ABSTRACT

This project aims to design and develop a Computer Supported Cooperative Work (CSCW) support as part of the Octopus middleware for network applications to communicate and conduct group session with ease. The CSCW also provides a list of generic facilities and services that the application writers might utilize for the development of complex applications.

1. INTRODUCTION

With the widespread availability of the internet, an increasing number of people are using the internet for group activities. Among the numerous group activities available on the internet, chat and multiplayer gaming remains one of the most prevalent activities carried out online. Despite its widespread use there are various communication platforms for these chat and gaming applications which results in incompatibility and inconveniences for internet users[1]. Furthermore, with the existence of dissimilar communication platforms, applications developed for one communication platform could not be ported to work on another communication platform easily. The competing communication platforms simply hinder the adoption of network services[2].

The CSCW layer provides a basic framework for collaboration software to perform network functions. The framework allows the creation of groups or services easily on the network. Users can also be added and removed from the groups spontaneously through the use of the CSCW framework. Above and beyond providing simple group functionality the CSCW layer also provides other services such as encryption, authentication, access controls and floor control which are essential components in any group collaboration activity that takes place on a network. Besides providing utilities in the CSCW layer for applications to use, it also provides several forms of generic applications - mainly for supporting audio, video and data transmission.

2. TECHNOLOGIES EMPLOYED IN THE CSCW FRAMEWORK

The CSCW framework employs several technologies for its implementation. In order to ensure that a wide variety of computing platforms and operating systems are able to benefit from the CSCW framework, the middleware has been programmed in Java. Other Java related technologies that has been employed includes Jini which aids registration of groups and services that have been created on the network. Jini would help to keep track of the services that have been created. At the same time Jini helps other applications on the network to find and match the services that they might require. Underlying Jini would be Remote Method Invocation (RMI) which basically allows methods in a
remote machine to be executed by other machines in the network. Thus Jini and RMI would work hand in hand to provide an effective solution to provide services (e.g. chat, multiplayer gaming etc) in a network running on distributed machines.

Another Java technology that has been employed in the CSCW middleware would be Java Media Framework (JMF). JMF is used to provide audio and video codec that is required in the middleware. This would facilitate the efficient transmission of audio and video mediums over a network with finite bandwidth.

Finally, the most important component in the design of the CSCW layer would be the Octopus component. Octopus forms the foundation of the CSCW middleware and provides the basic features in the CSCW component.

3. THE OCTOPUS LAYER

The Octopus layer which lies beneath the CSCW layer (Fig 1) provides the communication abilities for the CSCW entities. Octopus is responsible for the creation of streams and flows which are communication channels for computers on the network to transfer information. A stream can consist of many flows which are further broken down into several distinct types of flows – Data Flow, Audio Flow and Video Flow. A flow is a single communication channel that a computer on the network could use to communicate with other computers. The reason for separating the different flows into audio, video and data is due to the special nature of each data type. (i.e. Octopus could change the codec for audio and video should the available bandwidth decreases, thus assuring some semblance of quality of service)[3].

Due to the evolving nature of the Octopus layer, as it is an ongoing research work, and the complexity of the Octopus, a CSCW layer has been created above the existing Octopus layer to allow programmers to use the group services and the features provided in Octopus more easily.

4. DESIGN OF THE CSCW LAYER

Although Octopus does provide the ability to set up channels of communications between computers in a network, it does not however provide functionalities that would are commonly required in a network application. The CSCW layer thus aims to overcome this short fall through the implementation of several features. The CSCW layer (fig. 1) is largely separated into two distinct components – CSCW Core and CSCW utilities.

<table>
<thead>
<tr>
<th>CSCW Core</th>
<th>CSCW Utilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>MultimediaComponent</td>
<td>Authentication</td>
</tr>
<tr>
<td>VideoComponent</td>
<td>Encryption</td>
</tr>
<tr>
<td>AudioComponent</td>
<td>Access Control</td>
</tr>
<tr>
<td>DataComponent</td>
<td>Floor Control</td>
</tr>
</tbody>
</table>

Octopus Layer

Fig 1. CSCW Framework
4.1 CSCW Core

The CSCW Core layer consists of the main components that are necessary for any network applications. Its main task is to facilitate communication between computers located in the network. The core layer is also responsible for the creation of groups of services that computers could participate in. There are five main components in the CSCW core layer, AudioComponent, VideoComponent, DataComponent, MultimediaComponent and ControlComponent.

The components in the CSCW Core are mainly abstraction of the different flows in the Octopus layer. However, they have been design for easy use so that the programmer does not need to understand the complex workings of Octopus to benefit from its features. More importantly, the Octopus layer could be susceptible to changes as it is an ongoing research project. With the CSCW layer in place, any applications created using the CSCW layer would continue to function since it is an abstraction of the Octopus layer.

4.2 CSCW Utilities

Besides providing basic communication and group creation components in the CSCW Core layer, several utilities that would be commonly used in network applications are catered for in the CSCW Utilities layer. These utilities address mainly provide a secure form of communication for the different core components communicating with each other which Octopus currently does not provide.

4.2.1 Authentication

Authentication would help to verify if the different users that are joining the group are who they claim to be. In order to ensure the identity of the user, an asymmetrical encryption system would be use which would involve two keys, namely private and public key. Basically the private key would be held by the user and the public key would be held by in a centralized database for other users to retrieve. The public key would be used to encrypt a message which would be sent to the user. Only the intended user would be able to retrieve the message user his private key. Likewise, to confirm that the receiver has got the message, he would use the sender’s public key to encrypt the message and send it back to the sender.

4.2.2 Encryption

After the authentication process is over, a symmetrical encryption system would be employed. This is because the main reason being a symmetrical encryption system would be much more efficient compared to an asymmetrical encryption system. Secondly, it allows all the users in the same group to use the same encryption key. Furthermore the encryption system also allows the group to change the encryption key to ensure that the group session is secure and less prone to eavesdropping by undesirable users in the network.

4.2.3 Access Control

Access Control provides permission for the user. A user might be able to authenticate his identity. However, the user might not have access to a particular service or group. Thus access
control would check the user’s permission settings before the application provides the user to participate in the group activity.

4.2.4 Floor Control

Floor control provides some form of permission to the users in the group. Basically it would decide, base on the policy chosen, which users are able to communicate in the group. For example, in a chat application which has round robin as the chosen floor control policy, each users in the chat application would be given a certain amount of time to send their text message.

<table>
<thead>
<tr>
<th>Network Layer</th>
<th>Octopus Layer</th>
<th>CSCW Layer</th>
</tr>
</thead>
</table>

Fig 2. Illustration of the CSCW Layer in a Network Application.

6. APPLICATION DEVELOPED ON THE CSCW LAYER

In order to demonstrate the capabilities of the CSCW layer an e-learning application, at the application layer (Fig 2), has been designed. The e-learning application consists of a shared workspace[4], an ftp component, a powerpoint viewing component, a chat component and a remote desktop component. The components are designed in a modular manner and bundled with the CSCW layer so that future applications could make use of them (e.g. a multiplayer game could use the chat component to allow players to communicate and the ftp component to distribute files).

7. PROGRESS OF THE CSCW LAYER

Presently, the CSCW layer is almost completed with regards to the abstraction the features from Octopus and is undergoing further refinements to make it easier for the CSCW layer to be used. However, due to the continual development of the Octopus layer, the CSCW layer will continue to evolve yet maintaining backward compatibility with the applications designed based on the existing CSCW layer.

5. CONCLUSION

With the provision of the CSCW Core and Utility layer in the CSCW middleware, network application could be easily written. This is because the methods of communication have been abstracted from the programmer who does not have to go through a steep learning curve to benefit from the features provided by Octopus. Thus regardless of the physical medium used in the network layer (i.e. wireless network or wired network) used the programmer has the flexibility to create the application he desires without any knowledge of network programming. With the continuous progress made to the Octopus layer
6. ACKNOWLEDGEMENTS

I would like to thank Chaiwat Siriyuenyong who provided technical assistance and advises with the designing of the CSCW framework to ensure it is extensible and many others who provided technical assistances during design of the framework and its application components. Lastly I would like to thank Dr. Pung who made this work possible with the generous guidance and valuable insights provided.

7. REFERENCES


