**Video Conferencing With Gstreamer**

Sahil Chug

July 2011

**Contents**

**Chapter 2**

**SOFTWARE REQUIREMENTS AND**

**SPECIFICATION**

* 1. **Functionality**
  2. **HARDWARE REQUIREMENTS**
  3. **SOFTWARE REQUIREMENTS**

**Chapter 3**

**DESIGN**

**3.1 Design OverView**

**3.1.1 Client/Server Model**

**The Application is based on a Client/Server Model.**

**The Server is Listens For Clients to Connect to it,**

**as soon as the client connects the server passes**

**on the connection to another thread , so that server**

**is able to host more than one clients simeltaneously.**

**3.1.2 Server**

**The Server is a Multithreaded Server Which Can**

**handle Multiple Clients at same time. Server has**

**basically three jobs to perform .**

To Maintain Connection With Clients

Message passing between two Client is Mediated

through Server

A seperate Server thread will continuesly update

names of all the clients present online and send that

list to all connected clients after certain interval of

time on regular bases.

Server uses Python Select Module which on connection

by client will move client connection on a seperate thread.

Then that particular thread will serve client by entering into

an infinite Loop , it keeps listening for any messages from

client side to process.

**3.1.3 Client**

**Client side is where all the main functionality is**

**being handeled. The client has a GUI layout to**

**which user interacts , whreas behind the GUI**

**there are two threads working.**

**A networking thread.**

**A reciever thread.**

**In the main GUI thread Client firstly clicks on**

**the connect button which starts the networking**

**thread which connects the server.**

**After the connection is established the networking**

**thread goes into an infinite Loop which will listen to**

**any messages sent from server and will send**

**appropriate messages to the server according**

**to events taking place .**

**The main thread is provided with capability of**

**streaming the Webcam to other clients on request**

**sent by other clients.**

**The Reciever thread on other hand is responsible**

**for handling the incoming stream from other client.**