# Modular Web-Based Collaboration Platform

Ploypailin Intapong[1], Sittapong Settapat[3], Boonserm Kaewkamnerdpong[2] and Tiranee Achalakul[1]

[1]Department of Computer Engineering, [2]Biological Engineering Program
King Mongkut's University of Technology Thonburi, Thailand
[3]Graduate School of Engineering, Shibaura Institute of Technology, Japan

**Abstract.** Online collaborative systems are popular communication channels as the systems allow people from various disciplines to interact and collaborate with ease. The systems provide communication tools and services that can be integrated on the web; consequently, the systems are more convenient to use and easier to install. Nevertheless, most of the currently available systems are designed according to some specific requirements and cannot be straightforwardly integrated into various applications. This paper provides the design of a new collaborative platform, which is component-based and re-conFigureurable. The platform is called the *Modular Web-based Collaboration* (MWC). MWC shares the same concept as computer supported collaborative work (CSCW) and computer-supported collaborative learning (CSCL), but it provides conFigureurable tools for online collaboration. Each tool module can be integrated into users' web applications freely and easily. This makes collaborative system flexible, adaptable and suitable for online collaboration.

**Keywords:** CSCW / CSCL / Web-based collaboration / Autism community

## 1. Introduction

Collaboration is an activity in which two or more people or organizations interact and work together. Many collaborative systems have been implemented to support comfortable communication from different locations in virtual spaces. They provide workspace similar to the real world environment for interaction among people in virtual environment. People can use virtual tools which simulate communication activities that occur in real-world workspace. This makes collaborative systems be a popular communication channel for groups of people or organizations.

Through observing the literature, we found that there are two methods for using collaborative system. Most collaborative systems need users to install collaborative system beforehand. On the other hand, collaborative systems can also be used via a web browser of a central server with a few additional installations or even no installation required on client's side. Nevertheless, both methods provide virtual tools as a set or package for communication in virtual space.

The implementation of most existing collaborative systems combines overall virtual tools and collaborative functions together. This makes systems more complex and difficult to install; users have to handle the complexity of all functions even though they use only some. Moreover, when some parts of system fail, the main system and other parts also fail. Consequently, the system stability is decreased. In addition, combining everything together makes the systems inflexible to be applied to other works because developers cannot freely mix and match suitable virtual tools to their desired proposes. As a result, developers may need to construct new systems to cover and support all different desired requirements even though all systems hold the same basic functions. For example, collaborative systems designed for meeting focus on features that support communication whereas those designed for distance learning focus on features for presentation and explanation; both of them, however, provide same basic collaborative functions. To do so, it requires skills in programming and computer system technology which may be too difficult for normal users.

To support different requirements and to be integrated into various applications without implementing new systems, functions or tools of collaborative systems should be implemented so that they are independent from each other. Each tool can be set as a module, which can be reconFigureured to suit different proposes. Users can choose modules to be integrated into their application freely and easily without implementation. This enables flexible, adaptable online collaboration.

In this paper, we propose Modular Web-based Collaboration (MWC) to provide such flexibility and adaptability to online collaboration. MWC uses the same concept as Computer Supported Cooperative Work (CSCW) and Computer-Supported Collaborative Learning (CSCL) which provide a virtual space for people to interact and cooperate with each other in some purposes via computers and networks. However, MWC provides communication tools and services for community websites or social networks that require collaboration which are called host applications. The service selection is not limited. Host applications can manage virtual tools independently and can integrate MWC services into their applications easily. MWC can support multiple host applications in the same time. When communication occurs, messages are sent/received between host applications and MWC server. Once MWC services are integrated into a host community website, end users can use collaborative system in their own environment. The interaction using MWC tools and services makes end users feel as all the communication occurs in their own community and they have privacy without having to share collaboration with other groups.

To establish the motivation through existing collaborative systems in the literature, the concept of collaboration and their related works to MWC are discussed in section 2. The design framework of MWC is described in section 3. To demonstrate the usability and flexibility of MWC, a scenario for using MWC on autism community space is presented in section 4. The conclusion of this study is included in section 5.

## 2. Collaboration Concept and Related Works

### 2.1 Web-Based Collaboration

With the rapid growth of Internet applications, many researchers in CSCW [1] and CSCL [2] have focused on the attempts to design system and architecture for distance learning in collaborative virtual environment. In this section, we present the design and implementation of existing web-based collaboration to support various objectives.

A web-based CSCW framework designed by Wang et al [3] is a web-based collaborative workspace using Java3D. This framework allows users to share workspace for design reviewing, production monitoring, remotely controlling and trouble-shooting using Java 3D model. It reduces network traffic and increases the flexibility of remote monitoring. It enables web-based synchronous collaboration with interactive control and quick response. It shows a high potential for web-based real-time distributed applications. Another CSCW framework proposed by Su et al [4] uses internet techniques including client-server and web-services to create an online collaborative design and manufacture; apart from internet techniques, web-enabled environment for collaboration, CAD/CAM, RELSP and distributed product design are employed in [4]. This framework allows users to share AutoCAD design and communicate online through speaking, writing messages on the board and seeing each other on the screen during the collaborative design process.

A research in CSCL designed by Poonam and Bhirud [5] is an interactive web-based system for learning image processing. The developed system consists of four modules including dynamic website, web application with interactive contents, quizzes and assignments. This web-based training integrates java technology platform into the web application based on JSP web application framework. It uses java applets as well as HTML contents or XML for course contents which are essentially run from a web based learning system. Interactive learning object uses MATLAB engine for image processing purposes. This system is suitable for replacing the traditional homework assignments. Other CSCL presented in [6] is the virtual math teams' project (VMT). VMT software was designed for group of 2 to 10 students to discuss mathematics in real time via chat box and whiteboard. This framework integrates synchronous and asynchronous communication together. While the users are communicating by chatting and writing on whiteboards, the systems automatically create a wiki in the portal with the same information as in the chat environment.

### 2.2 Basic Collaborative System's Function

By reviewing existing collaboration systems [7-11], we can classify basic functionalities for collaboration into three groups: communication, presentation, sharing functions. The most simple communication function is messaging. All exist collaborative systems have chat tools for real-time messaging. In addition, some systems also provide communication in terms of voice and video. The presentation function of collaborative systems can support image, audio and video presentations so that explanation can be done more clearly and understandably; whiteboard is an example of presentation tools. The sharing function supports the exchange of information such as files, computer screens and applications. To meet specific needs, additional virtual tools can be developed; for example, in collaboration for meeting a recording function may also be included. In addition, functionalities for collaboration also include user and room managements which support user registration and creating virtual room.

In term of utilization, current collaborative systems can be categorized into two types. On one hand, users need to install the system on the client's side. For some systems, the installation can be complicated. This makes them difficult to use. However, most of them are non-commercial. On the other hand, users can use collaboration systems via provided central servers. Hence, the installation on the client's side is reduced. Users do not need to have programming knowledge. In other words, general users can operate such systems without further implementation. Nevertheless, users need to pay for the service that lacks uniqueness as the user interface of system is fixed; users cannot modify the interface to meet their requirements. Moreover, all virtual tools are usually combined into a set or package which is not flexible to use. Users cannot choose some virtual tools as desired but have to use all virtual tools for different purposes (for example, online collaboration for meetings, selling, training and other online events) included in the same package [11]. In the case that some parts fail, it will affect other parts even the main system.

One example of existing collaborative system is an open-source collaboration system called Openmeetings [10]. This system can support 25 users per meeting room and sustain several meeting rooms at the same time. It has two kinds of meeting rooms including video conference room and audio conference room; both types of meetings can be recorded for further utilization. However, openmeetings combines everything together. This can make the system unstable and difficult to install. Other commercial collaborative system is WebEX [11]. This collaboration provides several sets of virtual tools for supporting various requirements; those tool sets, however, contain different combinations of basic collaborative functions to serve pre-defined purposes. These existing collaborative systems are not flexible to be applied to other works.

With the popularity of using online collaboration, it is beneficial to improve collaborative functions and systems so that online collaboration can be effectively used. In this research, we propose a new design framework that is flexible for different tasks; a tool set can be integrated into any web application easily without having to implement a new system to include another tool on the application.

## 3. The Design Framework

### 3.1 Conceptual Design

Modular Web-based Collaboration (MWC) is a platform which can reconFigureure virtual environment for flexible usage as desired without implementing a new system. MWC designs the development virtual tools or functions independently which can be called as modules. Each module in

MWC platform can be integrated into any host applications that want to add virtual tools as parts of them.

MWC provides collaboration tools in virtual space for host application. Host application's clients can use collaboration tools with their host application directly without sharing virtual space with others. Host applications can integrate MWC tools and MWC services into their applications easily without the need of writing program all over again. Firstly, host applications must register to verify their identity before using MWC tools and services via MWC website. Next, they can freely select communication tools that they want. In other words, all tools are independent so host applications can mix and match those tools suite to their applications. After registration and tool selection, host applications receive Code ID for authentication and get their selected virtual tool as an object as well as a script for calling. In the case that host applications select more than one virtual tool, host applications will receive the same number of objects as the number of selected tools; these tools are not combined together. Host applications can put these objects in their web server and insert the scripts in the source code of web page that displays collaborative system. MWC provides communication tools and services for host application including chat box, audio/video conferencing, whiteboard and desktop sharing. The other functions such as user and room management must be provided by host application. After MWC installation is done, host application's clients are allowed to interact with their community in customized virtual environment through their host application directly.

With the proposed concept, the benefits of MWC platform include the following:

- *Privacy*: MWC allows users to use collaborative functions within the host application's environment. This gives users a sense of identity. They do not need to share collaborative systems with people outside the community and also have privacy within their own community.
- *Flexibility*: The MWC-based system can be designed to suit the usage of the community. When using the system within the community, users can learn only collaborative functions required for their community. Hence, it is easy to use and suitable for general users with no sophisticated knowledge on technology.

However, as communication tools are designed to be independent of each other, it requires time for installing each of the selected tools. In addition, cross interaction between two or more tools is restricted; for example, users are not allowed to watch a video in chat box. As experienced in current collaborative systems, the number of users using MWC is limited by the performance of the hardware. Nevertheless, server farm and load balancing can be introduced to increase the capacity of the systems.

### 3.2 Architecture Design

As MWC provides communication tools and services to be integrated into host applications so that users feel as if they use these tools through their applications, the interaction enabling such characteristic involves clients, host applications and MWC. Clients are subscribers of host applications and MWC. They directly contact with host applications and indirectly communicate with MWC. Host applications are both providers and subscribers. They use services from MWC to provide services for their clients. MWC is the providers of all communication tools and services. The architecture overview of MWC platform is illustrated in Figure 1.

The architecture shown in Figure 1 can be referred as client-server architecture, which is software architecture describing communication between client and server. In this case, clients interact with host applications (C-H) and host applications register with MWC (H-MWC). Before host applications can provide MWC services to clients, H-MWC interaction is required; host applications must subscribe with MWC for service registration. Developer registers via MWC web interface and selects communication tools and services. At C-H interaction, users access website and interact with their host interface. When a client communicates with other clients on the same host applications, communication sessions between client application and host servers are initiated. Upon incoming requests, host servers act as an agency forwarding communication messages to MWC. Then, MWC responds and sends the message back to the host application which forwards to clients via web interface. Note that web interface and workflow design on each host application can be different.

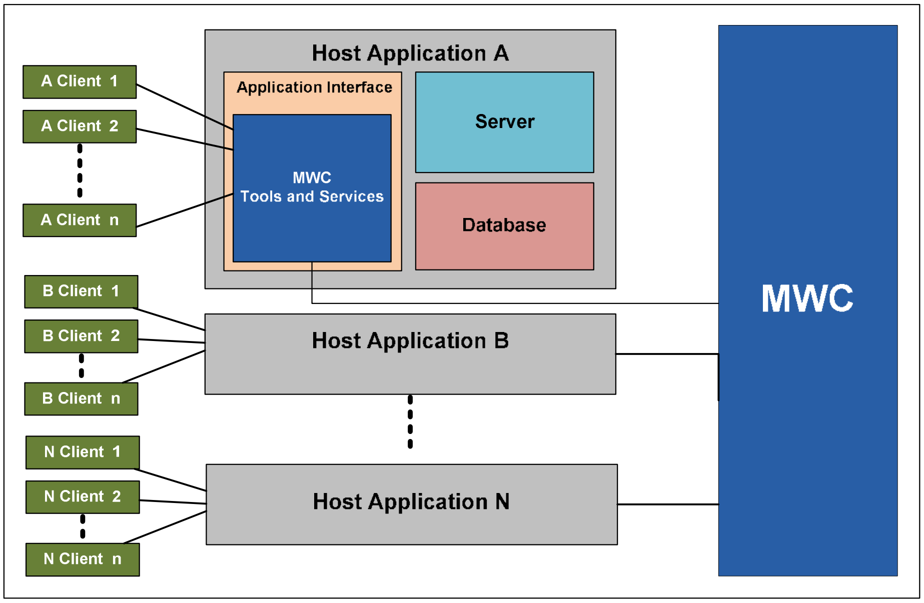


**Figure. 1.** Architecture overview of MWC platform

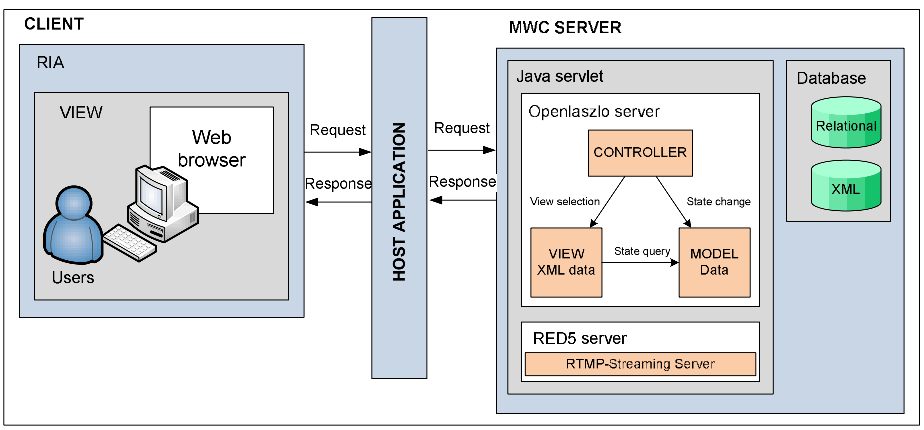


**Figure. 2.** Model-View-Control (MVC) architecture for MWC platform

MWC platform focusing on clients using MWC communication tools and services via agency (host application) can be elucidated in Model-View-Control (MVC) architecture presented in Figure 2. MVC architecture isolates the domain into three entities: the model (managing information), view (presenting the model) and controller (handling responses). The MWC communication tools are displayed in host application's interface. When users interact with web interface, the controller handles the event and converts into appropriate reaction. Then, the controller notifies the model of reaction and passes to view to display in web interface. This occurs in client's side with RIA technology. If it is unable to form model of reaction at client side, model will send the request to server and, then, server responds.

To provide some useful information for implementing collaborative systems based on MWC platform, we observe different technologies providing video/audio streaming used for implementing current collaborative systems and conclude the potentials as follows: *Adobe Flash* is a multimedia platform which already has libraries to support media streaming, so it helps reduce time for programming. It is positioned as a tool for RIAs which is in currently the top three platforms (apart from *Java Fx* and *Microsoft Siverlight*). Flash server transfers message communication, video stream and audio stream from clients to other clients. FMS (Adobe Flash Media Server) is the most mature one that supports mass scale of applications; it is, however, expensive. WMS (Wowza Media Server) is flash server that is suitable for application of primary and media enterprises for which cost and stability are most concerned. Red5 is an optimizing open source flash server written in Java. Adobe Flex and Openlaszlo are platforms for developing RIAs based on Adobe Flash. Adobe Flex is written in action script, whereas Openlazlo is deployed as traditional Java servlets. Laszlo applications are compiled and returned to the browser dynamically.

### 3.3 Protocol Design

As MWC supports multi host applications, it is essential that MWC knows from where the message was sent and to where it sends the response. The detailed protocol of MWC and supportive technology are described and illustrated in Figure 3. After host applications have already registered and integrated communication tools, communication between users and MWC occurs when a collaborative room is created. The collaborative room, a channel of one or more clients, is implicitly created when the first client joins it. First client in each channel becomes a channel operator.

The initial message for communication is sent to MWC server to open connection. It consists of three main parts including code ID, channel ID and user ID. Code ID refers to host application. Chanel ID refers to group of users such as channel code or channel name. User ID refers to client such as IP client or username. When other users join existing collaborative room, the host application sends user message to notify the server.

Messages used for communication among users consist of user ID and chat message. User can send a one-to-one message to communicate with another client; other clients do not receive the message. For one-to-one communication, the information specifying the receiving client must be added into a message. In one-to-many communication, the message is sent to all clients in channel.

If MWC server cannot find the code ID, MWC must ignore the initial message of that communication. MWC server opens connection with host application only when code ID is correct. Then, MWC server performs connection thread with initial communication information. MWC server adds users to connection thread when they join the collaborative room. It removes users from connection thread when users leave collaborative room and also closes the connection when the last user leaves the room.

When MWC server receives a message, it knows the user ID from the message and refers to corresponding thread connection. It broadcasts to all clients (or sends to single user) the chat message in the thread connection. The messages are sent according to queue as FIFO (First In First Out).

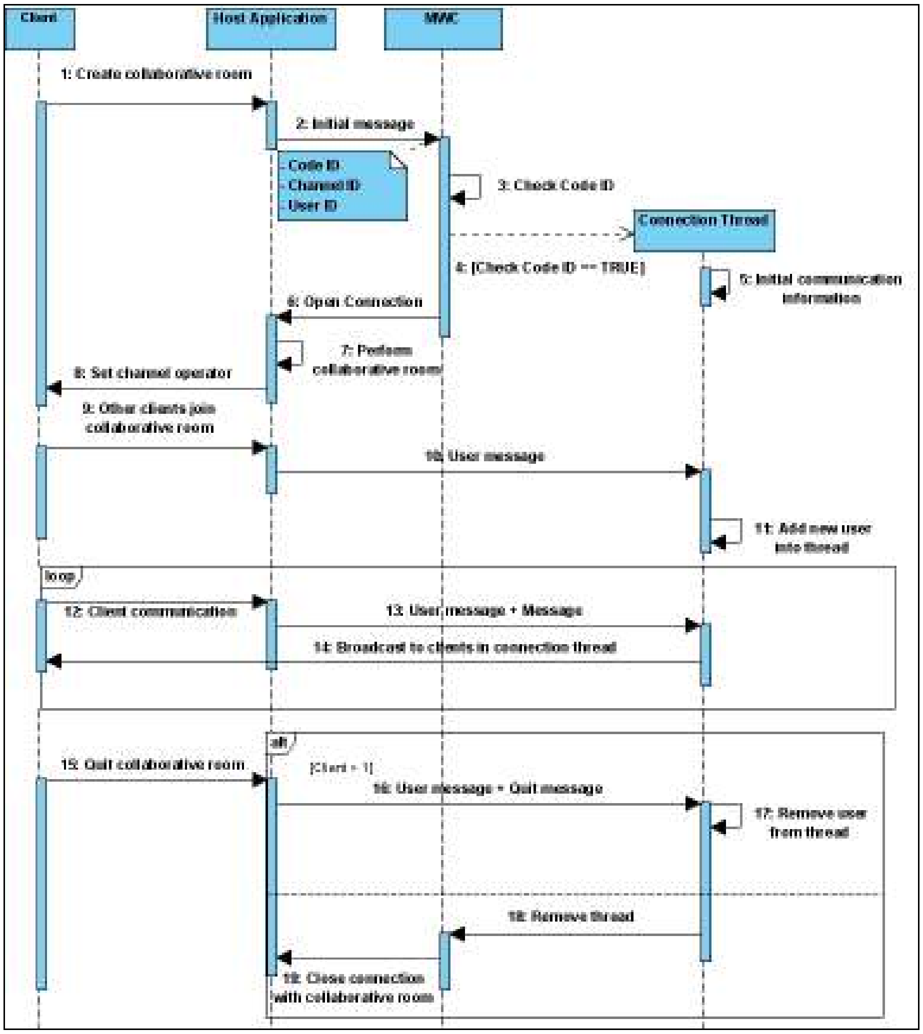


**Figure. 3.** Communication diagram for MWC platform

### 3.4 Preliminary Implementation of MWC

Based on the design of MWC framework, a preliminary implementation is conducted to provide detail information on using MWC framework. In this preliminary implementation, MWC is developed based on Java platform. On the server side, JSP technology and Java Servlet are used for web-based application. RED5 is also employed as a flash server for transferring communication messages and video/audio streams with clients operating through flash applications.

For this preliminary implementation, a chat module is developed. There are two independent parts involved in the chat module: chat box and user and room management. Applications are implemented on both the server and client sides. To define basic structure in Java application on RED5 server, we firstly import class *ApplicationAdapter*, which is base class for application on RED5, and then define java class extending from *ApplicationAdapter* class. Start() and Stop() methods are created for starting and stopping the application respectively. The structure of source code for preparing to connect with RED5 server is the following:

```
import org.red5.server.adapter.ApplicationAdapter;
public class Application extends ApplicationAdapter
```

```
{
    public boolean start () {

    }
    public void stop () {

    }
}
```

For establishing client connections, *IConnection* class is imported and Connect() and Disconnect() methods are created. Connect() method will be executed when the first client connects to RED5 server (the client creates a chat room), whereas Disconnect() method will be executed for disconnecting client connection to RED5 server.

```
import org.red5.api.IConnection;

public Boolean Connect(IConnection con, Object[] params) {

}
public Boolean Disconnect(IConnection con, Object[] params) {

 }
```

The above methods get a parameter called *IScope*, which is a statefull object shared between groups of clients that are connected to the same context path in hierarchical way. For this preliminary chat module, *IScope* will create host application scope and room scope when the first client of host application connect to the server; for another client creating a new chat room, *IScope* will create another room scope. As a result, MWC can identify the owners of receiving messages so that MWC could perform further responses accordingly. Apart from these methods, there are some methods for other functions such as methods for creating room or joining room, methods for security management, and etc.

At the client side, flash applications are employed for connecting to RED5 server. The applications require flash class called *NetConnection* to create connection as stream pipe between client applications and flash server applications. In the chat module, messages and user information are defined as *sharedobject* which can store data on flash server for clients to retrieve. A remote *sharedobject* can share data among clients in real time.

The connection can be started by defining a new connection. To ensure the establishment of the new connection, the status of connection is checked before further processing. If for any reasons the connection cannot be formed, event listener will be removed from chat with *removeEventListener* method. The action script for creating new connection is the following:

```
var chatNC:NetConnection = new connection();
```

In the case of successful connection, *SharedObject.getRemote()* method is used to gather the *sharedobject* containing user information and message dialog of this connection and to send the *sharedobject* to the server. Then, the flash application will use connect() method to establish a connection to the server by using the following action script:

```
SharedObject.getRemote();
chatNC.connect();
```

In terms of communications between clients, host applications and MWC, a host application communicates with MWC during registration and installation processes through HTTP protocol. The host application, then, receives an authentication code for verifying with MWC. Before a client connects to MWC (when the client wants to create or join room), the client will request permission from the host application which it has session with. The host application transmits an identification key to the client. Then, the client submits the initial connection to RED5 server via RTMP (Real Time Message Protocol) as shown in Figure 4. The design for authentication between client and host application is based on each host application, but it must be according to the standards provided by MWC.

To prevent unauthorized usage without registration according to the proposed concept of privacy, send/receive messages are encrypted with some keys that are known only to the rightful host application. The unique key such as client session ID is used for handshaking before exchanging the key between the flash application and host server. Messages are encrypted and sent to MWC server without decryption. The role of MWC is to receive and forward messages to their destinations. Only clients relevant to the communication messages can decrypt the received messages.

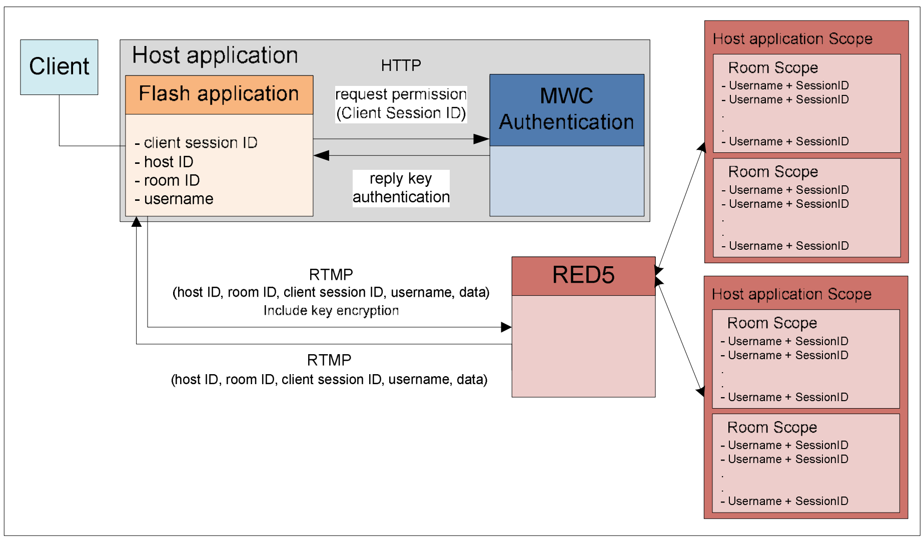


Figure.4 Data connection

After implementing the preliminary chat module, we found that RED5 documentations are hard to find and do not explain much in details. This requires time to learn how to use and experiment. Moreover, to maintain the privacy within the communication, the security management must be considered. For other modules i.e. audio/video conferencing modules, different protocols for send/receive data may be applied.

## 4. Scenario: Autism Community Space based on MWC

Autism is a disorder concerning abnormalities of brains. It requires specialists to diagnose; however, there are only a few in Thailand. The learning of this medical specialty requires clinical study, but most students are doctors in local hospitals outside the cities; it makes this specialized training almost impossible. Interaction in autism community can bring about benefits of sharing information. Nevertheless, in some cultures having child with abnormalities can be seen as embarrassment to families; hence, it is better to have privacy in using online collaborative systems in their own community. Apart from that, the flexibility of collaborative systems which is easy enough for general users to operate is appreciated as communities' members may not be so skillful in computer technology to utilize an advanced system to its maximum advantage. Therefore, integrating MWC tools in community's host applications seems suitable for autism community. To demonstrate the utilization of

MWC platform, a sample scenario of online collaboration in autism community in Thailand is discussed. Such online collaboration can be beneficial as a distance learning tool for local doctors and a consultation tool for teachers and parents.

**Sample scenario for using autism community space based on MWC:**
**John,** the director of autism community in Thailand, wants to create community website which has a virtual space for communicating and sharing information from different locations. He wants this to be easy to use; even someone who has no IT knowledge should be able to use this collaborative tools on the website. Moreover, he wants no links to other websites in the community website.

**Bob,** a web administrator, is assigned for this task. He creates autism community website with need collaboration tools so he accesses to MWC website and registers. After registration, he is allowed to select communications tools such as chat tools, audio/video conference tools that he can customize to suitable his purpose. Then, he receives code ID for authentication with MWC communication objects and scripts for installation in autism community website. He puts the objects in the community's web server and inserts scripts in each collaboration room design which has different virtual environments such as conferencing room, presentation room and chat room

**Alice**, a pediatric doctor in local hospital, wants to consult with **Sara** who is an experienced specialist of autism center in Bangkok about clinical study of her autism patients, but it is inconvenient and too expensive to travel to Bangkok. She reserves a private conferencing room and invites Sara through autism community website. She uses video conferencing tool to communicate with Sara, uses video playing tool to show video of her patients, and also uses file sharing tool to share treatment history.

**Clare**, a parent of an autism patient, has doubts on her child's behavior. Sara suggests her to join autism community website. Clare creates a public chat room on child behavior topic. She shares information with other parents and seeks advices for effective treatments. Sometimes, Sara joins a chat room as an audience to observe discussions. Sara always follows information sharing in autism community website so that she could gain interesting points for further study. When finding useful information, she reserves a presentation room for training on those topics. Other members can be audiences. Sara controls audio conferencing tool for receiving questions and uses whiteboard tool to assist her explanation for better understanding.

## 5. Conclusion

In this paper, we propose a design of a new collaboration platform which is highly flexible and can be applied to serve different proposes without having to implement a new system for each additional specific requirement. MWC provides conFigureurable tools for communication in virtual space which can be integrated into community own appropriate virtual environment easily. In sum, MWC platform provides privacy within the community and flexibility to collaborative systems which enables easy-to-use online collaboration. The preliminary implementation based on MWC framework is discussed in section 3.4. MWC has a potential to be effectively applied to fulfill different usages for autism community as demonstrated in section 4. Members can communicate and share information from different locations through autism community website easily without advanced IT knowledge. New virtual tools or versions of existing virtual tools can be easily added. The benefit of MWC can also be extended to other communities.

**Acknowledgement:** This research is a part of the project supported by The National Science and Technology Development Agency of Thailand (P-09-00346)

## Reference

1. Kevin L.: Computer-Supported Cooperative Work. In: Encyclopedia of Library and Information Science , pp.666-677 (2003)
2. David W. and Katherine A., “Why all CSL is CL: Distributed Mind and the Future of Computer Supported Collaborative Learning”, Proceedings of the 2005 conference on Computer support for collaborative learning: learning 2005: the next 10 years!, Taipei, Taiwan, pp. 592-601 (2005)
3. L. Wang, B. Wong, W. Shen and S. Lang.: A Web-based Collaborative Workspace Using Java 3D. In: Computer Supported Cooperative Work in Design, The sixth International Conference London, Ont, pp.77-82 (2001)
4. D.Su, J. Li, Y. Xiong and Y. Zheng.: Application of Internet Techniques into Online Collaborative Design and Manufacture. In: Computer Supported Cooperative Work in Design, Proceedings of the Ninth International Conference, Vol. 2, pp.655-660 (2005)
5. Poonam S, T. and Bhirud, S.G.: Interactive Web Based Learning: Image Processing. In: Application of Information and Communication Technologies, 2009 AICT 2009. International Conference on, pp. 1-4 (2009)
6. Gerry S. and Murat Perit C.: Integrating Synchronous and Asynchronous Support for Group Cognition in Online Collaborative Learning. In: Proceedings of the 8th international conference on International conference for the learning science, Utrecht, The Netherlands, Vol. 2, pp.351-358 (2008)
7. Xiao-min W., Peng-cheng W. and Jin-dong X., “A New Design of Remote Diagnosis System for Medical Images”, Management and Service Science, 2009. MASS’09, pp. 1-3 (2009)
8. Norihiro K. et al., “Construction Methodology for a Remote Ultrasound Diagnostic System”, IEEE Transactions on Robotics, Vol. 25, Issue 3 , pp.522-538
9. Wenhua H. et al., “Computer Supported Cooperative Work (CSCW) for Telemedicine”, Proceedings of the 2007 11th International Conference on Computer Supported Cooperative Work in Design, pp. 1063-1065 (2007)
10. Openmeetings [Online], Available: http://code.google.com/p/openmeetings [2010, February 20]
11. WebEx [Online], Available: http://www.webex.com [2010, February 20]