University of Ibadan
Journal of Science and Logics in ICT Research

Collaborate^{IT}: Development of an Architectural Framework for a Collaborative System in an Academic Environment

[1]Ayeni. J. A., [2]Olabiyisi S. O., [3]Omidiora E. O. and [4]Ganiyu R. A.

[1]*Department of Computer Sciences, Ajayi Crowther University, Oyo*
.[2]*Department of Computer and Cybersecurity, Ladoke Akintola University of Technology, Ogbomoso*
[3&4]Department of Computer Engineering, Ladoke Akintola University of Technology, Ogbomoso
[1]*ja.ayeni@acu.edu.ng*, [2]*soolabiyis@lautech.edu.ng*,[3]*eoomidiora@lautech.edu.ng*, *4raganiyu@lautech.edu.ng*

Abstract

The current state of Information and Communication Systems coupled with recent advances in software tools have opened up the opportunities that have enhanced the development of computer collaborative and conference systems. Computer collaborative systems allow users and participants to collaborate and share information/resources over the network from geographically dispersed locations. The floor of a collaborative system is made up of the users currently participating in a collaborative session. Floor control is the mechanism for managing which user(s) is (are) allowed to speak/text/access resources (to) the participants during a session. Floor control as an essential ingredient in collaboration, is lacking in most of the current systems and has been identified as the most essential aspect of result oriented collaboration and resource sharing in several studies. This paper addresses three floor control policies; free floor, Queue and Bids policies. The architectural framework for Collaborate^{IT}; a moderator-based Desktop Collaborative system capable of managing a seminar with pre-defined participants and roles is presented. The proposed collaborative system architectural framework allows participants seamless access to the floor with strict compliance with the floor control policy and ensuring system state consistency during sessions.

Keywords*: Floor Control Policy, Computer Collaborative Session, System State consistency, Free Floor*

1. INTRODUCTION

Computer collaborative systems also known as computer conferencing or desktop conferencing allow users to interact/share resources and results of research work over geographical dispersed locations using the network technology. The advancement of computer networking and the ever increasing need for the advancement of the virtual collaborations or conferences, researches have tended towards Remove 'in' and comma after the reference bracket providing solutions to the problems associated with these systems. Kangseok *et al.*[1], presented collaboration as about interaction among people and between people and resources.

According to Kangseok *et al.* [1] "Conference Collaborative Systems typically provide a group of users with asset of well-defined interactions to access applications and resources, and enabling communications between them". Stefan Werner and Pascal [2] define desktop conferencing as a combination of real-time-computer conferencing and teleconferencing where participants involved are distributed across several meeting rooms.

Computer conferencing is a way to support and structure network communication that enables individuals to share their know-how and retrace collective knowledge building. Communication and Information Technology have the potential to make collaboration independent from the proximity of time and place. These technologies commonly referred to as collaborative System's tools give new meaning to communication and information exchange.

Collaborative System's tools will fundamentally change the framework and concept of productive and innovative collaborative work [3].The scope of this study is limited to the implementation and use of Collaborate^{IT} in an academic environment. This paper is divided into Five (5) Sections as follows; Section one (1) introduces the concept of the topic. Section two (2) elucidates a review of related works on floor control in Collaborative and

Ayeni, J. A., Olabiyisi, S. O., Omidiora, O. A. and Ganiyu, R. A. (2021). Collaborate^{IT}: Development of an Architectural Framework for a Collaborative System in an Academic Environment. *University of Ibadan Journal of Science and Logics in ICT Research (UIJSLICTR),* Vol. 7 No. 1, pp. 48 -54

Teleconferencing systems. The architectural framework of the system is presented in Section three (3), Methodology while the implementation strategy is presented in part four (4). Finally, discussions and conclusion are presented in Section five (5).

2. Review of Related Works

Many different kinds of floor control have been proposed, but few have been evaluated in studies. Greenbag [5] discussed a number of different floor control mechanisms, and concluded, "surprisingly, there had been no attempt to evaluate these different methods in existing shared view systems". These methods were addressed in Ruibiao [4], Greenbag [5], Dommel and Garcia [6] and Dommel and Garcia [7] with several other issues related to floor control and solutions to how conference or collaborative tasks sessions could be well managed proffered.

Myers and other researchers [8] concluded in their work that no particular floor control strategy stood out as superior to others because user preferences and opinions differ about which of the policies work best**.** An XGSP-Floor policy as developed by Kangseok *et al.* [1] (floor control policy), defines how the participants in synchronous collaboration session request a floor for the use of a collaborative application, and how the floor for the use of the application is assigned and released when the participants share the synchronous collaboration application. In the design of the XGSP-FLOOR protocol,

Kangseok *et al.* [1] identified "two distinct relaxed and strict floor control mechanisms with the following considerations; possibility of concurrent activities among the growing number of participants in a heterogeneous collaboration environment, the effects of network transactions in optimistic mechanism versus the waiting time for turn-taking against non-optimistic mechanism and the intermittent network disconnection of cell phone devices with the complexity for managing them on optimistic and non-optimistic mechanism in synchronous collaboration domain.

In the PASSENGER, (a desktop Conferencing/Collaborative System), the floor control is implemented on the server side and handles the access to the floor, shared resources and coordinates the course of communication through an administration of different kinds of permissions, e.g. permissions to speak, permissions to alter the documents [2]. Floor control approaches usually are mainly technically or socially oriented that combines the advantages of technical and social floor passing methods while maintaining a permission list [2].

3. Theoretical Perspective

Rainer and Cegielski [9] defined teleconference as the use of electronic communication that allows two or more people at different locations to hold a simultaneous conference. Teleconferencing systems have varying degree of interactivity – capability to respond back to the user [10]. The term teleconferencing comprised of a range of different media, including video-conferencing, audio-conferencing and computer conferencing [11]. The common types of teleconferencing systems are: text, audio, video and mixed (audio, video and/or text).

The basic components of a collaborative system (computer teleconferencing) are [13, 14, and 15]:

a) The user interface
b) The distribution mechanism
c) The floor control scheme.

These components are explained as follows:

i. User interface

This is the aspect of presentation of the conference to the user. This consists of the following [16, 17]:

a. A user window, which accepts input when the user is the current holder of the floor.
b. A number of other-participant windows, which display their input.
c. A control window to show join/leave and other participant/conference information such as required.

ii. Distribution mechanism

The advent of system based technology has provided collaborative systems with well-designed communication mechanism that best matched the communication needs of a teleconferencing system and an appropriate distribution mechanism [11]. The system communication mechanism of a teleconferencing system is often characterized by "one-to-many" or "many -to-many" communication channels [12].

The general conference model is that of a uni -directional channel from each user to every other user. A user is a possible floor holder (speaker) and also a listener [13]. They often make use of the underlying multicast or broadcast mechanisms of the operating system environment. The main problem that the distribution mechanism system addresses is the idea of a user holding on to the floor selfishly as to impede the transfer of the floor to other participants in a collaborative system such as the teleconference system. Mendez *et al.* [18] proposed a distribution mechanism based on the game theory and implements a system that fully addressed this problem.

iii. Floor control

This is the mechanism for managing which user(s) is (are) allowed to speak to which listeners. The floor is made up of the users currently participating in the conference [19]. It requires some out of band channel to 'make bids' to be placed in a queue of speakers, and requires some out of bound channel to turn on and off some 'value' on the connection from the current holder of the floor to some other participant(s). Generally, it is not presumed that the user/participant at the end of an open channel is necessarily an active participant.

4. Methodology

The architecture of the desktop collaboration system is based on the client-server technique as implemented by the X system (figure 1). It could be recalled that X-architecture facilitates the implementation of client-server applications as X-system is based on the same client-server architecture. The desktop collaboration server is implemented as an X-server and provides the resources and connections to the desktop collaboration client(s). This forms the distribution mechanism function of the desktop collaboration server and it is implemented by creating the collaboration with the starter window and waiting for other client's (participants) requests. The desktop collaboration server (a veritable X-server) is created once and other collaboration clients (X-clients) are subsequently created upon request by would-be participants to join an on-going seminar or presentation. Each participant's window is a collaboration client window and sends requests to the server during a desktop collaboration session. Although, the desktop collaboration server maintains communication links to each of the collaboration client in a collaboration session, only one of such links is activated at a time (grant) depending on the floor control protocols and policies' decision.

a. Description of the collaborative session scenario

The collaborative session scenario could be depicted as follows (see Figure 2):

i) A registered user (moderator – departmental seminar coordinator) starts the session and invites others (PG student – Presenter, supervisor and co-supervisors and other PG students) to join (server)
ii) The invited users use some facilities to join the session *(*server).
iii) The coordinator moderates the floor based on the following floor control policy:
    a. The Presenter – explicitly granted the floor (Starting policy)
    b. The participating Post-Graduate Students, Lecturers and others granted the floor – (Bidding + Implicit)
    c. Supervisor and co-supervisor – Make explicit request for the floor using a function key
    d. The moderator ends the session by terminating the server after grabbing the floor and performing a review function.

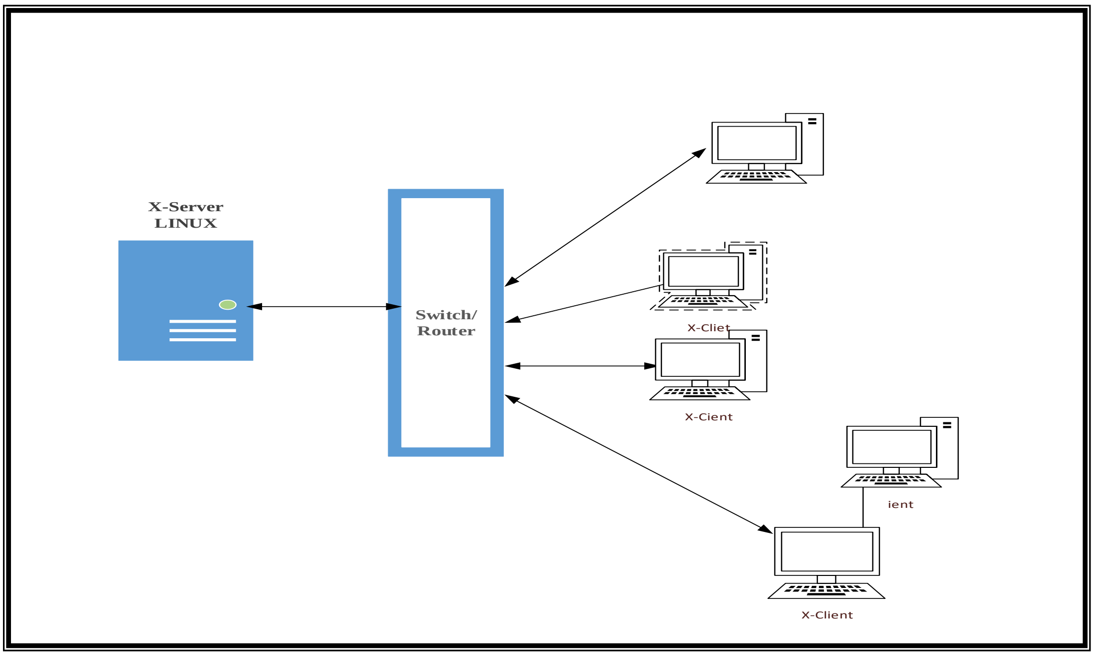

Figure 1: Typical X-Server/X-Client Configuration

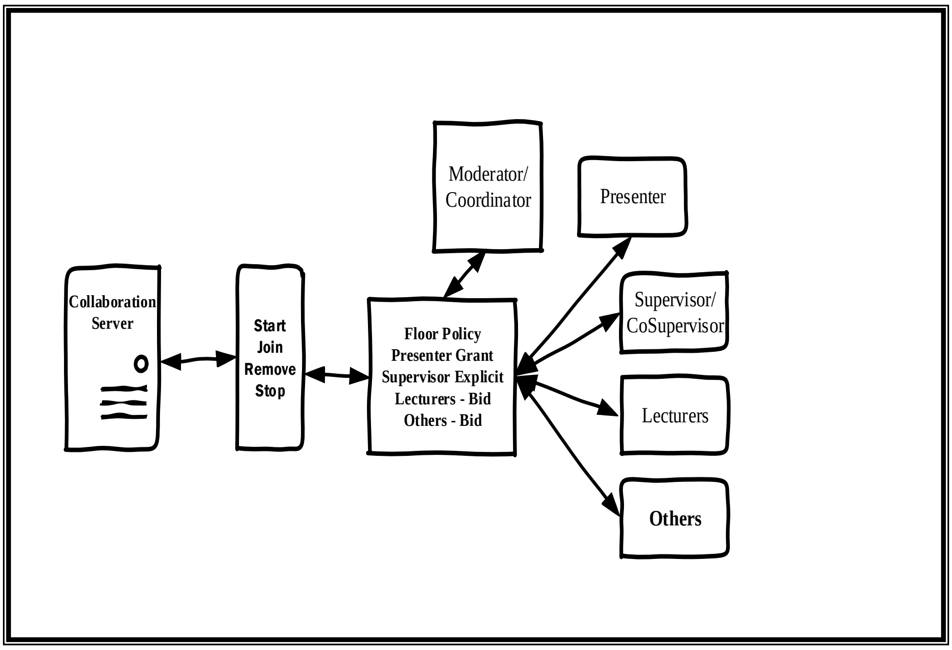

Figure 2: Schematic Representation of Collaborative Session

5. Implementation Strategy

The strategy for the implementation of the Collaborate^{IT} as depicted in Figure 2 consists of breaking down the components into modules for implementation. The conference start module (a major component) of the system is illustrated with the flowchart as presented in Figure 3. The corresponding functions are also presented in Table 1.

Table 1: Description of variables and functions used in Figure 2 (above)

#	Name of Variables/Functions	Description
1	Initiate_CollabSession ()	The Server-end that starts the collaboration. To be started or instantiated by the moderator
2	Int RoleID, CurPar, SessionID, Upar, ConfID, AS Integers. ParName as Char	Declare the following variables as Integer RoleID -1:Moderator, 2:Presenter, 3:Supervisor, 4:Co-Supervisor, 5: Lecturers, 6:Others CurPar = RoleID, SessionID = #Presenter, ConfID = #Serial, ParName = funcName(PresenterID)
3	Int funcName (Int PresID)	funcName (int) – Returns name of the presenter using the Presenter Id as parameter
4	Moderate_floor (Int RoleID, SessionID	Moderate_floor (int, int) – assign floor to the next participant validated by the moderator

6. RESULTS AND DISCUSSION

According to Dewan [20], the architecture of a collaborative application is characterized by the modules, layers, replicas, threads, and processes into which the application is decomposed; the awareness in these components of collaboration functions; and the interaction among these components [20].

The function, fairness, fault tolerance, ease of modification, and performance of the application, the amount of programming effort required to implement the application, and the reuse of existing single-user code are also influenced by the architecture of these systems [20].

In this proposed architectural framework of Collaborate^{IT}, the process of its development is broken down into three main steps:

i) Designing the functionality,
ii) Creation of sub-components of the application, and
iii) Use of programming tools for its implementation.

The scope of Collaborate^{IT} targets an academic environment and therefore the choice of X system being run by Unix operating system.

Unlike the floor control systems in other developed collaborative systems; Passenger – Stephan and Pascal [2]; XGSP-FLOOR Kangseok *et al.* [1]; and BFCP Pat [21], the distribution mechanism of the proposed architecture is aimed at not overloading the X-clients processes and reducing the load on the X-servers while maintaining the collaborative system integrity.

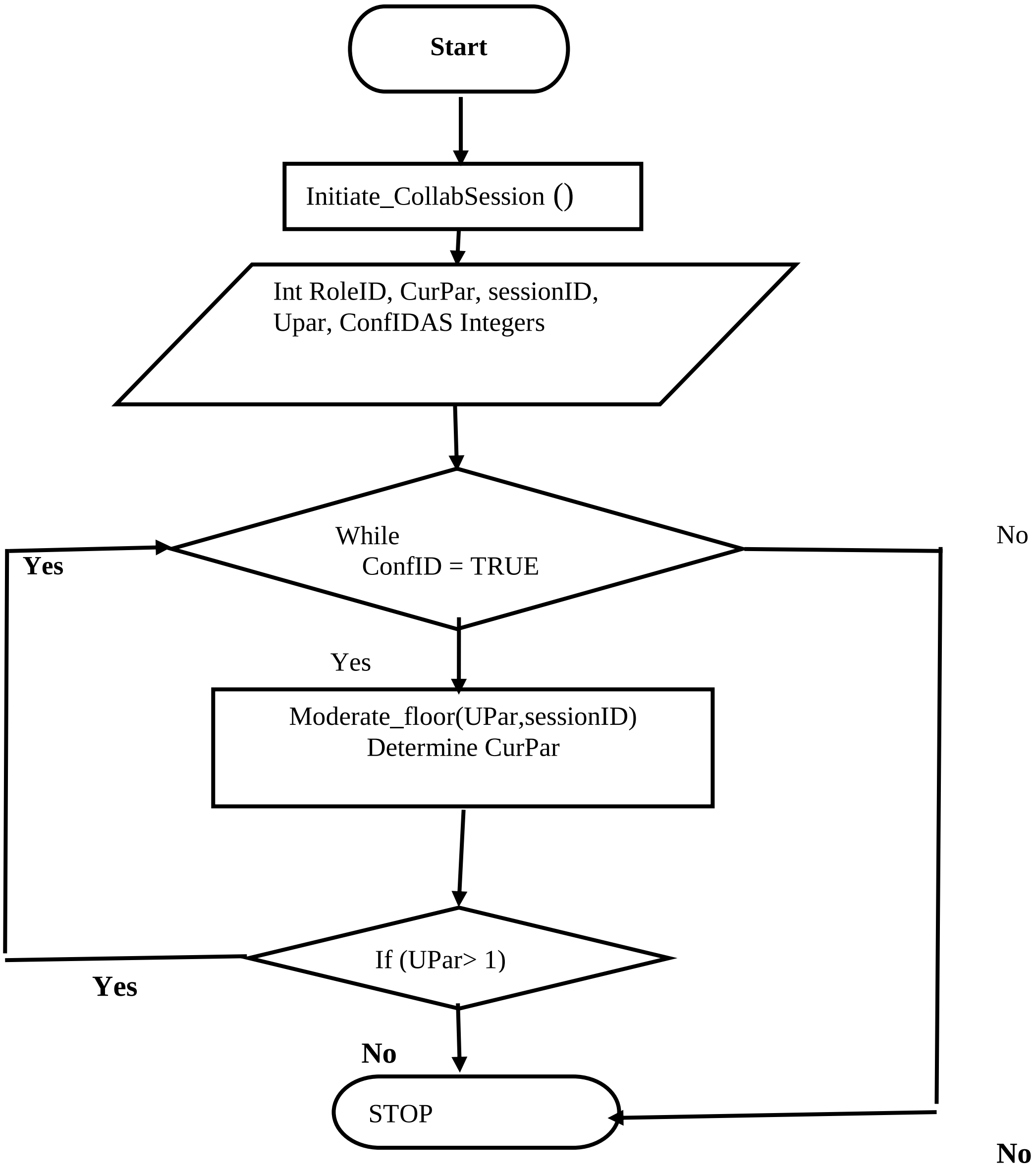

Figure 3: Flowchart illustrating the Start Collaboration functions

7. CONCLUSION AND FUTURE WORK

The architectural framework of Collaborate^{IT} has been presented with emphasis on its moderator-based floor control mechanism suitable for geographically dispersed participants of a seminar presentation in an academic environment. The proposed platform is the UNIX system for the X-Server and the windows platform for the X-Client embedded in a client-server architecture.

The Collaborate^{IT} is a tightly coupled system with each of the participant assigned a pre-defined role as a pre-requirement for the implementation of its floor control mechanism. The user and session management is proposed as a record like structure that could be implemented as a C-type ***'Struct'*** data type and stored on the server. Proposed future work includes the inclusion of multimedia tools; such as sound, images, PowerPoint slides and video streaming during session.

References

[1] Kangseok, K., Wenju, W. and Geoffery C. K. (2012): XGSP-FLOOR: Floor Control for Synchronous and Ubiquitous Collaboration. Grids Laboratory, Indiana State University

http://citeseerx.ist.psu.edu/viewdoc /summary?doi=10.1.1.157.9698/ Accessed, 15/08/2020

[2] Stefan Werner and Pascal A. Klein (2016): Desktop Conferences, LAB EXPERIMENT 4, CSCW Laboratory, University Duisburg-Essen Faculty of Engineering - Computer Engineering,

[3] Michel J., Guttormsen S. and Voor Horst, F. (1999): "Communicate-it- Developing the next generation teleconferencing systems". (Project presentation on the web) http://www.ihabepr.ethz.ch/research/ Accessed, 25/11/2020

[4] Ruibiao Q., Fred K. and Jerome R. (2010): A Conference Control Protocol for Highly Interactive Video- conferencing, Applied Research Laboratory, Department of Computer Science, Washington University, Saint Louis, MO63130, USA http://www.techrepublic.com/resourcelibrary/ whitepapers/ a-conference-controlprotocol-for- highly-interactive-video-conferencing /post/ Accessed, 10/03/2021

[5] Greenbag, S. (1991): "Personalize Groupware: Accommodating Individual roles and group differences". In the *Proceedings of the ECSW* 1991. Amsterdam pp. 17-32.

[6] Dommel, H. and Garcia, J. (1997a): Floor Control for multimedia Conferencing and Collaboration Multimedia Systems. (Springer-Verlag) 1997. pp. 23-38.

[7]Dommel, H. and Garcia, J. (1997b): Design Issues for Floor Control protocols. In the *Proceedings of Multimedia and Networking*, IS & T SPIE 2417: pp. 305 – 316.

[8] Myers, B., Chuang, Y., Tjamdra, M., Chen. M. and Lee, C. (2001): "Floor Control in a highly Collaborative Co- Located Task." (Technical Report). Human Computer Interaction Institute School of Computer Science, Carnegie Mellon University, Pittsburg. http:// www.cs.cmu.edu/~pebbles. Accessed, 10/01/2021

[9]Rainer, R.K., and Cegielski, C.G. (2010): Introduction to Information Systems: Enabling and Transforming Business. John Wiley & Sons, Apr 12, 2010 – pp. 172

[10] Lane, C. (1999): The distance learning Technology Resource guide 1st Edition, UCL Press, London pp. 42

[11] Galletta, D.F. and Zhang, P., (2006) Human-computer Interaction and Management Information Systems: Applications, M.E.Sharp Publisher, 2006

[12]Wakeman, I. (1999): Distributed Systems. Revised Edition, Prentice-Hall of India, pp. 236.

[13] Crowcroft, J. (1995): Open distributed systems. 1st Edition, UCL, Press, London. pp. 56

[14]Sauter, V. (2011): Decision Support Systems for Business Intelligence, John Wiley & Sons, Jul 20, 2011, pp. 254

[15]Hendrik, W. (2009): User Interfaces for Wearable Computers: Development and Evaluation Springer, Apr 20, 2009. pp. 289.

[16]Stair, R. (2012): Principles of Information Systems, 11th ed.Cengage Learning, Nov 5, 2012, pp. 441

[17 Johansen, R. (1984): Teleconferencing and beyond: communications in the office of the future, McGraw-Hill, Jan 1, 1984 pp. 123

[18] Mendez, A., S., Anta, F., A., Fernández. L., L. (2012): Quid Pro Quo: A Mechanism for Fair Collaboration in Networked Systems, http://arxiv.org/abs/1207.6045 Accessed, 20/07/2020

[19] Kausar, N. and Crowcroft, J., (2005): An architecture of Conference Control Functions, Term Paper, Department of Computer Science, UCL, Gower Street, London WC1E 6BT.

[20]Pra sun Dewan (1999). Architectures for Collaborative Applications *University of North Carolina Computer Supported Cooperative Work*, Edited by Beaudouin-Lafon. JohnWiley & Sons. Ltd

[21]Pat, J. (2012). Binary Floor Control Protocol – The Internal Protocol Journal, Volume 15, 0. 3, CISCO.https ://www cisco.com/c/en /us/about /press /internet-protocol/journal/ac-iisues/ table-cont57/153-binary.html Accessed, system is illustrated with the flowchart as presented in figure 3. The corresponding functions are also presented in Table 1.20/07/2021