Introducing Secured Node Registration and Supervised P2P File Sharing in Advanced Collaborating Environment

Mohammad Rezwanul Huq, Md. Ali Ashik Khan, Md. Rezwanur Rahman, Tamkin Khan Avi

Department of Computer Science and Information Technology (CIT),
Islamic University of Technology (IUT),
Gazipur, Dhaka, Bangladesh.

Abstract – A major challenge in a collaborative environment is to device an efficient file sharing and adaptation mechanism. Through this, the users at different nodes can achieve the highest degree of collaboration. Some authors have already proposed file sharing and adaptation framework. As per their proposed framework, users are allowed to share adapted files among them, invoking their file sharing and adaptation service built on the top of advanced collaborating environment (ACE). For file adaptation, a hybrid approach has been mentioned for adapting files which considers user's preferences as well as user's device capabilities. The goal of this adaptation approach is to provide the best possible adaptation scheme and convert the file according to the user's preferences and device capabilities.

In this paper, we propose some new features for file sharing and adaptation framework to have faster and more efficient and meaningful collaboration among users in advanced collaborating environment. We are proposing a Key based registration system as well as an algorithm for Key generation to automate the registration process for slave ACE nodes. We also propose an automated mechanism to distinguish trusted (registered) Slave ACE nodes, which are privileged for p2p file sharing under certain conditions. This approach will completely eliminate the manual intervention in the purposive node selection process. Moreover there would be a network traffic monitoring process which will radically reduce the network overhead to the master ACE node and introduce faster network collaboration in a distributive manner. Consequently, the approach leads towards better collaboration among the users of ACE.

Keywords: P2P File Sharing, Node Registration, Authentication, Collaboration.

1. Introduction

The notion of advanced collaborating environment is essential to provide interactive communication among a group of users. The advancement in the field of networking and multimedia technology has outmoded the traditional video conferencing concepts. The 3R factor that is- Right People, Right data and Right time, is the major concern of ACE, in order to perform a task, solve a problem, or simply discuss something of common interest [1].

Figure 1 depicts the concept of Advanced Collaborating Environment (ACE), where media, data, applications are shared among participants joining a collaboration session via multi-party networking [2]. Each ACE node has a couple of audio/video/interaction devices, LCD/projector/tiled display systems, and a number of support machines. Some early prototypes of ACE have been mainly applied to large-scale distributed meetings, seminar or lectures and collaborative work sessions, tutorials, training etc [3], [4].

Access Grid is a group-to-group collaboration environment with an ensemble of resources including multimedia, large-format displays, and interactive conferencing tools. It has very effectively envisioned the implementation of ACE in real life scenario. Venue server and venues concepts come from the Access Grid multi party collaboration system [4], [5]. Venue server is the server component for venues. Node represents the aggregation of hardware and software for participating in a meeting which is same as ACE nodes in ACE environment. Moreover, Venue Server has Data Store for each venue registered to it. Venue Data Store holds all the shared files from different users in the venue.

Services built for ACE environment also have been integrated to Access Grid. Here, it may be noted that, we use both the term 'node' and 'device' which are actually contain the same meaning in the later portion of the thesis.
We would like to define two new terms which will be used for the rest of our thesis.

- **Master ACE Node:** It is a kind of ACE node which has the capability to directly communicate to Venue through venue client as well as Venue Server. From the device configuration point of view, it has higher configuration as accessories like HD camera, microphone etc. are attached to this node.
- **Slave ACE Node:** Slave ACE node has less capabilities compared to Master ACE Node in terms of device configuration as well as it cannot communicate to Venue and Venue Server directly. Slave ACE node will be connected to a Master ACE node and all the functionalities needed for users in Slave ACE node are achieved through Master ACE node.

The rest of the paper is organized as follows. In Section 2, the Problem Statement is specified clearly. Section 3 begins

Manuscript received September 5, 2008.
Manuscript revised September 20, 2008.

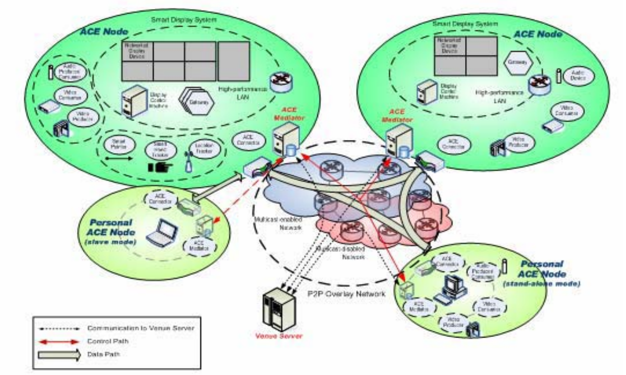


Figure 1 Advanced Collaborative Environment (ACE)

the discussion of related work. In section 4, our contribution through this paper is evidently described. In section 5, we describe our proposed mechanism followed by prototype implementation in section 6. Finally, we draw conclusion and talk about some of our future works in section 7.

2. Related Work

Much research has been initiated in the area of context-aware computing in the past few years. Many projects have been initiated for developing interactive collaboration. These projects enable users to collaborate with each other for sharing files and other media types.

The Gaia [8], [9], [10] is a distributed middleware infrastructure that manages seamless interaction and coordination among software entities and heterogeneous networked devices. A Gaia component is a software module that can be executed on any device within an Active Space. Gaia a number of services, including a context service, an event manager, a presence service, a repository and context file system. On top of these basic services, Gaia's application framework provides mobility, adaptation and dynamic binding of components.

Aura [11], [12] allows a user to migrate the application from one environment to another such that the execution of these tasks maximizes the use of available resources and minimizes user distraction. Two middleware building blocks of Aura are Coda and Odyssey. Coda is an experimental file system that offers seamless access to data [10] by relying heavily on caching. Odyssey includes application aware adaptations that offer energy-awareness and bandwidth-awareness to extend battery life and improve multimedia data access on mobile devices.

The work on user-centric content adaptation [13] proposed a decision engine that is user-centric with QoS awareness, which can automatically negotiate for the appropriate adaptation decision to use in the synthesis of an optimal adapted version. The decision engine will look for the best trade off among various parameters in order to reduce the loss of quality in various domains. The decision has been designed for the content adaptation in mobile computing environment.

3. Problem Statement

In this thesis, we have tried to enhance the efficiency of file sharing and adaptation framework described in [6], [7] by introducing Key based registration and automated trusted node identification to facilitate systematic file uploading and sharing**,** reducing network traffic through supervised p2p file sharing for better collaboration. These new features will establish the framework as an organized system and allow users at Slave ACE nodes to share files in an efficient and faster way. Thus our problem statement may be summarized as follows:

To provide secured registration as well as to set observation based privileges to nodes for efficient file uploading and sharing. Also to provide better and intelligent collaboration among the users by monitoring the network traffic around the Master ACE node and initiate supervised p2p file sharing as an effective alternative.

4. Our Contribution

To the best of our knowledge there is not much work in the issues like Data Adaptation and File Sharing in Advanced Collaborating Environment, though other related fields had been explored as we described in the previous section. File Sharing and Adaptation Framework illustrated in [6], [7] include file sharing and data adaptation service. File sharing service is demonstrated by the realization of data adaptation service. These two services are very much necessary to provide effective collaboration among users in advanced collaborating environment. But there are some problems associated with this approach. As for example, in the existing framework, the slave ACE nodes are registered by their MAC addresses which are not unique in every cases. Again another drawback is, the node registration for uploading files was a completely manual process executed by the network admin, which doesn't have strong ground on historical data analysis. Moreover, frequent requests from different nodes could potentially increase the network overhead towards Master ACE node and there is no systematic solution of this problem. Our target is to provide some extended features on the top of the existing framework so that the highest degree of collaboration among users can be realized.

In this paper, we tried to identify these problems and provide some effective solutions to address these issues. To ensure a secured registration we propose a **key based** mechanism for **user registration** to solve the problem with the MAC address. To increase the efficiency of the whole system we introduce **automated trusted node registration** which encompasses different observation phases. For solving the network issues there would be a monitoring process including different related parameters to reduce the Master-Centric network traffic and uploading the requested file in a distributed manner which opens avenues towards P2P file sharing system.

Thus, the comparison shows that our work will definitely encompass a meaningful advancement over the aforementioned work in the issues of secured user registration, node registration automation, reducing the network traffic at a central point of the system and increased collaboration among users through P2P file sharing to some extent.

5. Proposed Mechanism

Definitions

User Registration:
In Access Grid Collaboration Environment the User Registration is normally done by providing Email address along with other necessary information. As per our proposed system, each user will be given a Key for registration to avoid the complexity with the MAC address based registration.

Algorithm for Key Generation:
For generating a unique key we simply append the registration date and time with the given MAC address of the user. Thus the string becomes:
Unique Key = Registration date-time + MAC address.

Observation for trust-based nodes:
In our proposed mechanism the observation is done upon some specific criteria.
Pre-trusted Node: The system keeps track on the login period and downloaded file amount of a registered node. According to this data analysis some nodes get identified as Pre-trusted Nodes. Now they can upload files to the venue server up to a limited extent.
Trusted Node: For trusted node identification the system keeps track on the login period, downloaded file amount along with upload amount and frequency. Being a Trusted Node, a user can upload unlimited amount of files to the venue server also get privileged for p2p file sharing.
Requested node:
A Slave ACE node that requests for a file to Master ACE node is termed as Requested Node.

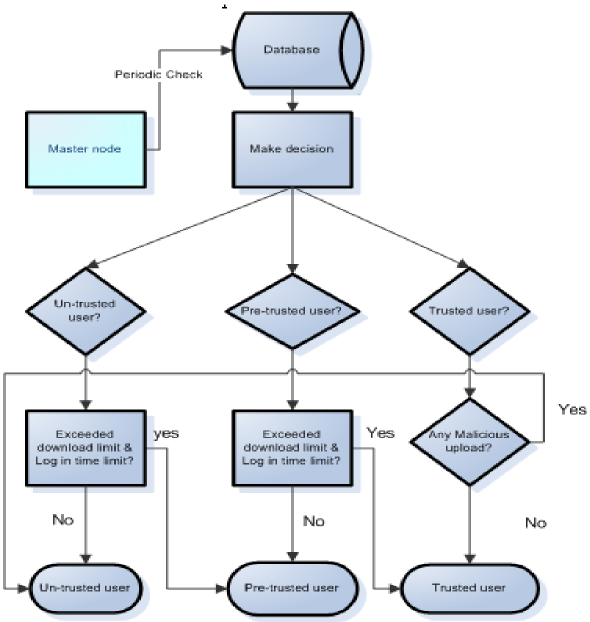


Figure 2 Action flow of Updating node privilege

Action Flow of Prototype Implementation

I. Action Flow for Updating Node Privilege:

In this process the master node performs a periodical check to the database to update the list of the trusted nodes. There are three different conditions. Firstly, for an un-trusted node, if the login time and download limit exceeds, it gets a promotion to be noted as pre-trusted node. Secondly, for a pre-trusted node, if the download limit and login time exceeds the given bound then it is promoted as trusted node. Finally, in case of a trusted node, if the user uploads any malicious file ever, it is demoted to the un-trusted state. Otherwise the state remains same. Figure 2 depicts the mechanism for updating the node privilege.

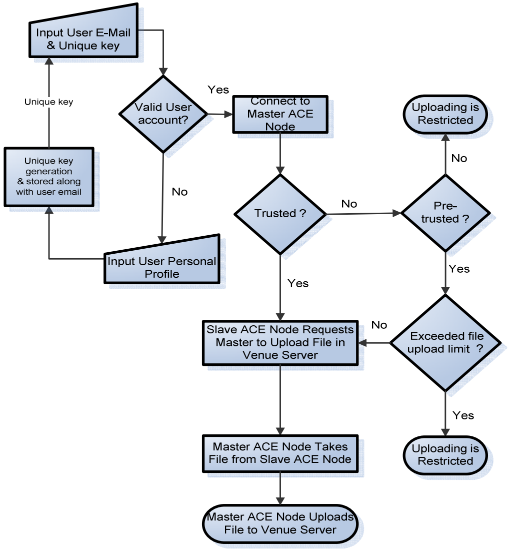

Input User E-Mail & Unique key
Unique key
Valid User account?
Yes
Connect to Master ACE Node
Uploading is Restricted
No
Unique key generation & stored along with user email
No
Trusted ?
No
Pre-trusted ?
Input User Personal Profile
Yes
Yes
Slave ACE Node Requests Master to Upload File in Venue Server
No
Exceeded file upload limit ?
Yes
Master ACE Node Takes File from Slave ACE Node
Uploading is Restricted
Master ACE Node Uploads File to Venue Server

Figure 3 Action flow of File Uploading mechanism

II. Action Flow for File Uploading Mechanism:

At first, user connects from Slave ACE node to Master ACE node. S/he needs to provide e-mail address, the assigned unique key and Master ACE node address. Then, the system checks whether the user already has a valid account with the system or not. If not then a **unique key**, generated for that particular node and user profile form are provided to that user for getting registered.

If the node has already been registered then the system checks whether the node is trusted or not. If trusted then the user can upload any amount of files to the venue server. If not trusted then the system checks whether it is pre-trusted or not. If not then, the user is identified as a normal node and cannot upload any file.

Now if the user is a pre-trusted node, then the system checks if it is below the assigned limit for uploading files then allowed uploading new files. Otherwise upload is restricted. Figure 3 depicts the File Uploading mechanism of the system.

III. Action Flow for File Downloading Mechanism:

Here again, firstly, user connects from Slave ACE node to Master ACE node. After being connected, the Slave ACE node requests for a file. The Master ACE node increases the Hit counter for that particular file. After this, two types of checking are executed.

The system checks whether the Hit counter exceeds the threshold value. If yes, then an advertise is multicasted to all the slave nodes of compatible device capabilities. The other checking is for whether the network traffic limit is exceeded or not. If yes, it checks for an online trusted node with the requested file. If such a trusted node is found, the request is redirected to that trusted node for downloading the file. And if such trusted node is not found then the request is entered in a queue till the network traffic gets low.

When the network traffic limit is not exceeded then any adapted version of the requested file of same preference already exists in the cache. If it is available then the file is simply sent from the cache. If not then the normal file adaptation approach is followed. Figure 4 depicts the File Sharing mechanism of the system.

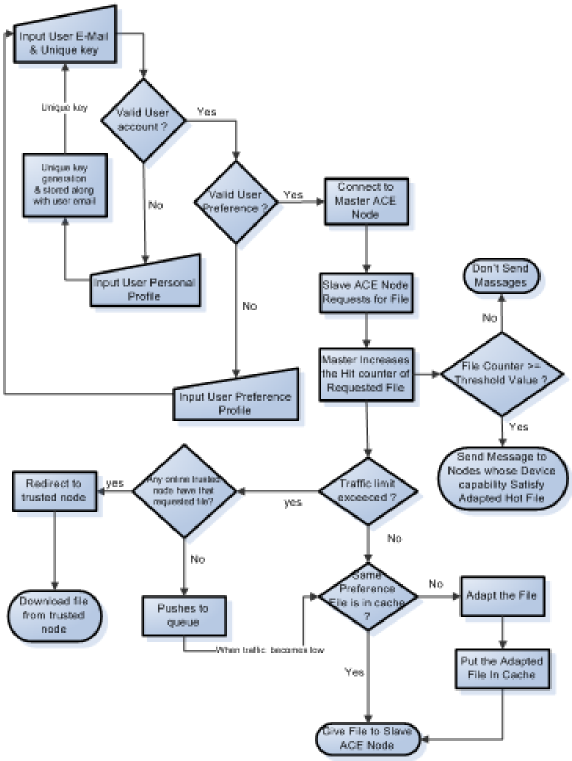
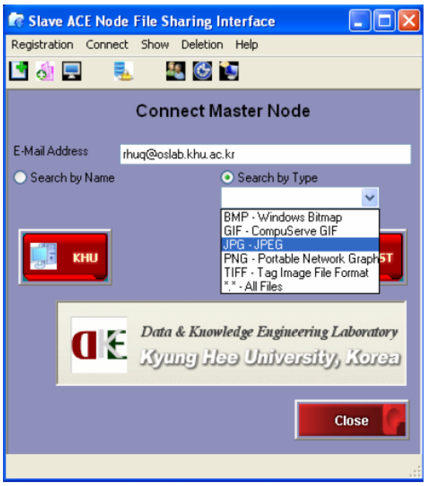

Input User E-Mail & Unique key
Unique key
Valid User account ?
Yes
Unique key generation & stored along with user email
No
Valid User Preference ?
Yes
Connect to Master ACE Node
Input User Personal Profile
No
Slave ACE Node Requests for File
Don't Send Massages
No
Master Increases the Hit counter of Requested File
File Counter >= Threshold Value ?
Input User Preference Profile
Yes
Send Message to Nodes whose Device capability Satisfy Adapted Hot File
Redirect to trusted node
yes
Any online trusted node have that requested file?
yes
Traffic limit exceeced ?
No
No
Download file from trusted node
Pushes to queue
Same Preference File is in cache ?
No
Adapt the File
When traffic becomes low
Yes
Put the Adapted File In Cache
Give File to Slave ACE Node

Figure 4 Action flow of File Downloading Mechanism.

6. Implementation Progress & Issues

Currently Implemented Modules

Master File Sharing Service: It had been implemented as an AG shared application. That is why, it implemented in Python. This module retrieves file from Data Store at venue server and sends this file for appropriate adaptation.


Slave ACE Node File Sharing Interface
Registration
Connect
Show
Deletion
Help
KHU
ST
Data & Knowledge Engineering Laboratory
Kyung Hee University, Korea
Close

Connect Master Node

E-Mail Address rhuq@oslab.khu.ac.kr

○ Search by Name ● Search by Type

BMP - Windows Bitmap
GIF - CompuServe GIF
JPG - JPEG
PNG - Portable Network Graph
TIFF - Tag Image File Format
. - All Files

Figure 5 Screen Shot of Previous work in [6]

Slave File Sharing Service: It had been implemented as stand-alone application. It was implemented in Python. It provides the interface to the user for entering venue URL and selecting the desired file.

Data Adaptation Service: It is a stand-alone application. It had been also implemented in python. That module has a decision engine which provides the appropriate adaptation scheme for converting the original file. At the beginning, the user enters his/her e-mail address and preferred file type for sharing. Then, he/she presses the appropriate button for connecting Master ACE node (see figure 5). After pressing the connect button the master file sharing service will connect to the venue data store and retrieves file of user mentioned type. Then, the user will select one of the files. The specified file will be downloaded through the master file sharing service, the decision engine takes the decision of selecting appropriate adaptation method and then it will be passed to the data adaptation service. The function DataAdapter() takes the file and convert it based on the decision provided by the decision engine. Then adapted file will be sent to Slave

Methods to Follow for Implementing New Features

Figure 6 shows the overall design of our proposed framework to enhance the already developed framework described earlier. The basic requirement of the file upload feature is the trusted node authentication. As per our proposed mechanism the Pre-trusted and trusted nodes are automatically validated by the system based on some periodic data analysis. As depicted in the schema diagram (see figure 7), the unique_key is the main factor which is set by the key generation algorithm, used to identify each of the nodes separately. The Pre-trusted nodes can upload the files up to a limit. The limit can be set depending on the application environment. The trusted nodes are facilitated with an unlimited amount of file uploading capability.

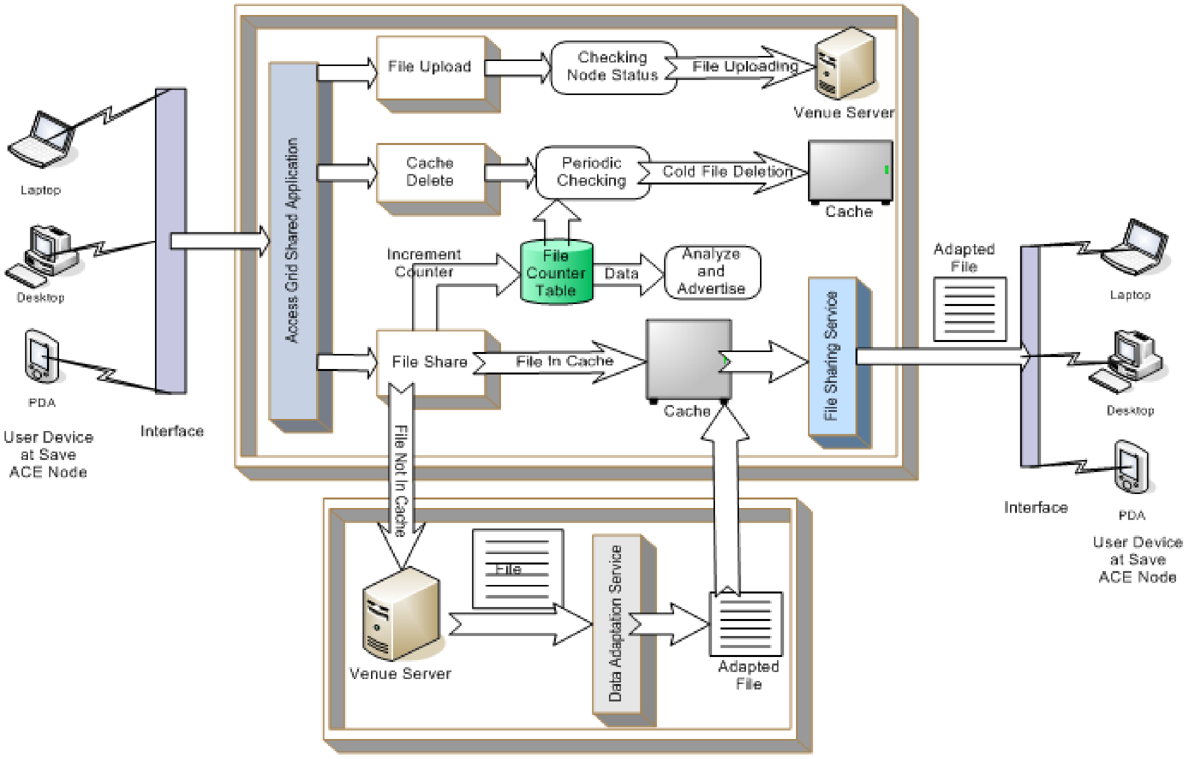


Figure 6 Block Diagram of the Proposed System

There are features of the frame work given in [7] like, hot file and cold file identification based cache updating, hit ratio based advertise multicasting etc are kept intact in our system. Additionally here the system keeps track of the downloaded file in the downloaded_file_tracking table.

Another feature of our proposed system is to monitor the network traffic towards the Master node. If the number of requests is more than a specified limit (in our system limiting value may be 10) then the system checks for the hit_file_id requested by a user in the downloaded_file_tracking table to find out a trusted node.

If found the trusted node can share the file with that requested user, which introduces p2p file sharing in the system.

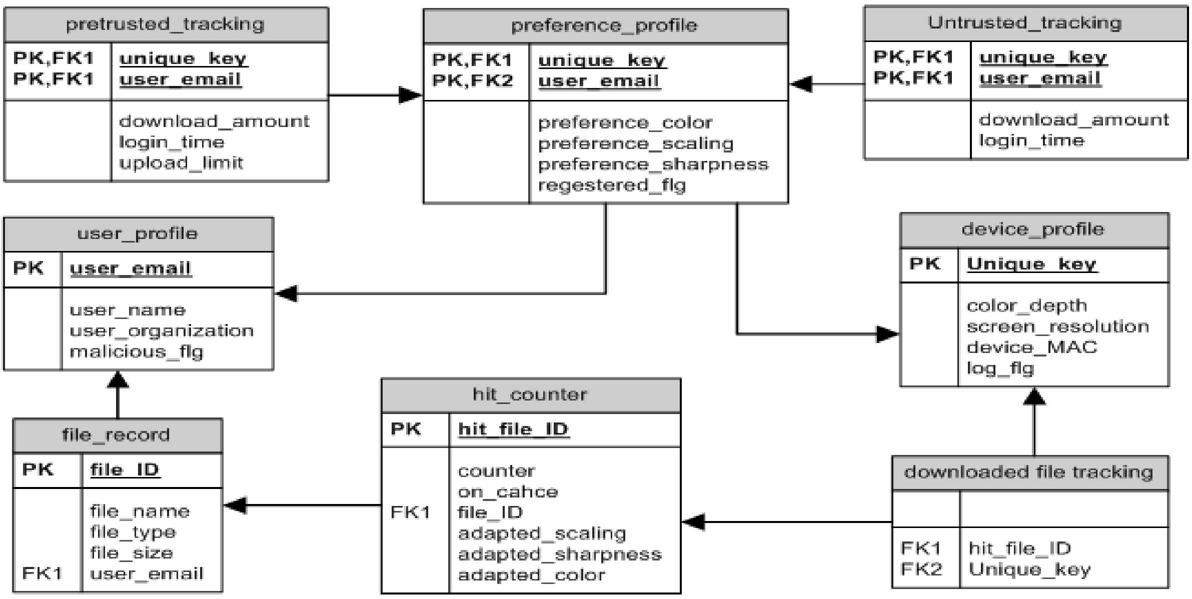


Figure 7 Schema Design for Backend Database

7. Conclusion & Future Work

In this paper, we have presented automated and network efficiency related features for file sharing and data adaptation framework in ACE. Our proposed features have made the framework more organized and enables the users to share adapted files faster and intelligently. Moreover, p2p sharing has added a new dimension to the interactivities of the users. These features together realize the improved file sharing service.

Lots of interesting works are to be done in the near future for efficient file sharing and adaptation service. An interesting filed can be Context Aware System that deals with issues like pattern matching and user requirement prediction, which will ensure higher degree of user collaboration.

We believe that our effort will certainly play a leading role for overcoming the deficiencies of this framework and also break new ground for more advancement in this field of research. Hopefully, our prototype implementation will be considered as a leading work in this domain in near future.

References

[1] B. Corri, S. Marsh and S. Noel, “Towards Quality of Experience in Advanced Collaborative Environments”, In Proc. of the 3rd Annual Workshop on Advanced Collaborative Environments, 2003.

[2] Sangwoo Han, Namgon Kim, and JongWon Kim, “Design of Smart Meeting Space based on AG Service Composition”, AG Retreat 2007, Chicago, USA, May 2007.

[3] R. Stevens, M. E. Papka and T. Disz, “Prototyping the Workspaces of the Future”, IEEE Internet Computing, pp. 51-58, 2003.

[4] L. Childers, T. Disz, R. Olson, M. E. Papka, R. Stevens and T. Udeshi, “Access Grid: Immersive Group-to-Group Collaborative Visualization”, In Proc. of Immersive Projection Technology Workshop, 2000.

[5] Access Grid, http://www.accessgrid.org/

[6] Mohammad Rezwanul Huq, Young-Koo Lee, Byeong-Soo Jeong, Sungyoung Lee, “Towards Building File Sharing and Adaptation Service for Advanced Collaborating Environment”, In the International Conference on Information Networking (ICOIN 2008), Busan, Korea, January 23-25, 2008.

[7] Mohammad Rezwanul Huq, Md. Abdul Mottalib, Md. Ali Ashik Khan, Md. Rezwanur Rahman, Tamkin Khan Avi, “Enhanced File Sharing Service among Registered Nodes in Advanced Collaborating Environment for Efficient User Collaboration”, In the International Conference on Grid Computing & Applications (GCA 2008), WORLDCOMP’08, Las Vegas, USA, July 14-17, 2008.

[8] Anand Ranganathan and Roy H. Campbell, “A Middleware for Context-Aware Agents in Ubiquitous Computing Environments”, In ACM/IFIP/USENIX International Middleware Conference, Brazil, June 2003.

[9] The Gaia Project, University of Illinois at Urbana-Champaign, http://choices.cs.uiuc.edu/gaia/, 2003.

[10] M. Roman, C. Hess, R. Cerqueira, A. Ranganathan, R. Campbell and KNahrstedt, "A Middleware Infrastructure for Active Spaces", IEEE Pervasive Computing, vol. 1, no. 4, 2002.

[11] M. Satyanarayanan, Project Aura, http://www-2.cs.cmu.edu/~aura/, 2000.

[12] M. Satyanarayanan, "Mobile Information Access", IEEE Personal Communications, http://www-2.cs.cmu.edu/~odyssey/docdir/ieeepcs95.pdf,Feb. 1996.

[13] W.Y. Lum and F.C.M. Lau, "User-Centric Content Negotiation for Effective Adaptation Service in Mobile Computing", IEEE Transactions on Software Engineering, Vol. 29, No. 12, Dec. 2003.

Author's Biography

Mohammad Rezwanul Huq received his Master of Computer Engineering degree from Kyung Hee University, Korea on February 2008. Currently, he is serving as Lecturer, CIT Dept. in Islamic University of Technology (IUT), Bangladesh.

Earlier he completed his B.Sc. in CIT from Islamic University of Technology (IUT), Bangladesh on September 2004. Later, he joined as Lecturer at the same university in Computer Science & IT department from December 2004 and still serves at the same university. His research interest includes Collaborative Environment, Ubiquitous Computing, Semantic Web and Data Mining. He is a member of the Institution of Engineers, Bangladesh (IEB). He also served as a program committee member for the 2008 Semantic Web and Web Services Conference held in Las Vegas, USA on July 2008.