() to exeute that job or not.
5. Steps to be executed – Run Read Action Objects() function.
6. Expected result – Read Action Objects() is receiving/reading jobs from the server and after that passing it to the Execute() to exeute that job.
7. Actual result - Read Action Objects() is receiving/reading jobs from the server and after that passing it to the Execute() to exeute that job.
8. Pass/Fail criteria - Pass if Expected result is same as Actual Result.
9. Bug Number - 0
10. Test result – Pass

IV. TESTING RESULTS

A. Unit Testing and Result

Unit testing focuses verification effort on the smallest unit of software i.e. the module. Using the detailed design and the process specifications, testing is done to uncover errors within the boundary of the module. All modules must be successful in the unit test before the start of the integration testing begins.

In this work, each service can be thought of a module. Each module has been tested by giving different sets of inputs. When developing the module as well as finishing the development, the module works without any error. The inputs are validated when accepting them from the user.

a) Interface and Authentication Unit Testing

1. Test Case Name – Interface and Authentication Unit Testing
2. Objective/Purpose –
a) To test proper building of the interface (GUI)
b) To test username and password.
3. Test Case ID – **UT1**
4. Test Case Description –
a) To find out whether GUI is being builded or not after running the function.
b) To test username and password.
5. Steps to be Executed –
a) Run the function BuildGUIframe() without any input
b) username and password in authentication page
6. Expected Result –
a) GUI builded

b) Authenticated on giving correct username and password
7. Actual Result -
a) GUI built

b) Authenticated on giving correct username and password

8. Pass/Fail - Pass if Expected result is same as Actual Result.
9. Bug Number – 0
10. Test result – Pass

b) File Server and File Client Unit Testing

1. Test Case Name – File Server and File Client Unit Testing
2. Objective/Purpose –
a) Test file server and file client.
b) To test the file received at client end.
3. Test Case ID – **UT2**
4. Test Case Description –
a) To test file server and file client
b) To test whether the file received at client end is the same as what was to be received
5. Steps to be Executed –
a) Start File Server and enter file path at file client side and send to file server on connection establishment
b) Run server function for sending file and client function for receiving file.
6. Expected Result –
a) File name received at client side.
b) File sent by server and same file being received by client.
7. Actual Result -
a) File name received at client side.

b) File sent by server and same file being received by client.

8. Pass/Fail - Pass if Expected result is same as Actual Result.
9. Bug Number - 0
10. Test result – Pass

c) Multi-Chat Server & Client Unit Testing

1. Test Case Name – Multi-Chat Server & Client Unit Testing
2. Objective/Purpose –
a) To test the chat message received at client end.
b) To test the chat message received at server end
3. Test Case ID – **UT3**
4. Test Case Description –
a) To test the chat message received at client end.
b) To test the chat message received at server end
5. Steps to be Executed –
a) Run server function for sending chat messages and client function for receiving chat messages.
b) Run client function for sending chat messages and server function for receiving chat messages.
6. Expected Result –
a) Message sent by server and same chat message being received by client.
b) Message sent by server and same chat message being received by client.
7. Actual Result -
a) Message sent by server and same chat message being received by client.
b) Message sent by server and same chat message being received by client.

8. Pass/Fail - Pass if Expected result is same as Actual Result.

9. Bug Number - 0
10. Test result – Pass

d) Monitoring Server Unit Testing

1. Test Case Name – Monitoring Server Unit Testing
2. Objective/Purpose –
a) To test Robot Action Queue
b) To test ScreenShot Thread .
c) To test WriterThread function
3. Test Case ID – **UT4**
4. Test Case Description –
a) To test whether reader thread is reading the actions and addJob() is adding jobs to the queue or not.
b) To test whether ScreenShot Thread is taking screen shots and compressing the image before sending.
c) To test whether WriterThread is writing/sending jobs to the client after taking it from Robot Action Queue or not.
5. Steps to be Executed –
a) Do some action like clicking or opening a file
b) Run ScreenShot thread function.
c) Run WriterThread function.
6. Expected Result –
a) Reader thread is reading the actions and addJob() is adding jobs to the queue
b) ScreenShot Thread is taking screen shots and compressing the image.
c) WriterThread is writing/sending jobs to the client after taking it from Robot Action Queue.
7. Actual Result -
a) Reader thread is reading the actions and addJob() is adding jobs to the queue

b) ScreenShot Thread is taking screen shots and compressing the image.
c) WriterThread is writing/sending jobs to the client after taking it from Robot Action Queue.
8. Pass/Fail - Pass if Expected result is same as Actual Result.

9. Bug Number - 0
10. Test result – Pass

e) Monitoring Client Unit Testing

1. Test Case Name – Monitoring Client Unit Testing

2. Objective/purpose – To test Read Action Objects() & Execute() functions.
3. Test case ID – **UT5**
4. Test Case description - To test whether Read Action Objects() is receiving/reading jobs from the server and after that passing it to the Execute() to exeute that job or not.
5. Steps to be **executed** – Run Read Action Objects() function.
6. Expected result – Read Action Objects() is receiving/reading jobs from the server and after that passing it to the Execute() to exeute that job.
7. Actual result - Read Action Objects() is receiving/reading jobs from the server and after that passing it to the Execute() to exeute that job.
8. Pass/Fail criteria - Pass if Expected result is same as Actual Result.
9. Bug Number - 0
10. Test result – Pass

B. Integration Testing and Result

After unit testing, we have to perform integration testing. The goal here is to see if modules can be integrated properly, the emphasis being on testing interfaces between modules. This testing activity can be considered as testing the design and hence the emphasis on testing module interactions.

In this paper the main system is formed by integrating all the modules. When integrating all the modules have been checked whether the integration effects working of any of the services by giving different combinations of inputs with which the two services run perfectly before Integration.

a) Integration Testing

1. Test Case Name – Integration Testing
2. Objective/Purpose – To test proper running of integrated system.
3. Test Case ID – **IT1**
4. Test Case Description –

a) To find out whether GUI is being builded or not after running the function.
b) To test username and password.
c) To test file server and file client
d) To test whether file received at client end is the same as what was to be received
e) To test the chat message received at client end.
f) To test the chat message received at server end
g) To test whether reader thread is reading the actions and addJob() is adding jobs to the queue or not.
h) To test whether ScreenShot Thread is taking screen shots and compressing the image before sending.
i) To test whether WriterThread is writing/sending jobs to the client after taking it from Robot Action Queue or not.
j) To test whether Read Action Objects() is receiving/reading jobs from the server and after that passing it to the Execute() to exeute that job or not.

5. Steps to be Executed –

a) Run the function BuildGUIframe() without any input
b) username and password in authentication page
c) Start File Server and enter file path at file client side and send to file server on connection establishment
d) Run server function for sending file and client function for receiving file.
e) Run server function for sending chat messages and client function for receiving chat messages.
f) Run client function for sending messages and server function for receiving messages.
g) Do some action like clicking or opening a file
h) Run ScreenShot thread function.
i) Run WriterThread function.
j) Run Read Action Objects() function

6. Expected Result –

a) GUI built
b) Authenticated on giving correct username and password
c) File name received at client side.
d) File sent by server and same file being received by client.
e) Message sent by server and same chat message being received by client.
f) Message sent by server and same chat message being received by client.
g) Reader thread is reading the actions and addJob() is adding jobs to the queue
h) ScreenShot Thread is taking screen shots and compressing the image.
i) WriterThread writing/sending jobs to the client by taking it from Robot Action Queue.
j) Read Action Objects() is receiving/reading jobs from the server and after that passing it to the Execute() to exeute that job.

7. Actual Result -

a) GUI builded
b) Authenticated on giving correct username and password
c) File name received at client side.
d) File sent by server and same file being received by client.
e) Message sent by server and same chat message being received by client.
f) Message sent by server and same chat message being received by client.
g) Reader thread is reading the actions and addJob() is adding jobs to the queue

h) ScreenShot Thread is taking screen shots and compressing the image.
i) WriterThread is writing/sending jobs to the client after taking it from Robot Action Queue.
j) Read Action Objects() is receiving/reading jobs from the server and after that passing it to the Execute() to exeute that job.

8. Pass/Fail - Pass if Expected result is same as Actual Result.
9. Bug Number - 0
10. Test result – Pass

V. Discussion

The system is a secure one and is far better than the existing systems on the same topic. This system consists of faster monitoring due to compression of the screenshots which is transferred from one end to the other which is not a feature of any existing systems. Also File transfer and Multichat facility are new features which are not available in existing systems.

It is easy to monitor remote systems one or many at a time by connecting at different ports. User can also view files at remote computer's hard drive and thus it will reduce memory cost at our end. Also video and images captured by the remote computer can be viewed. And after that it is easy to transfer those files from remote computer to our computer.

VI. Conclusion and Future Work

A. Conclusion

This Network service-based system may handle many different types of devices with different kinds of components. This Network based monitoring and control system is platform independent and works across multiple systems having different types of component devices to be monitored. The client machine can be monitored with the help of video and chat messages and further can be controlled with the execution of the desired applications on the client machine by the server machine and even by transferring files.

However, it has to cater for large number of users and simultaneous multiple user access in the future. A simple registration and message forwarding system will not be sufficient for the purpose. Apart from the obvious securities, some kind of conflict resolution is necessary if different users are commanding a machine to different things at the same time. Different conflict resolutions users will need are to be explored as the integrity and safety of the controlled machine will need to be maintained. In a word, industry standard should be used for message passing to provide a service platform that is hardware and software independent.

A. Scope of Future Work

Following on from the requirement that the controlled machine must always be in a safe state, due consideration must be given to the service server in handling the unfortunate, but possible, situation that the communication link between the remote user and the controlled machine is broken suddenly.

There is growing interest in the field of intelligent remote control and monitoring System. Various experimental laboratories and prototypes have been built to demonstrate the feasibility. Ontology knowledge representation in an intelligent Network service-based system, progress has been made in developing more generalized architecture for remote control applications. It could integrate ontology knowledge and service into the system, Although intelligent web service technologies have advanced significantly in recent years, there are still many major issues that are specific to remote control that need to be addressed.

The above work could be easily extended to cover the followings:-

- It can be used for Monitoring and controlling the physical devices attached to the Computer systems either through cable or wireless.
- Integrating the Video chat for more real-time chatting and message transfer will increase the reliability.
- Use of Cloud computing to store the data being monitored and controlled for the future needs.
- The System being monitored can be automated to act on itself if some action is being performed to overcome the problems.

REFERENCES

[1] Vu Van Tan, Dae-Seung Yoo, and Myeong-Jae Yi, “Efficient Web Service based Data Exchange for Control and Monitoring Systems”, International Journal of Information Technology, Vol. 14 No. 1 2008.

[2] Wensheng Liu and Yu Guo, “ The Development of Web Service-Based Remote Control and Monitoring System”, in internation conference proceedings of Computational Intelligence and Software Engineering, CiSE 2009, pp. 1-4, 11-13 Dec. 2009 .

[3] Jian-Wei Lin and Yuan-Cheng Lai, “ A Manageable Web-based Video Streaming System for On-Line Monitoring of Remote Laboratory Experiment ”, IEEE International Instrumentation and Measurement Technology Conference Victoria, Vancouver Island, Canada, May 12-15, 2008.

[4] C. H. Wang, F. Teng, Y. F. Wang, and G. C. Ma, “Web-based remote control service system”, IEEE Trans. International Symposium, vol. 1, pp. 337–341, 9-11 Jun. 2003.

[5] M. H. Hung, K. Y. Chen, S. S. Lin, “Development of a web-services based remote monitoring and control architecture,” In Proceedings of IEEE lmrnational Canfannce on Robotics & Automdion, pp. 1444-1449, April 2004.

[6] Cihan Şahin, and Emine Doğru Bolat, “Development of remote control and monitoring of web-based distributed OPC system”, Journal of Computer Standards & Interfaces, Volume 31, Issue 5, September, 2009.

[7] Han Lin, and Hongji Lin, “Remote Video Monitoring and Controlling System based on Web-Services”, Advanced Materials research, Vols. 383-390, pp. 4439-4445, 2012. (Online available since 2011/Nov/22 at www.sciecntific.net).

[8] R. Kirubashankar, K. Krishnamurthy, J. Indra, B.Vignesh, “Design and Implementation of Web Based Remote Supervisory Control and Information System”, International Journal of Soft Computing and Engineering (IJSCE), Volume-1, Issue-4, September 2011.

[9] Chung-Hsien Kuo, Fang-Ghun Huang, Keng-Liang Wang, Ming-Yih Lee, Huai-Wen Chen, “Development of Internet Based Remote Health and Activity Monitoring Systems for the Elders “,Journal of Medical and Biological Engineering, Vol. 24(1), pp.: 57-67, 2004.

[10] Liu Yang, Linying Jiang , Kun Yue ,Heming Pang, “Design and Implementation of the Lab Remote Monitoring System Based on Web Technology ”, IEEE Conference proceeding of Information Technology and Applications (IFITA), pp. 172 – 175, 16-18 July 2010.

[11] J. W. Overstreet and A. Tzes, “An internet based real-time control engineering laboratory,” , IEEE Control Syst. Mag., Vol. 19, Oct 1999.

[12] Shuang-Hua Yang, “Internet Based Control Systems - Design and Applications “,Springer EBooks. London 2011.

[13] Pasquale Daponte, Claudio De Capua, Annalisa Liccardo, “A technique for remote management of instrumentation based on Web Service”, Proc. IMEKOTC4 13th Int Symp Meas Res and Ind Appl, pp. 687 2004.

[14] Kuk-Se Kim, Chanmo Park, Kyung-Sik Seo, Il-Yong Chung, and Joon Lee. “ZigBee and The UPnP Expansion for Home Network Electrical Appliance Control on the Internet”. In the 9th International Conference on Advanced Communication Technology, pp. 1857-1860, 2007.

[15] Wang, C., Teng, F., Wang, Y. and Ma, G. “Web-based remote control service system”. In IEEE International Symposium on Industrial Electronics, pp 337-341 vol. 331, 2003.

[16] Ha, Y.G., Sohn, J.C. and Cho, Y.J. “Service-oriented integration of networked robots with ubiquitous sensors and devices using the semantic Web services technology”. In IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3947-3952, 2005.

[17] Ha, Y.G., Sohn, J.C. and Cho, Y.J. “ubiHome: An Infrastructure for Ubiquitous Home Network Services”. In IEEE International Symposium on Consumer Electronics, pp. 1-6, 2007.

[18] Ha, Y.G., Sohn, J.C., Cho, Y.J. and Yoon, H. “A robotic service framework supporting automated integration of ubiquitous sensors and devices”, Elsevier, pp. 657-679, 2007.

[19] Helal, S., Mann, W., El-Zabadani, H., King, J., Kaddoura, Y. and Jansen, E. “The Gator Tech Smart House: A Programmable Pervasive Space”. IEEE Computer Society Press, pp. 50-60, 2005.

[20] Xi Guo, and Paul W. H. Chung. “The Architecture of a Web Service-Based Remote Control Service System”, iiWAS2008, November 24–26, 2008.

[21] Chiranji Lal Chowdhary and Chandra Mouli P.V.S.S.R., “Design and Implementation of Secure, Platform-free, and Network-based Remote Controlling and Monitoring System” Proceedings of the IEEE International Conference on Pattern Recognition, Informatics and Medical Engineering, March 21-23, 2012.