4. WEBDEDIP ENVIRONMENT

4.1 INTRODUCTION

As in the parallel processing, processes are executed simultaneously on different computers, connected through LAN, frequent data transfer is necessary for effective performance. These data transfer takes place through messages. Also for load balancing, process starting, and process termination, an effective message passing system is necessary. Therefore, to achieve estimated performance, suitable message passing tool is required. PVM and MPI are two well known such tools. But use of these libraries requires some concepts of parallel programming like data communication, process forking, process joining, and data synchronization etc. Moreover, it requires special debugging technique for testing the parallel application. Generally, engineers are not expert in these concepts of parallel processing. In such a situation, implementation and performance measurement of parallel processing becomes difficult. So it is general expectation to have some application software, which can provide easy way of implementing parallel processing over network of computers. WebDedip is one of such user friendly distributed processing environment, which helps user to develop parallel application easily over network of heterogeneous system.

4.2 OVERVIEW OF WEBDEDIP

The WebDedip (Web Enabled Development Environment for Distributed Image Processing) [99] has three tier architecture; GUI, DedipServer and Agents, as shown in Fig 4.1. The GUI is the web enabled graphical user interface to make the entire user interaction truly system independent. It supports various Java Applets for application configuration, application building, application operation initiation, application progress monitoring, and session controlling. The user initiates the interaction by visiting a predefined site using a standard browser. The standard web server loads the required GUI on the web browser. It has a back-end DedipServer running on the web site. When the GUI submits the request to the DedipServer, it reads the application configuration information from the configuration file. The DedipServer initiates the execution of the first process in the interdependency chart. Normally, most of the applications have a single starting process. If any application has multiple starting processes, it

initiates execution of all such independent processes. It informs the agents on the target node to start the execution of the process. The agent sends the status information back to the DedipServer when the process is completed. The DedipServer finds out the dependent processes on the successful completion of a process and initiates the execution of each such process. The required files are transferred from one node to another. WebDedip has a callable library in Java to interface with the FTP server that helps in transferring files. The required process is automatically inserted in the configuration when IP designer inserts the IO dependency information between two processes. The DedipServer stores complete information about all the applications configured on the web site. The DedipServer exchanges information with the Dedip Backup Server making the model fault tolerant.

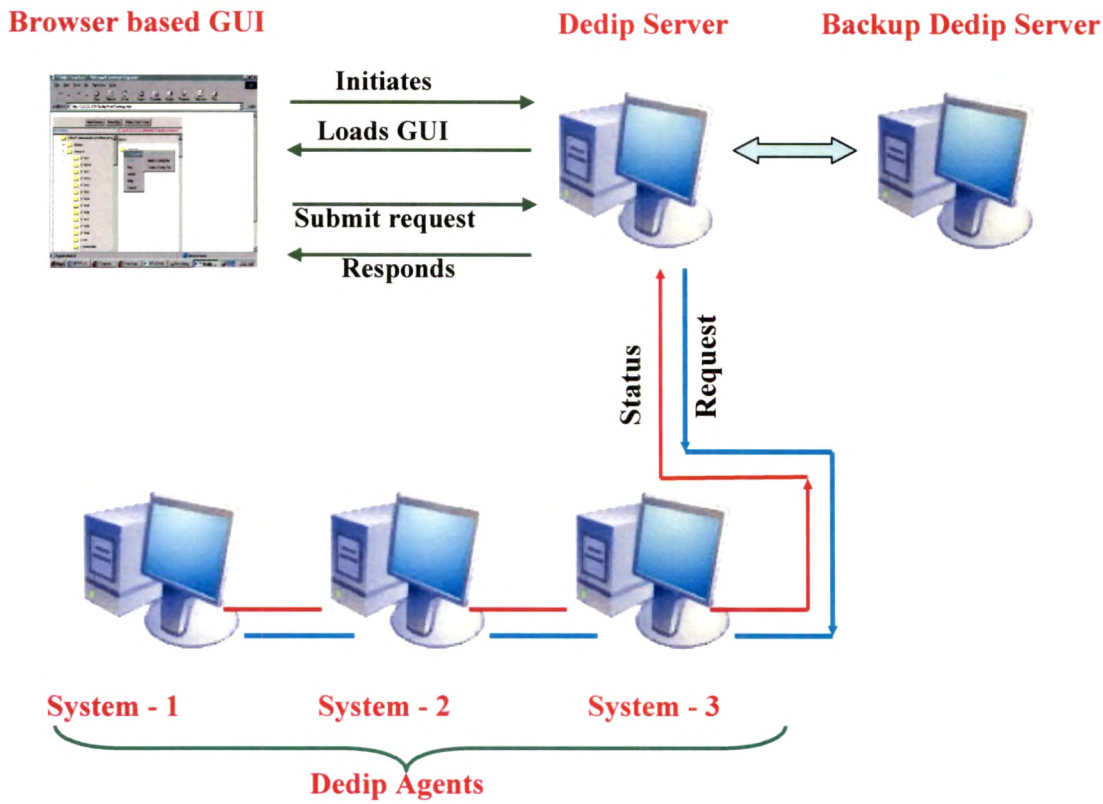


FIG. 4.1 OVERVIEW OF WEBDEDIP ENVIRONMENT

The agent accepts requests from the DedipServer, executes them and provides the status information when completed. It has process building (compilation), execution, and monitoring capabilities. It can schedule multiple processes in parallel. It does not control the synchronization among the parallel processes;

instead it depends on the DedipServer for this job. It treats each process as a single independent entity. The WebDedip not only caters to the requirements of the application designer, but also addresses all the requirements of the operation manager as well as operators.

The application configuration and building is a privileged task, carried out either by the application designer or operation manager. During the regular operations, the operator can initiate any required application, monitor progress, do error handling, and terminate the application, if necessary.

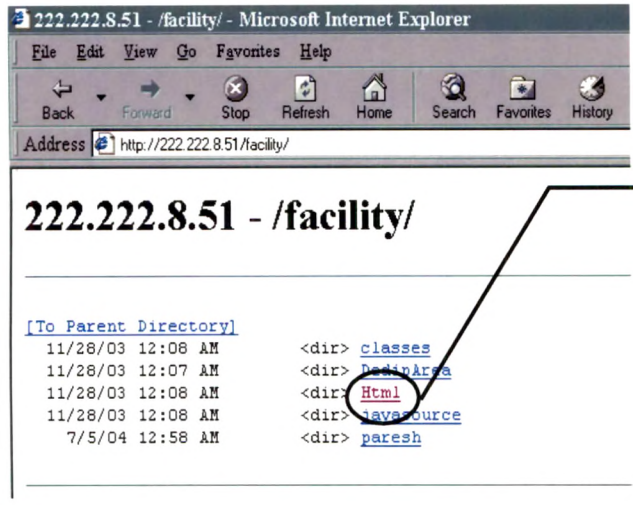
Object-oriented modeling (implemented in Java) is used for the design of the WebDedip. The application is modeled as an object while the process is modeled as an embedded object. The object inter linking capability is used to maintain interdependency information for an application. Java distributed object architecture is used along with the object serialization for network communication among GUI, DedipServer and agents. Hence, WebDedip can be used on a LAN or on Internet. Agents may run on any system over Internet. On start, an agent makes connection with DedipServer on a predefined port and volunteers for computation workload. The Windows-explorer is used as a metaphor in developing the navigation GUI due to its popularity and ease of use.

4.3 APPLICATION DEVELOPMENT USING WEBDEDIP

Implementation of parallel application using WebDedip is very simple. User needs to visualize the parallelization potential of application and accordingly divide into subtasks and define flow of data between subtasks. Use of WebDedip for parallel application development need not require any background of parallel processing. Sequence of processes for parallel implementation of an application is described in the following subsections:

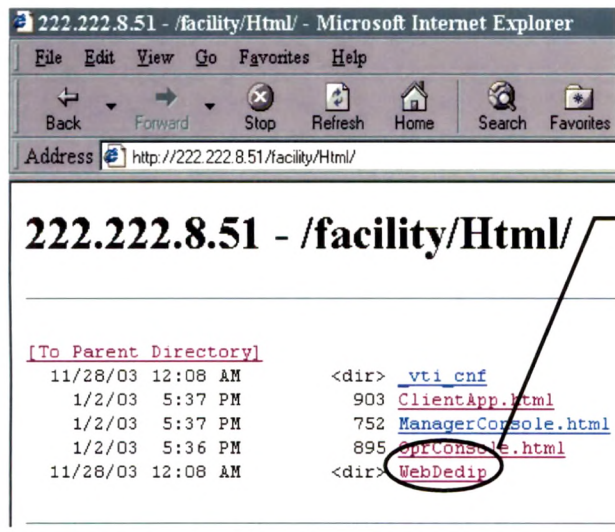
4.3.1 Load GUI

As a very first step, user has to initiate by submitting the request for loading GUI. User can load GUI using web browser. Sequential steps for loading GUI are as shown in following screenshots (Fig. 4.2).



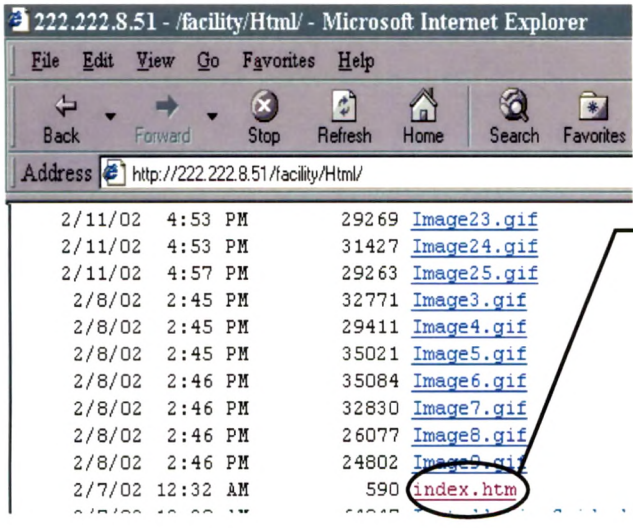
After giving the address of Web server in web browser, as a first step user has to select this option

(a)



After selecting the Html option, this page will open from which user has to select WebDedip option

(b)



As a next step user has to select index.htm option

(c)

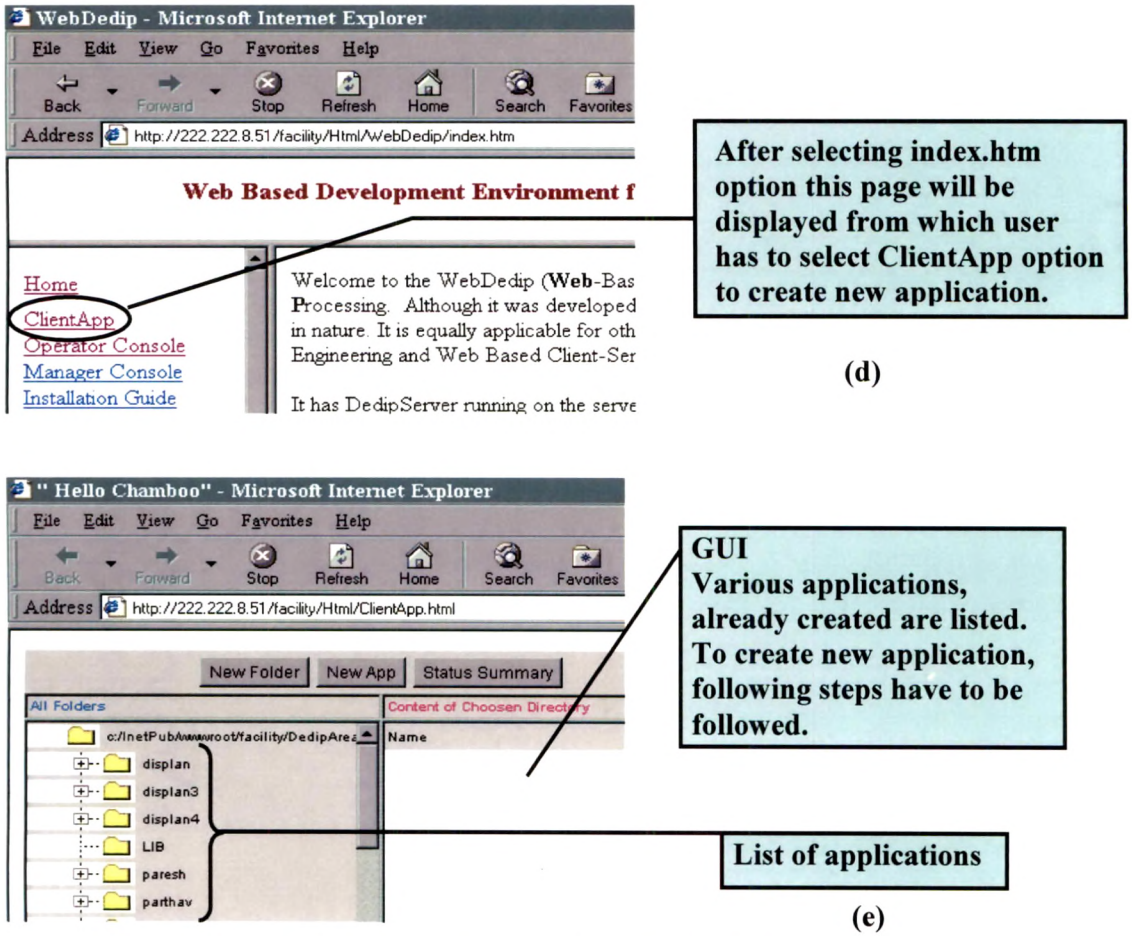
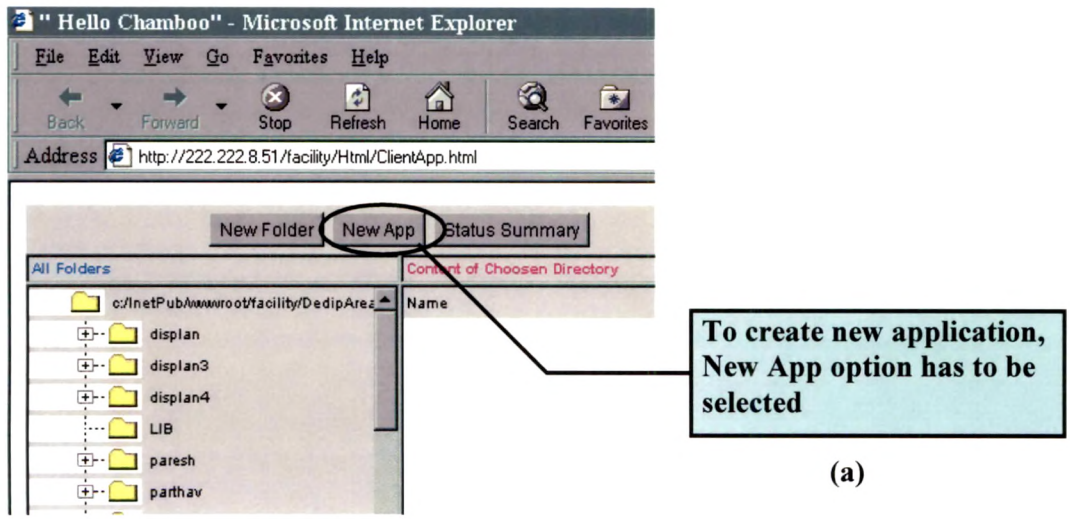


FIG. 4.2 STEPS FOR LOADING GUI

4.3.2 Creating New Application

Selecting the New App option from the client app window can create new application. Details to be filled in New App form are as shown in following figures. (Fig. 4.3 (a) and (b))



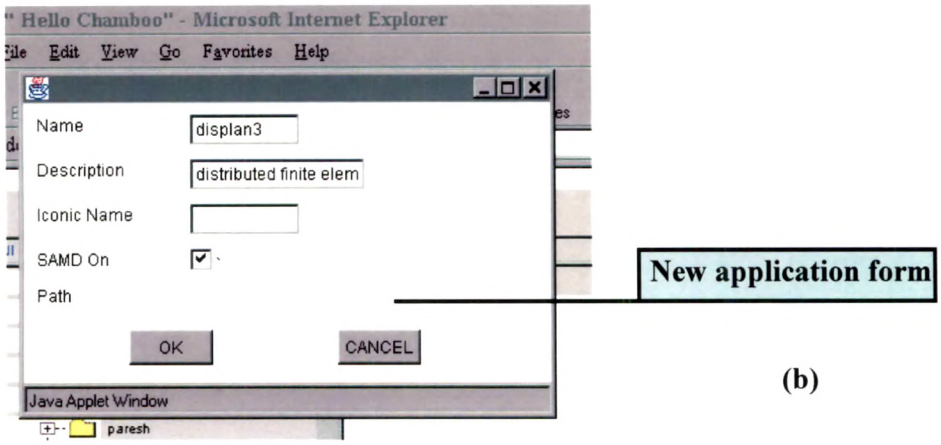
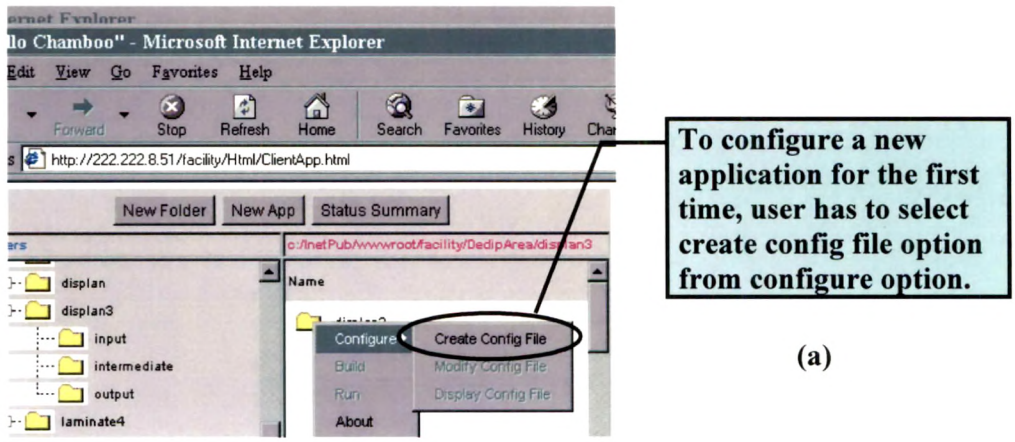
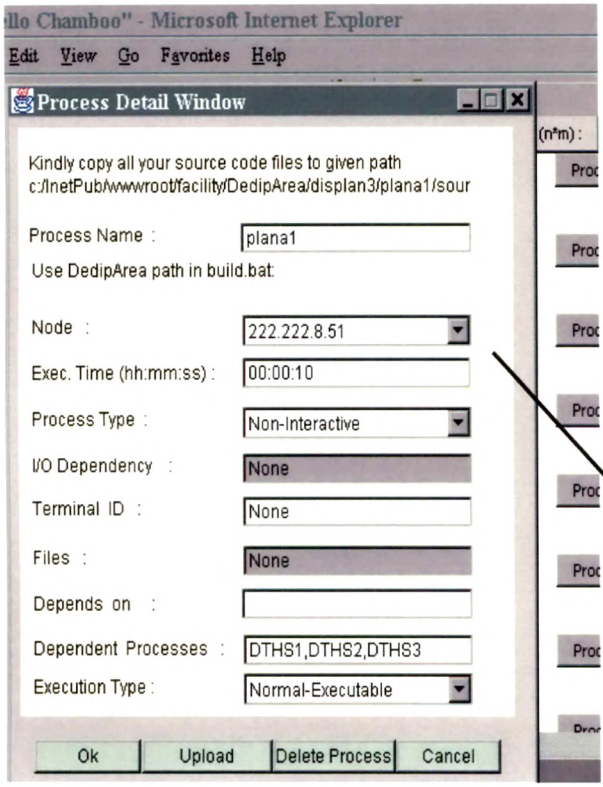


FIG. 4.3 CREATING NEW APPLICATION

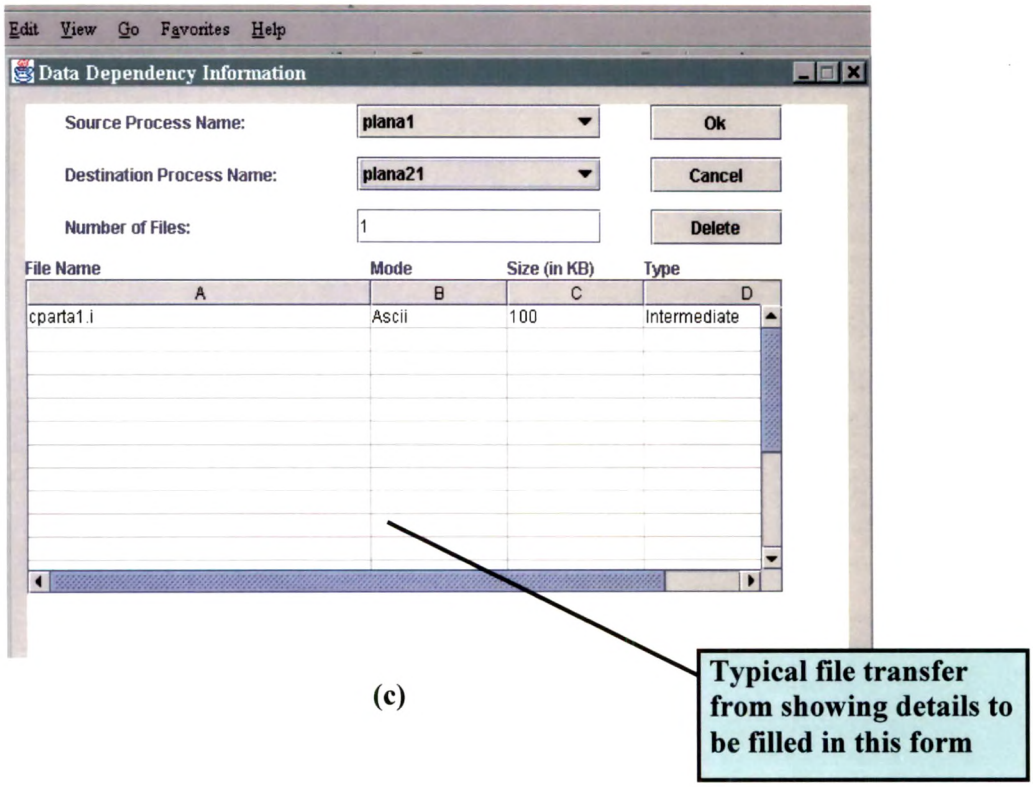
4.3.3 Application Configuration

The application designer has to first decide the configuration of his/her application. It depends on the distributed resource requirement, parallel processing requirement, input/output of each process, etc. The WebDedip supports a nice GUI for the same as shown in figures. Fig. 4.4(a) shows the overview of the application. The detailed information about each process is shown in Fig. 4.4(b). The detailed information about file transfer process is also shown in Fig. 4.4(c). The typical interdependency chart, generated interactively, is shown in Fig. 4.4(d). The line joining two processes shows their interdependency in top-down model. The input output dependency, if any, is a part of this interdependency and it can be easily configured. User can modify his application configuration file any time by selecting the modify configuration option as shown in Fig. 4.4(e). The effect of the modification will be applied on next execution of the application.




llo Chamboo" - Microsoft Internet Explorer
Edit View Go Favorites Help
Process Detail Window
Kindly copy all your source code files to given path
c:/InetPub/wwwroot/facility/DedipArea/displan3/plana1/sour
Process Name :
plana1
Use DedipArea path in build.bat:
Node :
222.222.8.51
Exec. Time (hh:mm:ss) :
00:00:10
Process Type :
Non-Interactive
I/O Dependency :
None
Terminal ID :
None
Files :
None
Depends on :
Dependent Processes :
DTHS1,DTHS2,DTHS3
Execution Type :
Normal-Executable
Ok
Upload
Delete Process
Cancel
(n*m) :
Typical process detail form

(b)


Edit View Go Favorites Help
Data Dependency Information
Source Process Name:
plana1
Ok
Destination Process Name:
plana21
Cancel
Number of Files:
1
Delete
File Name
Mode
Size (in KB)
Type
A
B
C
D
cparta1.i
Ascii
100
Intermediate
Typical file transfer from showing details to be filled in this form

(c)

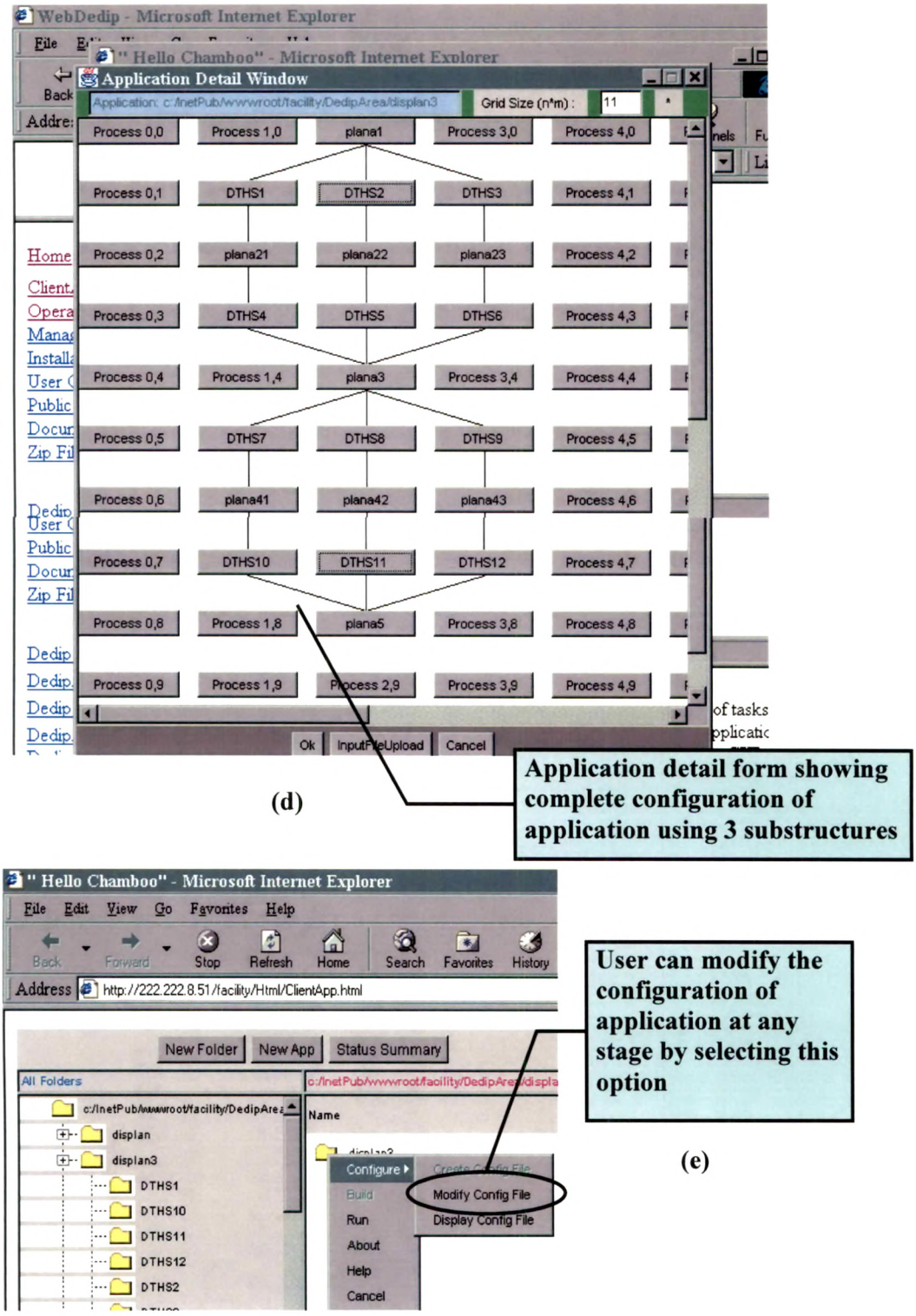


FIG. 4.4 APPLICATION CONFIGURATION

4.3.4 Application Building

An application consists of many processes. All these processes need to be compiled on the target node. The WebDedip has facility to automate all these

compilation. The configuration information has all the required details about each process. The DedipServer copies the source code and make-file (Fig. 4.5), required to build a process, on the target node in a predefined temporary area. It then requests the agent on the node to build the process using the make-file. It carries out this task for each process given in the configuration. The agent creates designated directory and preserves the executable in it. The application designer can build the processes externally on all systems in case he is not willing to give the code. The GUI provides necessary support for such external readiness indicator.

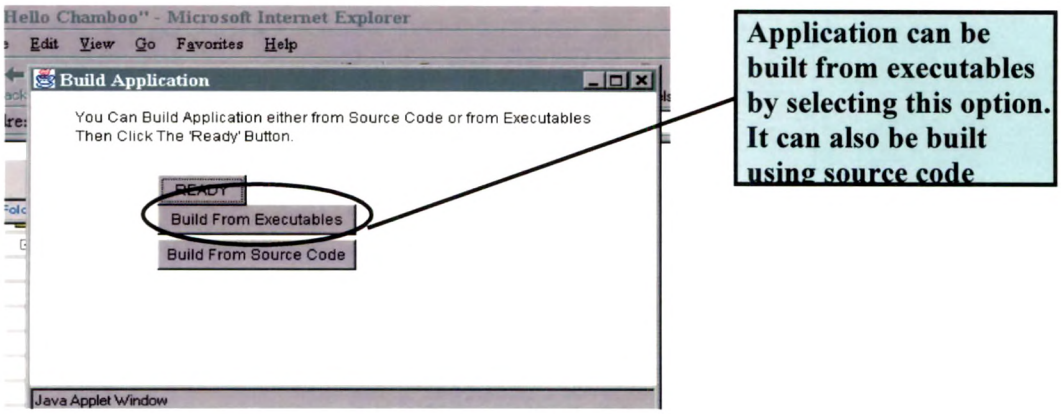
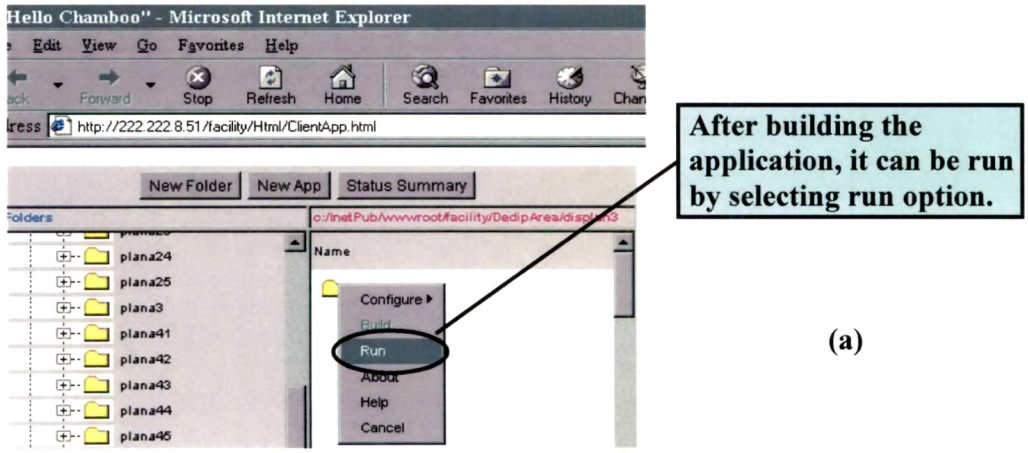


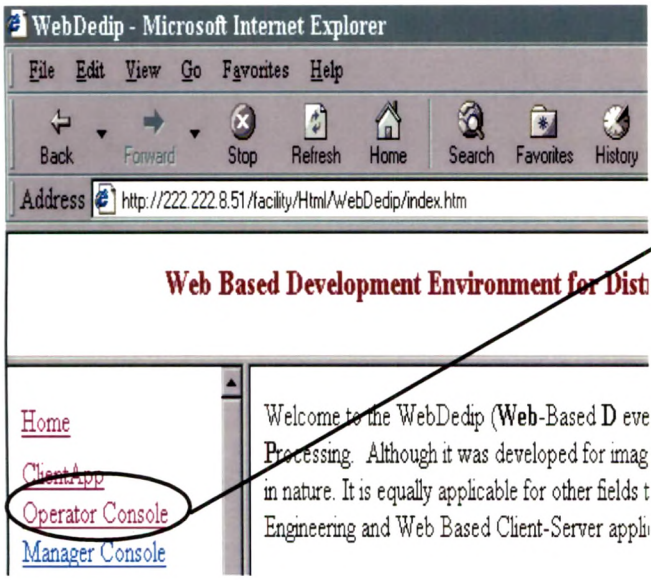
FIG. 4.5 APPLICATION BUILDING

4.3.5 Application Execution and Monitoring

The operator can start execution of any application from any machine on the net using the standard browser. GUI displays the configured applications to the operator for selection. Operator can start/abort/suspend/resume an application. Figures show the GUI for session and application progress information.

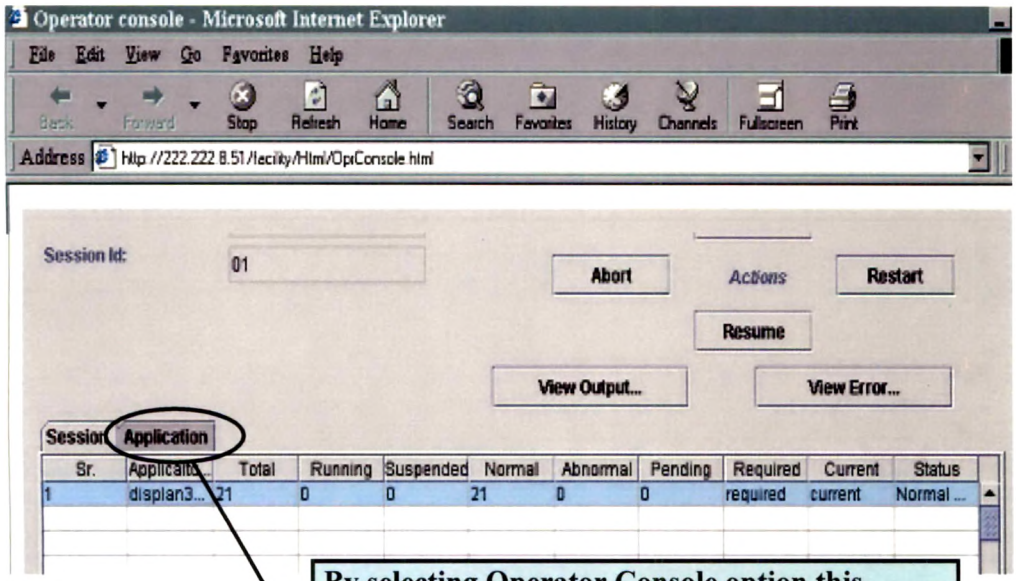


(a)



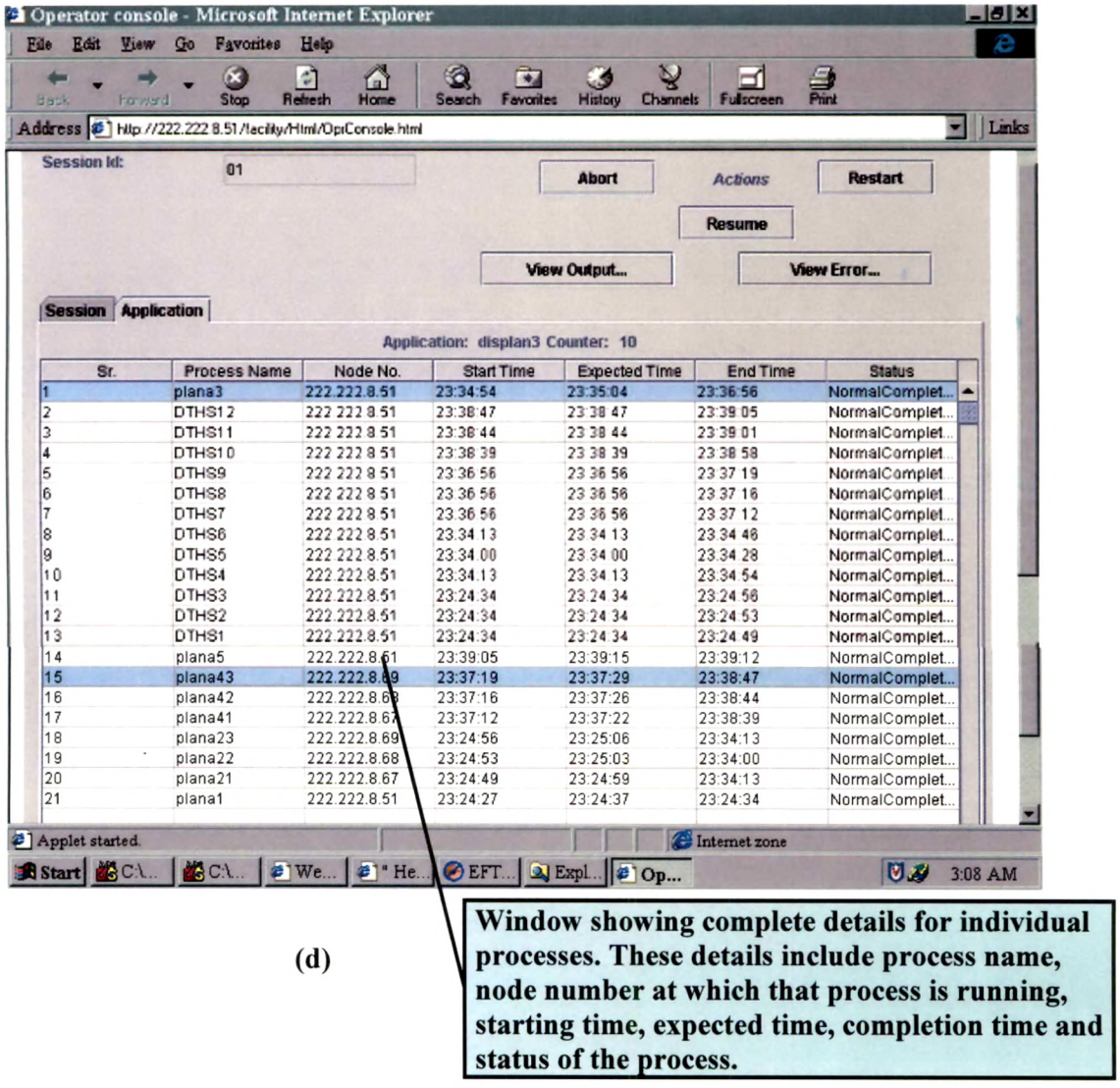
After completion of process successfully, time required for individual processes can be noted by selecting Operator Console option from main window.

(b)



(c)

By selecting Operator Console option this window will be open, showing overall details of the latest process run. Individual process details can be viewed by selecting Application option in same window.



(d)

FIG. 4.6 APPLICATION EXECUTION AND MONITORING

4.3.6 Error Handling

In case of abnormal completion, the DEDIP Server displays the error message with error code to the operator. Application designer provides these error codes and error messages. WebDedip keeps this information in the configuration file. The operator can restart the process after taking the necessary actions. In addition, the operator has the options of either restarting the entire application or aborting it.

4.3.7 Session Management

Each time an operator logs in, DEDIP scheduler starts/restarts a session for him. Each session has a unique session identification number. It keeps all the

information about the session on the server. The operator has multiple options to log out. He can close the session, terminate the session, suspend/resume the session, or submit the session for progress in background before logging out. He can close the session only after normal completion of all the requests he has submitted. He can also terminate the session immediately in case of emergency. In case of termination, the WebDedip kills all the processes of all the requests submitted by the operator irrespective of the status. The background processing is very effective in the case of non-interactive processes. The WebDedip gives the detailed status to the operator at the next logon.

4.3.8 WebDedip System Management

The WebDedip system consists of a DedipServer and agents. The DedipServer can detect the agent termination. It displays the message on operators' console as well as on operation manager console. The DedipServer is the most important process in the entire system. Its failure, for example, due to system crashing, can cause a severe problem. Dedip Backup Server is designed to handle the failure of the DedipServer. The software package Dedip backup Server runs on the machine of the backup server and duplicates the required information from the DedipServer. An agent sends a trigger to Dedip Backup Server when it fails to communicate with the DedipServer. The Dedip Backup Server validates the DedipServer failure. It takes over the complete responsibility from that moment onwards and informs the operation manager. The servers are exchanging information only in case of external events like termination of process, start of new process, initiation of an application by the operator, the start of new session, etc. The frequency of such possible events is very low. Furthermore, the volume of the information is negligible. Hence, the communication overhead for maintaining the back-up server is very low.

4.4 INSTALLATION OF WEBDEDIP

WebDedip environment can be installed on any network of computers. In present study it is installed over LAN. In the installation one computer is configured as server and other computers are clients. The requirements of server machine and client / slave machine are described in following subsections:

4.4.1 Server Machine

It should be higher than Pentium-1 with 64 MB RAM. Dedip itself does not require any significant disk space however it requires disk space in the Dedip area as per the respective applications requirements.

The operating system of server can be either NT Server or Window2000 Server. It is running software like JRE (Java Runtime Environment) and JDK (Java Development Kit) if source code of WebDedip server is to be compiled. Web Server IIS (Internet Information Services) is to be run on server. Jakarta Tomcat server is to be installed for uploading source code. FTP server EFTP (Encrypted File Transfer Protocol) is to be installed on server for file transfer between computers.

For WebDedip environment Dedip server needs to be installed on server in addition to Web Server, FTP server.

Web server configuration: User has to set web root path. It is preferred on drive other than OS drive. So if drive D is used for web root path then wed root folder will be located at D:\inetpub\wwwroot\.

FTP server configuration: To configure FTP server first of all it has to be installed by running eftpserver.exe file. After completion of installation process, user has to create user account. Here user name is WebDedip and password is 'haresh0'. To set home directory user has to select D:\inetpub\wwwroot\facility\dediparea folder.

DedipServer Installation: DedipServer is developed using Java. DedipServer has following components:

1.) Dedip packages
    Dedip
    Ftp
    Sharable Objects
2.) Dedip configuration files
3.) Dedip area
4.) HTML files

Various Dedip packages are installed from class files. Following steps can configure these packages:

1. Create "facility" folder under web root directory (D:\inetpub\wwwroot)
2. Create "classes" folder under "facility".
3. Extract DedipClasses.zip to D:\. Path information is saved in the zip file.

Hence, following directory structure will be available.

D:\inetpub\wwwroot\facility\classes
SharableObjects – containing class files
ftp – containing class files
Dedip- containing class files
Icon- contains imported image files
Config-contains two important configuration files.
-dedip.inf
-dediparea.inf

4. Create "DedipArea" folder in webroot directory\facility. So it will be at D:\inetpub\wwwroot\facility\DedipArea
5. Extract Sample test application from DedipArea.zip to D:\. Path information is saved in the zip file. Hence, it will copy the files under home directory. If user doesn't want to extract sample test applications, he/she has to extract at least following:
   D:\inetpub\wwwroot\facility\DedipArea\SessionInformation
   D:\inetpub\wwwroot\facility\DedipArea\lib\fileopen.c
6. Open dedip.inf, located at D:\inetpub\wwwroot\facility\classes\Dedip\config
   - The first entry defines the DedipArea path. The default value is set to d:\inetpub\wwwroot\facility\DedipArea
   - The second entry defines the Dedip Icon directory. The default value is set as facility/classes/Dedip/icon
   - The third entry is the server IP address.
   - The forth entry is the Backup Server IP address.
7. Open dediparea.inf, located at D:\inetpub\wwwroot\facility\ classes\ Dedip \config.
   Each row contains following three fields
   - The first field is the machine IP address

- The second field is the physical location of DEDIPAREA
- The third entry is the operating system of the machine.

To configure HTML files, user has to extract DedipHTML.zip to D:\. It will create following directory structure.

D:\inetpub\wwwroot\facility\html
D:\inetpub\wwwroot\facility\html\WebDedip

4.4.2 Slave Machine

It should be higher than Pentium-1 with 64 MB RAM. Dedip itself does not require any significant disk space however it requires disk space in the Dedip area as per the respective applications requirements.

The Windows operating system is preferred. It is running software like JRE (Java Runtime Environment). FTP server EFTP (Encrypted File Transfer Protocol) is to be installed for file transfer between computers.

For WebDedip environment DedipAgent needs to be installed on slave machine in addition to FTP server.

DedipAgent is developed using Java. It has following components.

1) DedipAgent packages
   Dedip
   ftp
   SharableObjects
2) Dedip configuration files
3) DedipArea

Various DedipAgent packages are installed from class files. Configuration can be done in the same way as it is for DedipServer on server machine.

4.5 IMPLEMENTATION OF APPLICATION ON WEBDEDIP

Various steps to be followed to implement an application on WebDedip can be summarized as follows:

1. To start the WebDedip Session: To start the session, user needs to start DedipServer on server and DedipAgnet on all slave machines. If BackupServer is made then it also has to be start.

2. To start DedipServer: The DedipServer has to be run on Server machine. It can be run from DOS prompt. In DOS prompt following command has to be given to run DedipServer.

   cd D:\inetpub\wwwroot\facility\classes

   java –classpath . Dedip.DedipServer

3. To start DedipAgent: User needs to start DedipAgent on all slave machines. Same as DedipServer, DedipAgent can also be run from DOS prompt. Following command will start the DedipAgent.

   cd D:\inetpub\wwwroot\facility\classes

   Java –classpath . DedipAgent.DistScheduler

4. To start DedipBackup server: If Backup Server is made then Backup server can be run from DOS prompt by following command.

   cd D:\inetpub\wwwroot\facility\classes

   java –classpath . Dedip.DedipServer –Backupof "SERVER_IP"

5. To configure new application: New application can be configured by following steps:

   (i) To configure new application user has to open WebDedip home page (http://SERVER_IP/facility/html/WebDedip)

   (ii) Click on the ClientApp. It will open the GUI in new window.

   (iii) Click new application button. It will open a dialog box. In which user has to give application name and description. Select the SAMD option. After clicking OK, user can see the application folder on left hand panel of GUI window.

   (iv) On clicking the application icon, it will appear in the right hand panel of GUI.

   (v) Right click on application icon will show a popup menu. From that menu user has to select config-> Create Config File. It will display a dialog frame with button matrix. The default size is 11X11. User can change the size of matrix by entering the values in matrix field.

(vi) When a first row button is clicked on the button matrix window, a dialog box will appear, showing option for Process Detail Form and File Transfer Form. First user has to select Process Detail Form. This will open the Process Detail Form. Various process details such as process name, node on which the process will run and expected time for process has to be filled in this form. After completing process detail form, for first row, another process form has to be filled in third row of matrix. One row should be left between two processes for file transfer process. After filling details for processes, user has to give file transfer information. This information includes, the source and destination process, file name, approximate file size and type of file (i.e. input, intermediate or output). In this way the basic structure for the application will be ready. Now application is ready to built.

6. To build the application: Right click on the application icon will show the popup menu, having option build. Selection of build option will show a new window. User can build the application either from source or from executables. In case of build from executables, the Dedip will transfer the executables on the target node in appropriate directory. In case of build form source code, the Dedip will transfer the complete source code directory on target node and ask remote DedipAgent to compile code using buil.bat file. Once application is built successfully, user has to click ready button in build dialog box.

7. To run configured application: After building application successfully, the application icon will disappeare from right hand panel. So on clicking the application icon on left hand panel of GUI, again it will appear in the right hand panel and right click on application icon will open a popup menu containing an option for run. Selection of run option will start execution of application as per configuration.

8. To monitor/control the progress of running application: The status of the running application can be known by opening the WebDedip home page (http://SERVER_IP/facility/html/WebDedip). From home page user has to select Operator Console. It will open an operator GUI in new window. Session tab has to be clicked to get summary report of the session. Selection of

application from the table and then application tab will give the details about application status. User can abort/ suspend/ resume/ restart an application. To do so user has to click on the session tab. Select the application and then click button of desired action.

9. User can terminate the DedipServer using manager console. It provides three level of termination.

   (i) Immediate: It kills all processes running on different slave machines.
   (ii) Process wide: It waits for all processes currently running on different slave machines. Then it terminates. It disables the DedipServer to take new application for processing during the period.
   (iii) Application wide: It waits for all processes currently running and pending for initiated application. Then it terminates. It disables the DedipServer to take new application for processing during the period.

The DedipServer is capable of restarting the previous session from the point of termination so that loss of processing power is negligible. It can restore the session even in case of Server System crash.

4.6 WEBDEDIP RULES

When one runs his/her application through terminal window, it is known that where are input, intermediate and output files. WebDedip has to handle the same. So WebDedip suggests to use input directory for input files and intermediate directory for intermediate and output files.

Furthermore, user may wish to create multiple instances of application (i.e. run his/her application more than once simultaneously on different input data). This mode is called Single Application Multiple Data (SAMD) mode. As intermediate files are created in intermediate directory, files will be overwritten by another instance of the application. It is very difficult for user to handle such case.

WebDedip provides a simple solution for this problem. User has to follow the following steps:

1) Insert the following code just after the include statement

```
FILE *fileopen();
```

2) Insert the following three lines at the beginning of main function

```
Strcpy(DeipArea,argv[argc-3]);
Strcpy(Application Name,argv[argc-2]);
Counter = atoi(argv[argc-1]);
```

3) Use "fileopen" function provided by WebDedip to open file instead of stdio.h routine. The fileopen.c is lying at DedipArea\lib on all slave machines.

For example,

```
f1=fileopen("Testi.inp","r",DedipArea,ApplicationName,counter,"input");
f1=fileopen("Test1.","w",DedipArea,ApplicationName,counter,"intermediate");
```

The first two arguments are the same as that of "fopen". The next arguments are that are supplied by WebDedip to user's process as argument to main function. The last argument mentions whether the file resides in "input" or "intermediate" directory.

The "fileopen" function does the following:

1) Prefix the "DedipArea/Application /input" to file, if last argument is "input".
2) Prefix the "DedipArea/Application /intermediate" to file, if last argument is "intermediate". It also concatenates "_XX" as the end of file name. Where XX represents the number of execution of application. Means user can create 99 instances of his/her application simultaneously.

4.7 ADVANTAGES AND DISADVANTAGES OF WEBDEDIP

Ease of parallel application development using network of computers or LAN without any additional resources is the main advantage of WebDedip. User without any background of parallel processing, operating system or networking can utilize this environment. Visualization of parallel application and decomposing the applications into different tasks, which can run independently on different set of data, are required for distributed computing using number of computers. These computers are general purpose and can be utilized by other users as processing is done in background.

The data to be communicated between computers is written into files, which is subsequently transferred by FTP. This process increases overhead, which reduces the speedup and consequently efficiency. Static load balance can be achieved by considering processing speed of computers, number of computers available and size of problem. But if any computer is engaged into some other computational process by other user, transfer of work load to other relatively idle computer, known as dynamic load balance, is difficult to achieve. If network is busy during peak hours by users, data transfer may cause problems sometimes.