USER-PRESENCE AWARE SYSTEMS

Project Group:
Ch. Abhishek 2009H112122P
Nitin Jose 2009H112116P

IMPLEMENTATION PHASE:

The design has been depicted in the form of Functional block diagram and Process Flow diagram presented previously. The design has been further enhanced considering Security and Accuracy into account.

The main modules as identified are:

1. Bluetooth based locking/Unlocking
2. Detection of Proximity
3. Streaming of media

Technical Specifications:

Implementation Environment: Linux-Ubuntu 9.04 and 10.10
Coding language: Python 2.6
Packages used: Bluez, Pybluez , vlc-plugin-pulse, libavcodec
Technology chosen:

- Bluetooth: For user presence awareness, between mobile(carried by user) and laptop
- LAN or WiFi : For streaming of data

Streaming server: Vlc 0.9.9: Streaming of audio and video done through it.
Streaming client: Vlc 1.1.4

The main modules and their implementation is elaborated here:

1. Bluetooth based locking/Unlocking:

Motivation:

The users of any system have to generally type in their credentials to get access to the system, this process needs to be repeated each time the user leaves the system and comes back to his workstation. This whole task can be avoided by using something which everyone carries along with them, the mobile phone further bluetooth has become a common feature available on most of the phones. It has become ubiquitous and inexpensive over the last few years. This technology can be used to avoid the users the effort and time wasted on accessing a system. The Bluetooth device becomes the way of identifying the user’s presence.

Functionality:

This module deals with the main aspects of the system. The functionality is to pair with a particular user, detect his/her presence through bluetooth signal from his/her mobile phone, if the system is in “Lock Screen state” upon detection of paired user(device) the system will automatically unlock the screen and make it usable to the user directly without the need for entering password manually. When the user leaves the range of the system the system automatically locks the screen. The security has been taken care by allowing only root user to change the paired device statically.

Code: Filename: “lock_unlock_machine.py”
Implementation command: $python lock_unlock_machine.py

Implementation details:
The code has been structured to continuously detect the available Bluetooth devices in its range. If it detects a device using the discover_devices( ) , once the device identified. The entire list of devices identified maintained in nearby_devices( ) are compared with the device with which it is paired with as specified in target_address . If the device is the required device and if it leaves the range then a subprocess is called invoking lock and the launch of screensaver. This is done through subprocess.call( ) .
Further the continuously checks for nearby devices to find out whether the target_device is back in the ranger, upon which it unlocks the system in a similar mechanism.

Run the code through terminal and allow it run as long as the system is on.
Implementation command: $python lock_unlock_machine.py

2. Detection of Proximity:

Motivation:

The location of the user obtained through Bluetooth device inquiry can be in the range of 10meters here, but such levels of precision is sometimes not sufficient as the sytems for streaming of application might have overlapping zones. So some kind of mechanism needs to be devised to improve the accuracy. The signal strength gives a reasonable indication of the position of the user about desirable for our application. Thus RSSI has been used.

Description:

This module has major usage in other two modules. The functionality is to detect the proximity of Bluetooth device thereby allowing accurate implementation of other two modules. It considers the strength of the Bluetooth signal received, analyses it over few samples, maps it based on observation into distance or distance zones broadly. The module of Received Signal Strength Identification (RSSI) has been used in achieving this.

The rssi modules base functionality is taken from python libraries. It has been modified in accordance with our specific requirements.

RSSI works by first pairing with the specific device which is being surveyed for proximity. The signal received is further analyzed and it produces us with a number which is an indicator of distance. The values obtained don't give accurate results but it can give results which helps us map the distances into zones and use them for our application.

3. Streaming of application:

Motivation:

In today's world it has become common for people to have multiple systems in their homes and further they all are set up into a network either through LAN or wi-fi, in the worst case they might be connected through internet. Further generally people tend to play music or access data on one system and tend to move around the house simultaneously taking care of daily chores. Thus if a user is playing music on system and moves into another area he cannot enjoy the music despite having a system in his vicinity as the music is being played on another system. This lead us to this module which helps to make use of the available vicinity to access data such audio or video. As already mentioned the ubiquitous nature of the Bluetooth is used here coupled with availability of the network through LAN or Wi-Fi.

Functionality:

This helps in transfer of application, it is a method to use the users distance from a pc or laptop to transfer running applications to the nearest system. The user moves from one area into another within his/her house the data which the user was accessing on one system can then be shifted to the system closest to him using the Bluetooth phone to detect his location falls under which systems range.

Code filename:

"streamer_stream.py"
"streamer_slave.py"

Note: Please use version specific code.

Implementation details:

The device_inquiry_with_wthe_rssi( ) module is to perform a simple device inquiry, followed by a remote name request of each. The module has been taken from the python libraries and modified and further functionality added to meet our requirements. Pyrssi is developed by MIT. check_signal_strength()is a function that checks the signals being received. For accuracy it done three times, returns a value indicating its distance or 0 if it is not available.

Initialize() starts the process of looking for the device, if found it launches the vlc done by launch_vlc(), upon leaving the specific system in which streamer_stream.py is running it ends the music it is playing and streams the music onto the system where the Bluetooth device comes into range. Thus the process repeats continuously shifting the application in this music in accordance with the user.

Implementation commands:
Run the command $python streamer_stream.py in the stream server system
Run streamer_slave.py in the device where the user might move into

REFERENCES:

1. Bluetooth Essentials for Programmers - Huang and Rudolph.
2. Project Oxygen, MIT. *(inspiration)*
3. A Byte of Python
4. A tutorial Network Programming with Python, N Matloff, UC at Davis.