JOURNAL OF ENGINEERING AND TECHNOLOGY, Vol. 8 No. 2 December 2010 ISSN 1684-4114

http://www.iutoic-dhaka.edu/jet

Implementation of Intelligent File Sharing in Advanced Collaborating Environment with Data Adaptation and Cache Optimization Service

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

Received 18 August 2010; Accepted 31 December 2010

ABSTRACT

In a collaborative environment devising an efficient file sharing and adaptation mechanism is a major challenge. This Paper shows the implementation of file sharing (upload and download) in advanced collaboration environment through a secured node from the unsecured ones. It also shows the data adaptation and cache optimization techniques to achieve the faster, more efficient and meaningful collaboration among users. It introduces a Key based registration system as well as an algorithm for Key generation to automate the registration process for slave ACE nodes. It also proposes 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.

Keywords: *File sharing, Adaptation framework, Advertise, Automatic Cache Update, P2P File Sharing, Node Registration, Authentication, Collaboration.*

* *Islamic University of Technology (IUT), Gazipur, Bangladesh.*
***KAZ Software, Dhaka, Bangladesh.* ****Sullivan University, Kentucky, USA.*
*****University of Twente, Enschede, Netherlands. E-Mail : citmrr@gmail.com*

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].

Fig.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.

We would like to define two new terms which will be used for the rest of our journal paper.

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.

Venue Server: Venue server holds the centralized data
(i.e. files, multimedia files, pictures etc). Master node directly connected to venue server, while Slave node connected Venue Server via Master node.

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.

The rest of the journal paper is organized as follows. Section 2 begins the discussion of related work. In Section 3, the Problem Statement is specified clearly. In section 4, our contribution through this journal paper is evidently described.

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 paper.

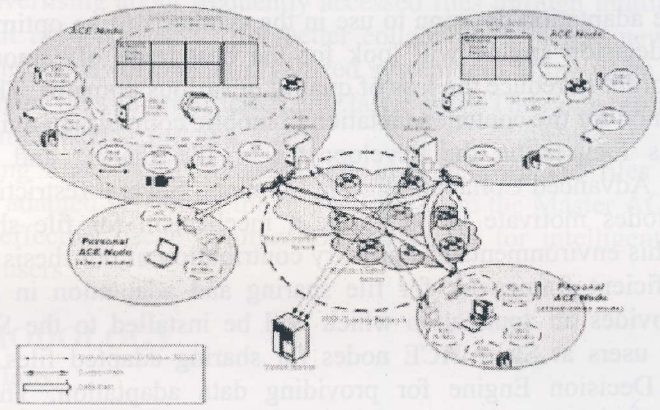

Figure 1: Advanced Collaborating Environment (ACE).

The rest of the journal paper is organized as follows. Section 2 begins the discussion of related work. In Section 3, the Problem Statement is specified clearly. In section 4, our contribution through this journal paper is evidently described. Section 5 describes our proposed mechanism followed by implementation and issues in section 6.

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.

This thesis focuses on the development of file sharing and adaptation framework for Advanced Collaborating Environment. Several restrictions on the Slave ACE Nodes motivate us to devise a mechanism for file sharing and adaptation in this environment. The primary contribution of our thesis is to build the overall efficient framework for file sharing and adaptation in ACE. Our framework provides an application which will be installed to the Slave ACE node to allow users at Slave ACE nodes for sharing adapted files. We have designed our Decision Engine for providing data adaptation. The primary concern of the decision engine is to find out the best match adaptation scheme depending on user preferences provide and Slave ACE node capabilities. During downloading request from Slave Node, master ACE nodes retrieve original files from Venue Data Store and then have the adapted version of the original file which is based on user preferences and device capabilities and also cached that adapted file to reduce redundancy of file adaptation requests. In uploading request Slave Node sends the file to Master node and Master Node finally upload the file to Venue Server. We devise a mechanism to facilitate the users with an intelligent and easy experience of collaboration based on hit ratio of files. Our framework includes Periodic Cache Update for better cache maintenances and advertizing files to Slave nodes.

To ensure a secured registration we propose a key based mechanism for user registration, which ensured a minimal privilege for the slave ACE nodes to upload files to the venue server via Master Nodes. 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.

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.

2 PROBLEM STATEMENT

In this paper, we have tried to develop a framework for file sharing and adaptation in order to provide file sharing and adaptation service to the users at Slave ACE nodes. We enhance the efficiency of file sharing and adaptation framework by introducing slave node registration to facilitate file uploading to the venue server as well as downloading from venue server via master, reducing redundancy of file adaptation requests using cache, Dynamic cache update, Key based registration and automated trusted node identification, supervised p2p file sharing, advertising about frequently accessed files through multicast messages, applying hit ratio analysis 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:

Allowing Slave ACE nodes to upload and download files as well as to provide an adapted version of the original file from the Master ACE nodes. Also providing effective cache optimization techniques for intelligent collaboration among the users.

3 METHODOLOGY

3.1 Definitions

Data Adaptation Service: The decision engine is the central part of data adaptation service. Data adaptation service will adapt data according to the decision provided by the decision engine. Decision engine will take decision based on user's preferences and device capabilities.

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.

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.

Requested node: A Slave ACE node that requests for a file to Master ACE node is termed as Requested Node.

Hit ratio: Hit ratio is the ratio between the number of times a file is requested and the total number of files requested. A mathematical representation of this term can be:

Hit ratio, H = Number of times f1 is requested / Number of total requests of files. Where, f1 = any particular file.

File counter: A process that counts the number of time a file is requested is termed as file counter

Threshold Value: It is a dynamic value that is used in our framework to determine the frequency of the request for a particular file and thus rank out the files for advertising.

Hot listed files: The files, the number of requests for which exceeds the Threshold Value are termed as Hot listed files.

Cold listed files: The files, for which the number of requests remains below the Threshold Value is termed as Cold listed files.

File Cache: It's a temporary storage where the files are stored for a limited period of time after successful adaptation

Periodic Cache Update: Cache will be update periodically, by deleting cold files from the cache.

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.

Three tier architecture: Our proposed Framework is based on three tier architecture; Server tier, Master tier, Slave tier. Server tier actually is Venue server, the centralized data storage. Master tier consists of Access Grid Shared Application and developed scripts. Finally, the Slave tier installed our developed shared application and interacts with venue server via master node **(Fig.2)**

3.2 Action flow of prototype implementation

3.2.1 Action flow for data adaptation service

The principle concern of data adaptation service is to adapt data according to the user's preferences and device capabilities. User's preferences and device capabilities will be stored into backend database. For adaptation decision, data adaptation service uses decision engine which will be described later. The main tasks performed by data adaptation service are as follows.

• Receives the original file sent by Master file sharing service.

• Executes the decision engine for choosing best match converter based on user preferences as well as device capabilities.

• Sends the original file to the best match converter to adapt the file which is the optimal adapted version taking under the consideration of user preferences and device capabilities.

• Sends the adapted file to the master file sharing service.

Upon receiving the request, Data adaptation service calls Decision engine for

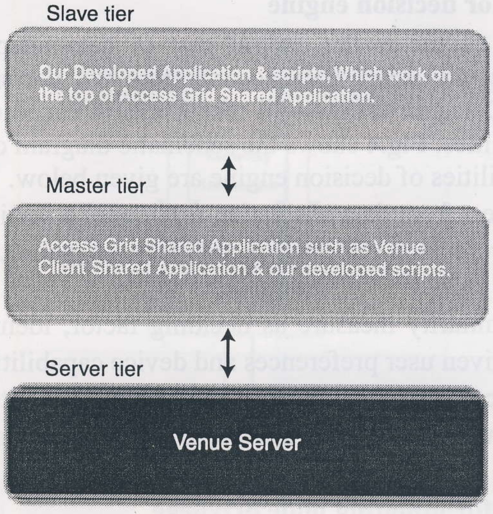

Figure 2: Three tier Architecture Framework.

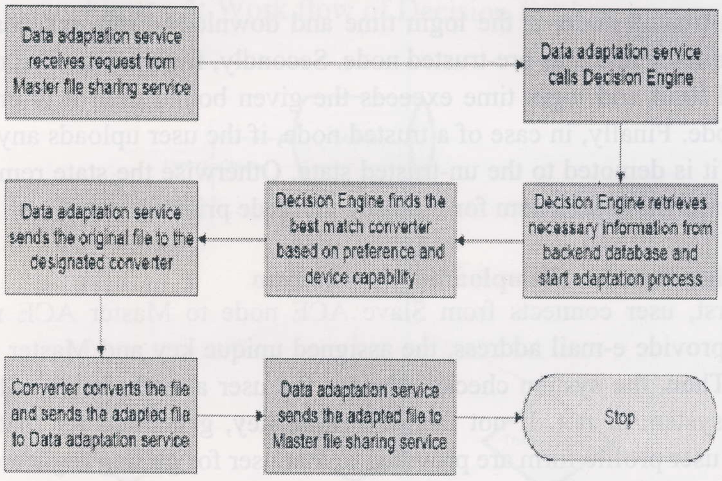

Figure 3: Control flow of Data Adaptation Service

choosing best match converter. Decision engine retrieves related information from backend database and based on user preferences and device capabilities finds out the best match converter. Data adaptation service sends the original file to the appropriate converter in order to get the adapted file. After receiving the adapted version of the original file, it sends the adapted file to Master file sharing service. The total control flow for Data adaptation service has been depicted in **Fig.3**.

3.2.2 Work flow for decision engine

The decision engine is the central part of data adaptation service. Data adaptation service will adapt data according to the decision provided by the decision engine. Decision engine will take decision based on user's preferences and device capabilities. **Fig.4** shows the schematic diagram of Decision Engine.

The responsibilities of decision engine are given below.

- Accepts the request from data adaptation service for a particular file.
- Retrieves user preference information and device capabilities from the backend database.
- Using cosine similarity measure as deciding factor, identifies the best match converter for the given user preferences and device capabilities.
- The best match converter will convert the original file into an adapted version which reflects the user preferences as well as render in the user device.

3.2.3 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. **Fig.5** depicts the mechanism for updating the node privilege.

3.2.4 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. **Fig.6** depicts the File Uploading mechanism of the system.

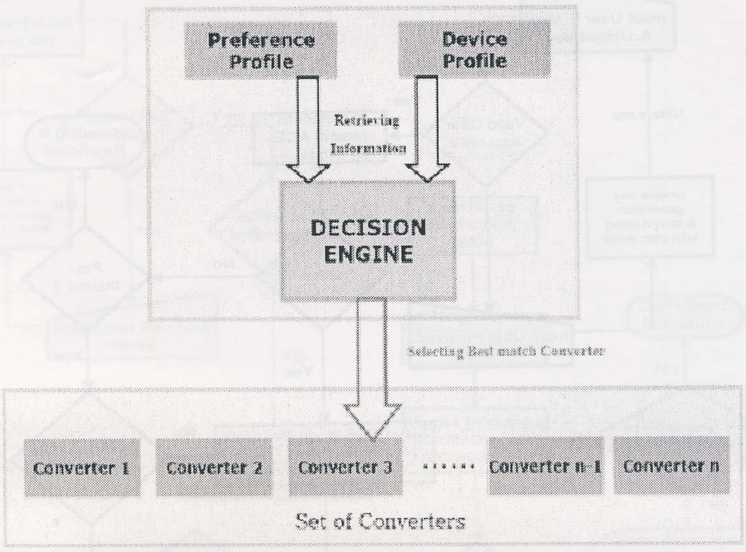

Figure 4: Work flow of Decision Engine.

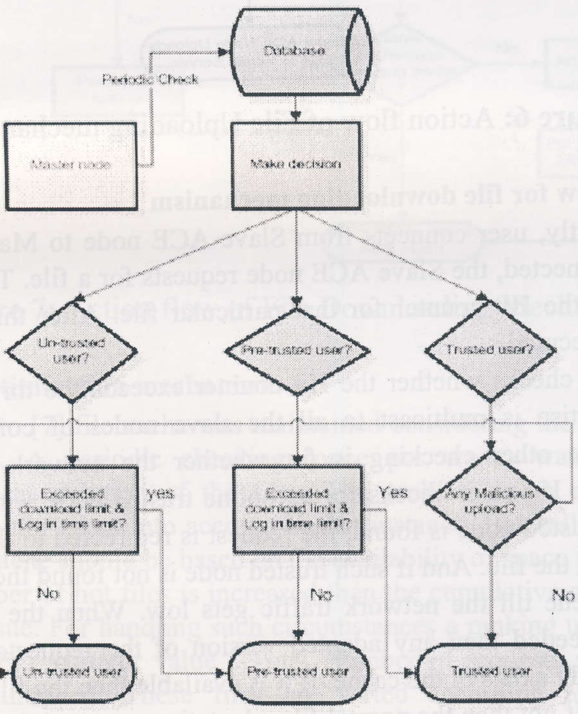

Figure 5: Action flow of Updating node privilege.

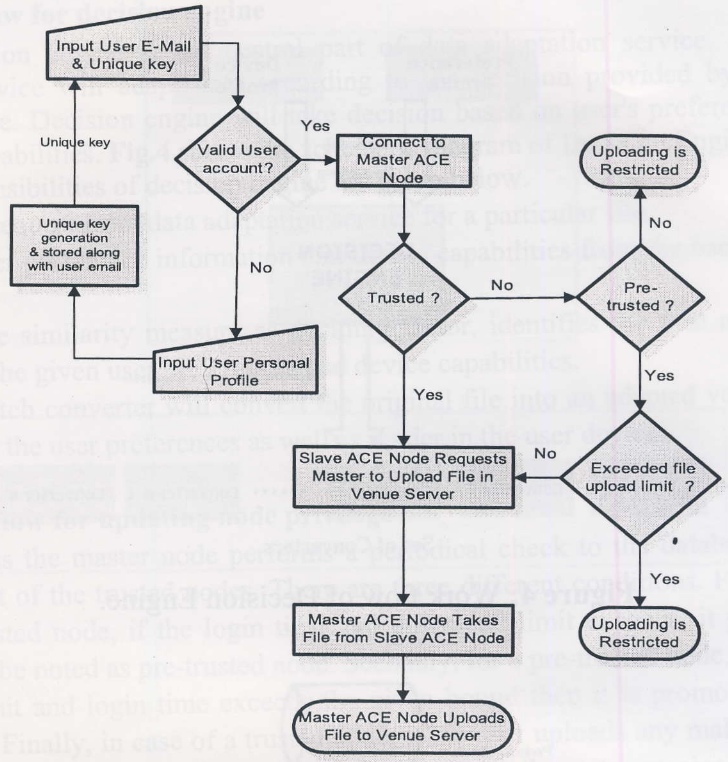

Figure 6: Action flow of File Uploading mechanism.

3.2.5 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 advertise is multicast 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. **Fig.7** depicts the File Sharing mechanism of the system.

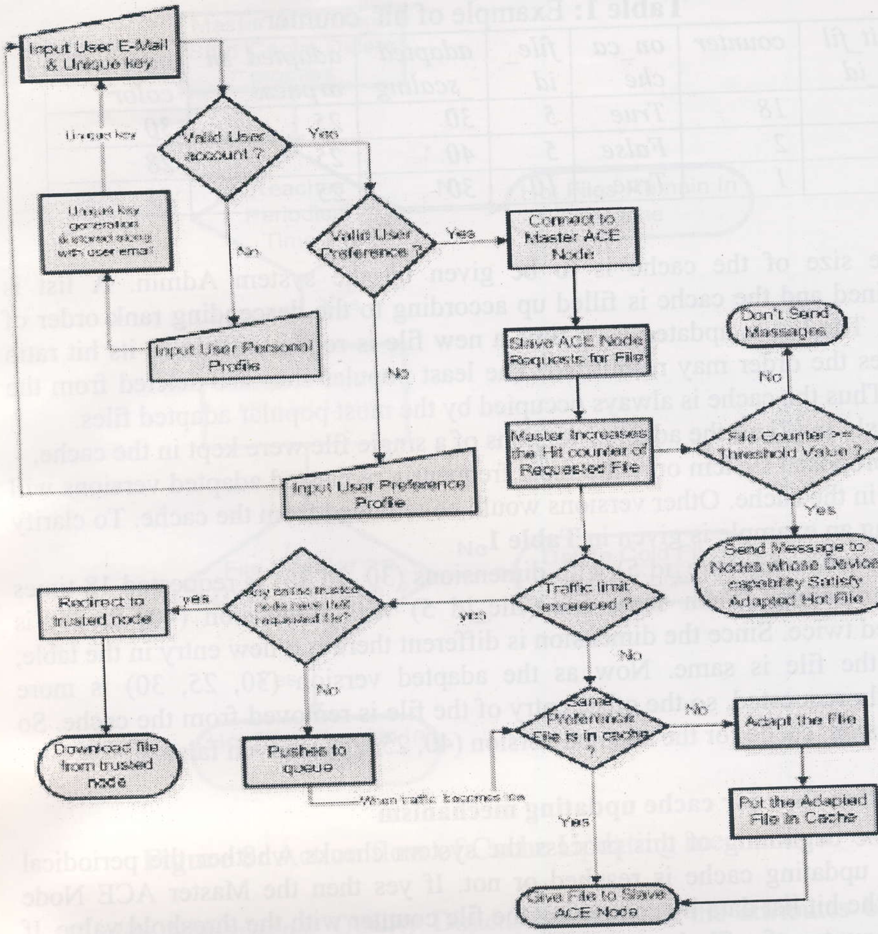

Figure 7: Action flow of File Downloading Mechanism.

3.2.6 Cache optimization mechanism

According to [7] file cache was maintained considering the hit ration of any requested files. A periodic checking (once per week) was mentioned for accomplishing the deletion of the files. The problem is an important factor, Cache size was not taken into account in that instance. Basically the deletion of the files from cache should be based on the availability of space in the cache. When the number of hot files is increased then the cumulative size of these files would be an issue. For handling such circumstances a ranking using the hit ratio formula H = (File counter value / Total Number of requests arrived) × 100% should be maintained. These files are sorted according to the rank and maintained in a queue.

Table 1: Example of hit_counter

hit_file_id	*counter*	*on_cache*	*file_id*	*adapted_scaling*	*adapted_sharpness*	*adapted_color*
1	*18*	*True*	*5*	*30*	*25*	*30*
2	*2*	*False*	*5*	*40*	*25*	*28*
3	*1*	*True*	*10*	*30*	*25*	*55*

The size of the cache is to be given by the system Admin. A list is maintained and the cache is filled up according to the descending rank order of the list. The list is updated each time a new file is requested and as its hit ratio increases the order may reshuffled. The least popular files are deleted from the cache. Thus the cache is always occupied by the most popular adapted files. Previously in [7] all the adapted versions of a single file were kept in the cache. In our proposed system only the most frequently requested adapted versions will be kept in the cache. Other versions would be removed from the cache. To clarify this thing an example is given in **Table 1**.

Suppose, a file (file_id 5) with dimensions (30, 25,30) is requested 18 times for downloading. Then same file (file_id 5) with dimension (40, 5, 28) is requested twice. Since the dimension is different there is a new entry in the table, though the file is same. Now as the adapted version (30, 25, 30) is more frequently requested, so the other entry of the file is removed from the cache. So the status on_cache for the adapted version (40, 25, 28) is given false.

3.2.7 Action flow for cache updating mechanism

At the beginning of this process the system checks whether the periodical time for updating cache is reached or not. If yes then the Master ACE Node analyze the hit list data by comparing the file counter with the threshold value. If the file counter of a file is more than the threshold value then that particular file would be treated as hot file and it would remain in the cache. If the file counter does not exceed threshold then it is treated as cold item and is deleted from the cache. **Fig.8** depicts the File Sharing mechanism of the system.

4 IMPLEMENTATION PROGRESS & ISSUES *(IMPLEMENTED MODULES)*

4.1 Decision engine

Decision Engine provides the best match converter for a given file considering the user preferences and device capabilities. We use some terminologies related to Decision Engine for decision generating mechanism. The definition of each of these terms has been given below.

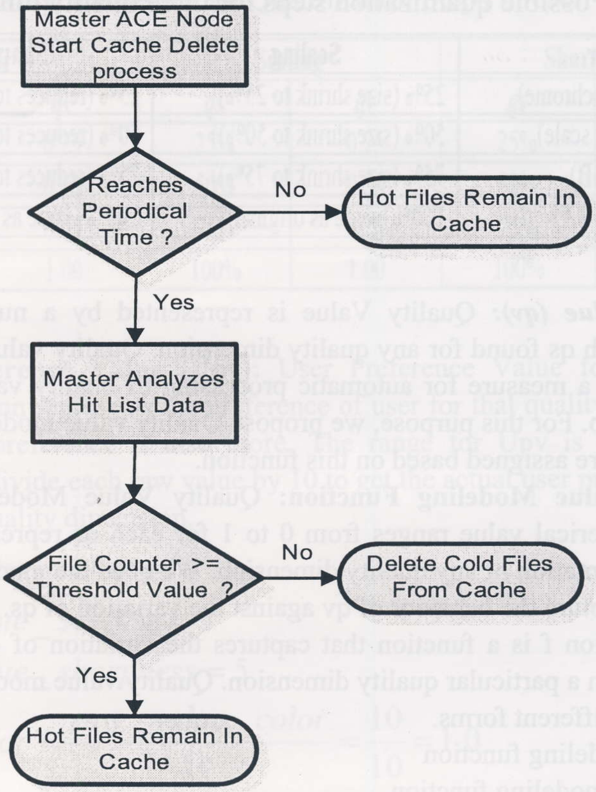

Figure 8: Action flow of Cache Updating mechanism.

Quality Dimensions: Quality Dimension refers to the attributes based on which the representation of the same file differs. A set of quality dimension can be identified for different types of multimedia content. These quality dimensions together forms the QoS for any content type. The quality of a certain version of an object can be seen as a point in an n-dimensional space, where n is the number of different qualities. Take the QoS of an image file an example:
QoS image = (color, scaling, sharpness)

Here, color denotes to the color depth of an image where as scaling refers to the size of the image and sharpness refers to the sharpness of an image.

Quantization Step (qs): Quantization Step refers to each possible representation state available for any quality dimension. **Table 2** shows the possible quantization steps for an image file on three quality dimensions such as color, scaling and sharpness.

Table 2: Possible quantization steps for three quality dimensions

Color	Scaling	Sharpness
1 bit image (monochrome)	25% (size shrink to 25%)	25% (reduces to 25%)
8 bits image (gray scale)	50% (size shrink to 50%))	50% (reduces to 50%)
24 bits image (RGB)	75% (size shrink to 75%))	75% (reduces to 75%)
32 bits image (RGBA)	100% (same as original)	100% (same as original)

Quality Value (qv): Quality Value is represented by a numerical value assigned for each qs found for any quality dimension. Quality value ranges from 0 to 1.We need a measure for automatic processing of quality values based on quantization step. For this purpose, we propose Quality value modeling function. Quality values are assigned based on this function.

Quality Value Modeling Function: Quality Value Modeling Function assigned a numerical value ranges from 0 to 1 for each qs represents different states of representation of any quality dimension. We need the modeling function so that it can capture the behavior of qv against the variation of qs. Quality Value Modeling function f is a function that captures the variation of qv against the variation of qs in a particular quality dimension. Quality value modeling function may have two different forms.

- First order modeling function
- Second order modeling function

For the first order modeling function, qv increases (or decreases) linearly with increase (or decrease) of qs and can be described as follows:

$$qv = f(qs)$$

$$f(qs) = \frac{qs - qs_{\min}}{qs_{\max} - qs_{\min}} \qquad where, \quad qs_{\min} = 0$$

For the second order modeling, the modeling curve is characterized by the second order equation. The second order modeling function can be represented as follows:

$$qv = f(qs)$$

$$f(qs) = a(qs)^2 + b(qs) + c \quad where, \quad a, b, c \quad are \text{ constant}$$

For our prototype implementation we use first order quality value modeling function. The quality values using this modeling function for different quantization step in each dimension has been given in **Table 3**.

Table 3: Quality Values for all Quantization Steps in each Dimension

Color		Scaling		Sharpness	
qs	qv	qs	qv	qs	qv
1 bit image	0.25	25%	0.25	25%	0.25
8 bits image	0.50	50%	0.50	50%	0.50
24 bits image	0.75	75%	0.75	75%	0.75
32 bits image	1.00	100%	1.00	100%	1.00

User Preference Value (Upv): User Preference Value for a particular quality dimension represents the preference of user for that quality dimension. If value is then preference is also more. The range for Upv is from 1 to 10. Therefore, we divide each raw value by 10 to get the actual user preference value Upv for each quality dimension.

$$raw\_value\_color = 10$$

$$raw\_value\_scaling = 5$$

$$raw\_value\_sharpness = 5$$

$$Upv\_color = \frac{raw\_value\_color}{10} = \frac{10}{10} = 1.0$$

$$Upv\_scaling = \frac{raw\_value\_scaling}{10} = \frac{5}{10} = 0.5$$

$$Upv\_sharpness = \frac{raw\_value\_sharpness}{10} = \frac{5}{10} = 0.5$$

As for example, Here Upv for any quality dimension cannot be less than 0.0 and greter than 1.0.

Similarity (SIM): Similarity defines the similarity measure used to identify the similarity between User Preference Vector and Converter Preference Vector. We use cosine similarity (**Fig.9**) measure for this purpose. Cosine similarity measure finds the angle between two vectors. If the value becomes 1, it means that two vector coincides with each other. If the value is 0, it means that two vectors are perpendicular to each other. The mathematical formula has been given below.

$$SIM = \frac{Vup * Vcp}{|Vup| * |Vcp|}$$

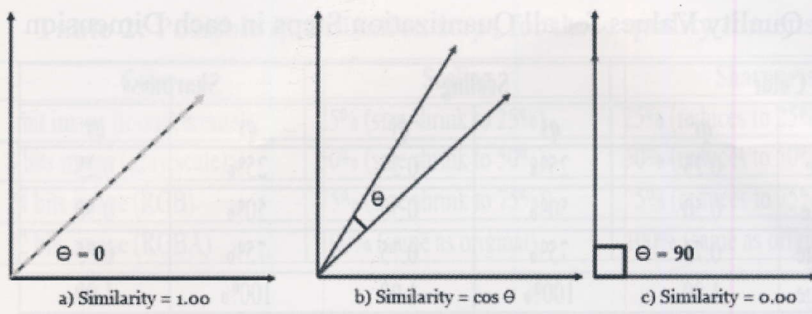

Figure 9: Cosine Similarities in 2-D Space.

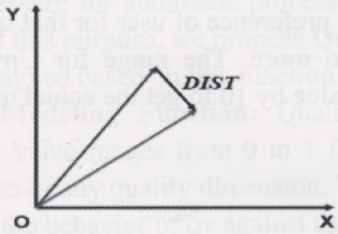

Figure 10: Distances between Two Vectors in 2-D Space.

```
Input: V_up, {V_cp}, Device_color_depth, Device_screen_resolution, Original_Image_Resolution

Output: Best match V_cp

FOR each V_cp in {V_cp}
    COMPUTE SIM
    COMPARE SIM >= MAX_SIM
        IF TRUE, COMPARE SIM = MAX_SIM
            IF TRUE, COMPARE DIST < MIN_DIST
                IF TRUE, COMPARE Device_color_depth supports N_cp-color
                    IF TRUE, MIN_DIST = DIST and Best match V_cp = V_cp
                        BREAK
                    ELSE, CONTINUE
                ELSE, CONTINUE
            ELSE, COMPARE Device_color_depth supports N_cp-color
                IF TRUE, MAX_SIM = SIM and MIN_DIST = DIST and Best match V_cp = V_cp
                    BREAK
                ELSE, CONTINUE
        ELSE, CONTINUE

COMPARE Device_screen_resolution >= Best match N_cp-scaling * Original_Image_Resolution
    IF TRUE, RETURN Best match V_cp
    ELSE, SET Best match U_pv-scaling_x = Device_screen_resolution_x / Original_Image_Resolution_x
        SET Best match U_pv-scaling_y = Device_screen_resolution_y / Original_Image_Resolution_y
        COMPARE Best match U_pv-scaling_x < Best match U_pv-scaling_y
            IF TRUE, UPDATE Best match U_pv-scaling = Best match U_pv-scaling_x
            ELSE, UPDATE Best match U_pv-scaling = Best match U_pv-scaling_y
        RETURN Best match V_cp
```

Figure 11: Formal Algorithms for Decision Engine

Distance (DIST): Distance **(Fig.10)** is a measure used to calculate the distance between converter preference vector Vcp and user preference vector Vup in n-dimensional space where 'n' is the number of quality dimensions. We use DIST measure [25] when multiple number of converters having same similarity to user preference. The converter preference vector having smallest distance from user preference vector will be chosen as the best match converter. In a word, the DIST measure will be used as a tie-break option when multiple converters have same SIM value.

4.2 Data uploading module

There is a python script, named upload script with is responsible for uploading files to venue server from slave and resides in Master Node. Upload script listen on socket port 8001. When a slave user select a file for uploading then the selected file name send via socket to the upload script and at the same time file transfer to the File Zilia FTP server at Master node. Thereafter upload script takes the file from file zilla and finally upload the file to venue server.

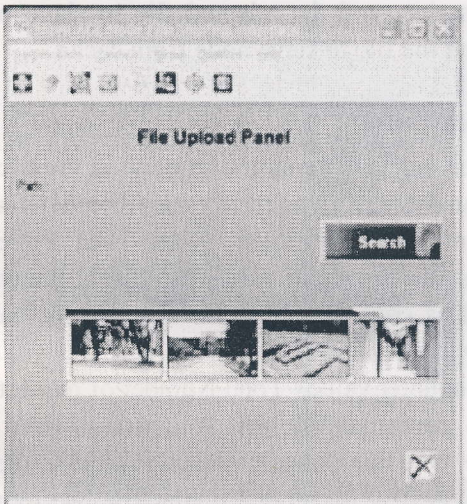

Figure 12: Screen Shot of File Uploading Interface

4.3 Data adaptation service

It is a stand-alone application. It had been also implemented in python. That module has a decision engine **(Fig.11)** 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 **(Fig.12)**. Then, he/she presses the appropriate button for connecting Master ACE node (see **Fig.13**). 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

4.4 Data downloading module

At first slave user must connect to the master node, then retrieve python script at master node communicate with chooser script at slave node thereafter the chooser **(Fig.14)** script poop up the file chooser interface. When the user select the desire file then the selected file name is sent via socket port 4001 to download script. Download script check whether the requested file in cache. If not in cache then retrieve the file from the venue server and cached it. Now decision engine adapt the file based on user preferences and device capabilities. After that download script send the adapted file to slave node. Otherwise, if cached then simply send the file to slave node from cache.

4.5 Cache maintenance

The existence of any adapted version of the file in the cache mainly depends on the hit ratio based analysis. A dynamic threshold value is used for ranking the files according to the number of time the file has been requested. That is if the file counter is reaches the threshold value the file is known as Hot File and otherwise it is labeled as cold file. Now the formula for calculating threshold value devised as follows:

Hit ratio T = (File counter value / Total Number of requests arrived) × 100%

If $T < 5\%$ for any particular file, that is termed as Cold file and the file is deleted from the cache. One thing must be mentioned here, the deletion of files is done periodically. If $T > 30\%$ for any particular file, we termed that file as hot file. They are not deleted from the File Cache.

4.6 Periodic cache update

Cache update script is a Daemon (background thread) which becomes active periodically and deletes the cold listed files from cache. After the deletion it goes to sleep.

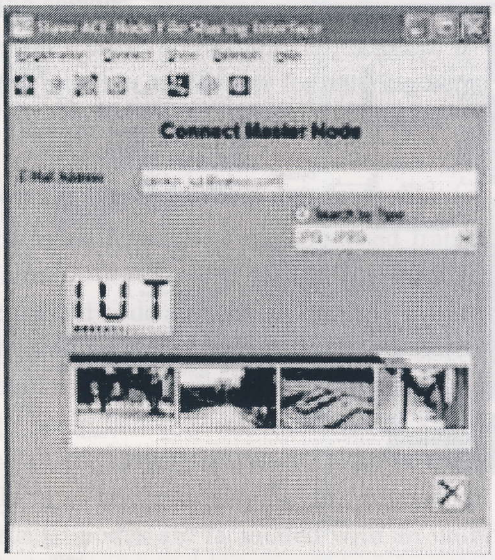

Figure 13 Screen Shot of File Downloading Interface.

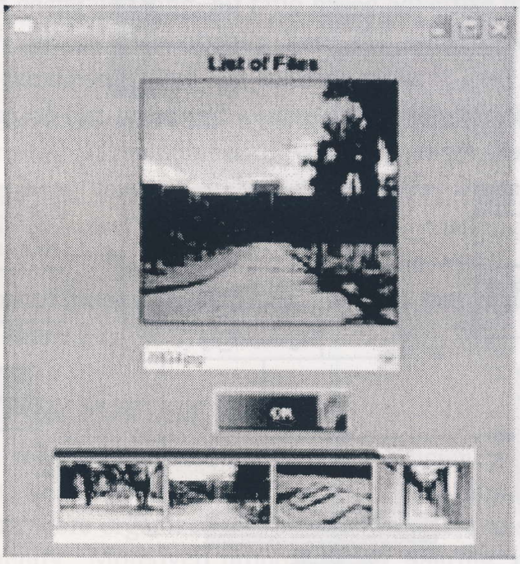

Figure 14: Screen Shot of File Chooser Interface.

4.7 Master files 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.

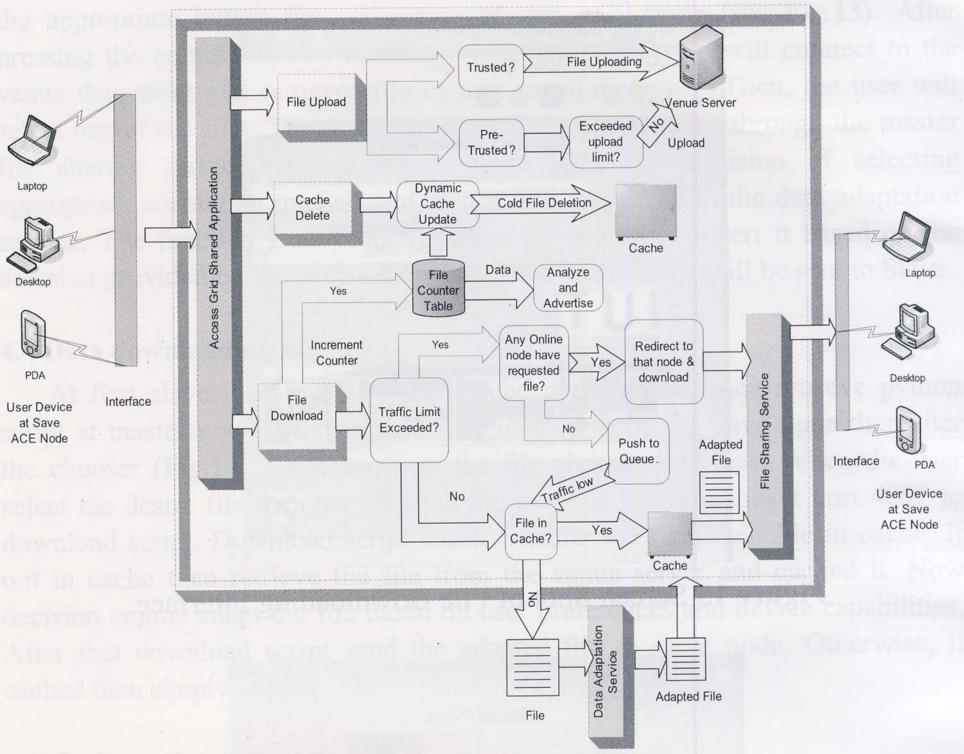

Figure 15: Block Diagram of the Proposed System.

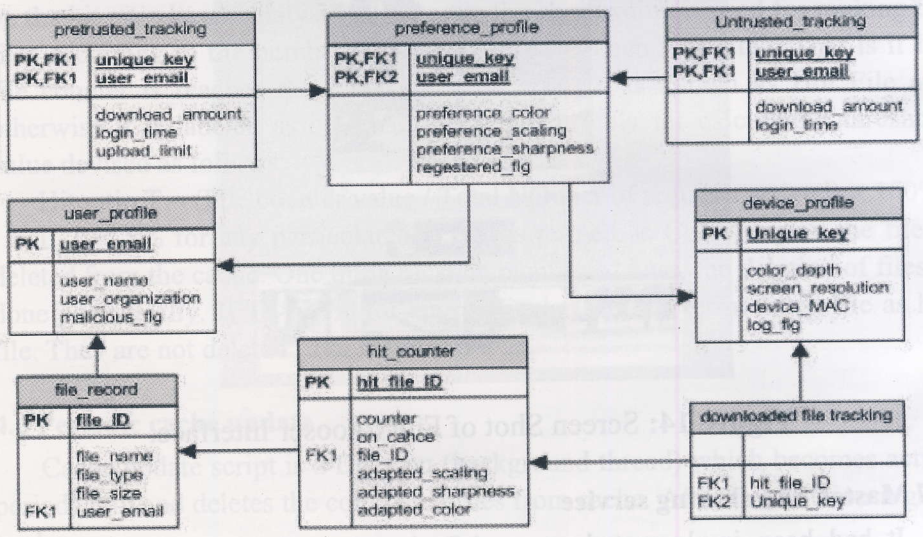

Figure 16: Schema Design for Backend Database.

4.8 Slave files 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.

4.9 Methods to follow for implementing new features

Fig.15 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 **Fig.16**), 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.

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.

5 CONCLUSION & FUTURE WORKS

This paper has shown automated and network efficiency related features for file sharing and data adaptation framework in ACE. The implanted features have made the framework more organized and enables the users to share adapted files faster and intelligently. Moreover proposed 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.

It is believed that the 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, the prototype implementation will be considered as a leading work in this domain in near future.

REFERENCES

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

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

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

[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.

(Contd. from 2nd cover page)

8. SI units must be used in the manuscript. However, other units may be used in parenthesis.

9. Tables should be referred to in consecutive Arabic numerical. Each table must have a table caption.

10. Line drawings must be in a form suitable for reproduction e.g., laser printout, drawn in Indian ink on white paper or on tracing paper. Photographs should have a glossy finish. Each figure must have a number and a figure caption. Elecrtonic mode is preferred.

11. References should be set out in alphabetical order of the author's last name in a list at the end of the article. They should be given in standard form as in the following examples :

(a) Journal
Bloomer G and Wright A. (1984) Scheduling of Vehicles from Factory to Depot. *Journal of Vehicles*, Vol. 12: pp. 590-598.

(b) Book
Best, John., and Kahn, James V., Research in Education, Prentice-Hall, New Jersey, 1986.

(c) Monograph
Syedali, M.M. "Computer Controlled Car", Thesis, M.Sc. Engineering, Department of Mechanical Engineering, BUET, 1990

(d) Conference Paper
Hasan M and Ullah MS. "Tourism development in the Chittagong Hill Tracts of Bangladesh after the peace agreement of 1997", a paper submitted to the Regional Conference on physical mobility and development in the mountains", Tribhuvan University, 15-17, March, 2000 Kathmandu, Nepal, pp.12

(e) Unpublished paper
Ahmadi, R and Tangs : Production Allocation with Dual Provisioning, Working Paper, Anderson Graduate School of Management, UCLA (1991)

12. The University does not accept responsibility for loss or damage of manuscript while in mail.

13. The responsibility for opinions in the contributions rests entirely on their authors.

14. The author (s) must submit declaration that the paper was not published elsewhere.

15. In case of joint papers, communication will be made with the first author.

16. The University will reserve the copyright of the paper once it is accepted for publication in the Journal. The authors must obtain written parmission from REASP, IUT for publication elsewhere.

Procedure for acceptance of papers and publications :

1. Papers submitted for publication will be referred to the listed reviewers for assessment. However, the editorial board will make initial screening of the papers.

2. After the assessment, the authors may be requested to modify/clarify certain points.

3. Accepted/modifed/corrected papers will be published in the next issue of the Journal.