Working Paper Series
ISSN 1177-777X

SOCIAL INTERACTIONS USING AN ELECTRONIC RABBIT

Alexandru Calin Zaicu and Annika Hinze

Working Paper: 09/2013
December 2013

Department of Computer Science
The University of Waikato
Private Bag 3105
Hamilton, New Zealand

1. Introduction

In this project we use an electronic rabbit called Karotz, created by French company Violet. The rabbits have the ability to connect autonomously to a WI-FI network. IN this project we use Karotz to record an audio log that will contain sounds of the environment. We also programmed a way for the rabbit to send audio to its other Karotz friend. We explored if Karotz can be used to help people stay in contact with each other and to feel less homesick.

1.1 Scenarios

We created scenarios to show how Karotz might be useful in different situations.

Domestic connection. An example would be, Peter and Tanya who are a couple. Peter usually works long nights and would like to have some way to check up on his partner. Luckily he now has Karotz. After turning it on with the long press of a button he can get audio feedback from Tanya. Peter can also just short press the record button and he will be able to record something for Tanya. Once he is finish she will hear the message played back automatically.

International connection. Another example, let's say Anna has moved to the UK and all her family is still here in New Zealand, she has quite a hard time adjusting because she is really homesick. In this situation Karotz really shines, all she needs to do is start Karotz and listen to the recordings that her mom made for her. She can also remote trigger Karotz to record so her mom does not have to do anything. The best thing is that Karotz will play back the audio track in Anna's own time so she will not miss anything.

Local with specific interaction. Another example would be, Dan who is at work in a different city and he misses his daughter Kelly who is only 2 years old. He is able to record an audio log for Kelly with her favourite bedtime lullaby because it is her bedtime. Dan can also remotely trigger Karotz so he can listen to Kelly to make sure she is ok and see what she is doing.

International connection with specific interaction. Let's now say that Anna and her mum are on Karotz at the same time. So now they can either take turns or they both record audio logs at the same time and Karotz will play it back to them after the recording is done. So they can chat to each other or keep audio and listen to it later in their spare time. Now Jim, Anna's brother wants to hear some of the audio logs that his mom recorded too so all he has to do is turn on Karotz and connect up with mum and Anna and he will get back all the stuff recorded from both of them.

1.2 Project Overview

The remainder of this report is structured as follows:

In the second chapter we evaluate the previous work that has been done to gain a better understanding of other approaches. We then present our research plan.

In the third chapter, we introduce details of the Karotz Operating System as well as give some information on background for a better understanding of how Karotz is supposed to work. In this chapter we also included some communication diagrams.

In the fourth chapter, we present details of the application as well as give an in depth report of the problems encountered. We explain how we set up a dedicated server for the Karotz application to run properly because Karotz only has 55 Mb of internal storage.

In the fifth chapter presents our user study. This way we will be able to analyse the data received from then and come up with a conclusion if the application would be a success in helping people feel closer to each other or not. The data entries and questionnaires are available from the University of Waikato in the FCMS data archive.

In the last chapter draws a conclusion from the user study data and present suggestions for future work continuing or improving this application.

2. Previous work done on the subject

We researched previous projects and discovered a number of similar projects Since we have quite a lot of material we will only briefly present some of the projects that we found really interesting. If the reader requires or wants to read the whole publications please refer to the reference page for all of the information about the publications.

The first one it the original inspiration for our work: “Living with The sound of the Past: Experiencing Sonic Atomic Interaction Using the Sound Diary” written by Hanif Baharin & Ralf Mühlberger. They have developed a system that records sounds. It gives another user the possibility to play it back after some time has passed. So in this article the authors are looking in recording audio logs and getting a person to listen to them after some time. It is really interesting to listen to past experiences as you can recreate the image an image and visualise the location in your mind. This is why we feel that this will be similar to my project, because the participants will be using Karotz.

The same authors have published another article “Utilising the Open Channel Created by Telecare: Acoustic Communication Analysis of Radio Sounds at Home”. This article shows a nice explanation of how they have built an internet radio that plays every day sounds. This is similar to my project because the authors talk about how he can use a device to play back sounds using the internet as a medium. So although it is really similar to my project there is a significant difference as we use a device that connects automatically to a server and it can blend in the environment. Also we will be using some other cool technologies like RFID tags to launch the application or voice recognition.

Another interesting one is in the publication “SWEETPEA: Software Tools for Programmable Embodied Agents” written by Michael Kaminsky, Paul Dourish, W. Keith Edwards, Anthony LaMarca, Michael Salisbury and Ian Smith . In this article they are using all type of interesting dolls to help people have a better work area. They use these toys to interact or communicate to the user. This is similar with my project by having an avatar that represents a memory. We want to make Karotz an avatar that will give the possibility to people to remember certain locations.

Another example is the competitor to Karrotz/Nabztag rabbit there is a little Linux mascot called Tux Droid. The developers of the tux droid were planning on creating a Skype app that is slightly similar to what we am attempting to achieve. For the Karotz/Nabztag there have not been any applications to record and transmit environment noises. This is probably the most similar device to what we have to work with. Unfortunately they have discontinued it in august 2010 when the company that makes them Kysoh field for bankruptcy.

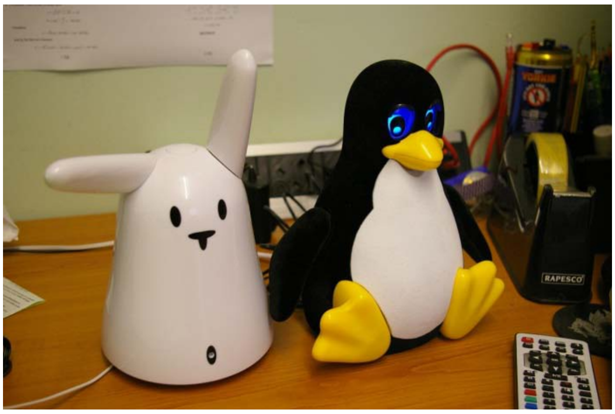

Fig. 1 Nabztag (the old version of Karotz) next to tux droid

Another really interesting project is developed by Avocado Software Inc. The application is called Avocado and is sold as an application to keep people together. It enables users to send pictures, messages even remote hugs and kisses to each other. This is a really good idea because a person will have their cell phone everywhere with them so this will give them to possibility to stay connected with their love one. This application is really nice but it is quite different from my project because they are not using any audio recordings. My application will allow the users to imagine themselves in the location where their friends are.

One more really simple project using the internet and a lot of HCI studies is “Communicating Intimacy One Bit at a Time” by Joseph ‘Jofish’ Kaye, Mariah K. Levitt, Jeffrey Nevis, Jessica Golden and Vanessa Schmidt. They have made an application that stays on the windows start bar near the clock and gives the possibility to get some useful information to the user about if there partner is close to computer. It also gives the possibility to give a bit of information to the partner by clicking the “Virtual Intimate Object”. Once clicked this will change the other VID into a different colour or shape. This project is similar to what we attempt to achieve but it is really simple and the data that is transmitted from one user to the other is not enough to make them feel closer.

Another project is “Sharing Empty Moments: Design for Remote Couples” by Danielle Lottridge, Nicolas Masson, Wendy Mackay. In this project the authors look at some interesting new ways of getting people that are far apart to somewhat communicate with each other. One of the idea is to have a frame that lights up when the person at the other end touches it or another idea is to use vibration cups so they vibrate when the person on the other end touches it. This project uses some interesting features to try and make people closer together but we feel that the sound in Karotz would make the experience a lot different from this.

All of the related work projects that we have found have some features that are similar to what we need but none of them are exactly the same this is why we feel this project will really help people that are feeling homesick.

3. Background and research on Karotz

First of all there has to be a clear idea of how the Karotz operating system works. After checking this at the violet website we have discovered that before programming, you are required to put in a description of the application as well as comments. This has to be done before the application is even started.

3.1 The Karotz Operating System

After researching documents about Karotz, it seems that it has a cloud Operating System. So any application needs to be installed online and then it gets send to Karotz.

For development Karotz gives you two options.

The first option is API REST which is a full online system that remotely manipulates the Karotz using the Violet server. This option seems quite straight forward as all you have to do is initiate a session between the remote application and Karotz. This is done by using an interactiveID().

Once this has been crated we can use REST functions to control Karotz. After more research on the subject, it seems that for some unknown reason the record function in the REST API is stroked out. This means that It might be difficult to achieve the main goal of this application and that is to record audio using Karotz.

The second option is to use JavaScript, this enables you to use the application on Karotz locally so this would have a slight advantage over the API REST because there will be a lot less network speed problems or drops. It seems that the functions are quite straight forward. The record function for this seems to be ok so it probably is not deprecated.

We also discovered that you have to create and describe an application before you are allowed to develop one no matter what option you chose. This is really bad practice in a way because you may want to test and experiment before actually going on a full on application. So before we started making the application , we had to write a description, help instructions, decide on what market to put the Karotz app and make an Icon for it.

3.1.1 Communication Diagrams and planning

Because there will be a lot of things communicating with each other to make Karotz work there has to be a diagram showing an in depth analysis of what is happening. So we have made up to diagrams that estimate how the devices interact with the server and with each other. The first 2 diagrams (fig.3-4) show the JavaScript local application, as you can see the interaction between the devices is a lot simpler. The 3 diagram (fig. 5) is of the API Rest method, this is a lot more complex because we will have to create interactions between two servers. The biggest problem there is not the interaction but is the actual distance of the server. One of the servers that are programmed by me will be located here in New Zealand and the other one is the Violet server. The Violet server is the manufacturers’ server and is locate in France. It has to act as a middle man in the API Rest method of programming. So as you can imagine there will be a significant network lag when sending commands over the internet from New Zealand to France.

JavaScript (on the Karotz application)

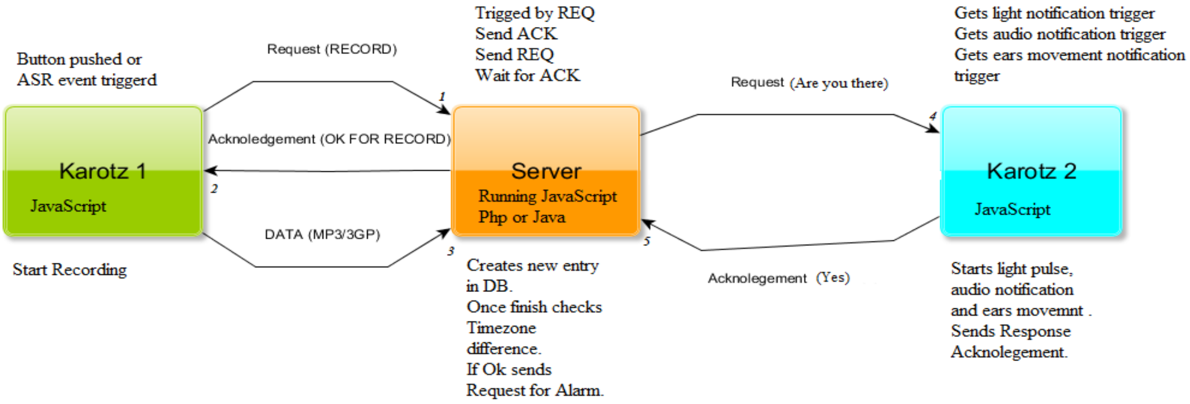

Fig. 2 JavaScript triggering the recording diagram

The first Karotz gets a record command and sends a request to the server. The server will get the request and tell the first Karotz that it is ready to get the data. The first Karotz starts recording and sends the data to the server, while the data is being received by the server, it sends a request to see if the second Karotz is there and waits for a response.

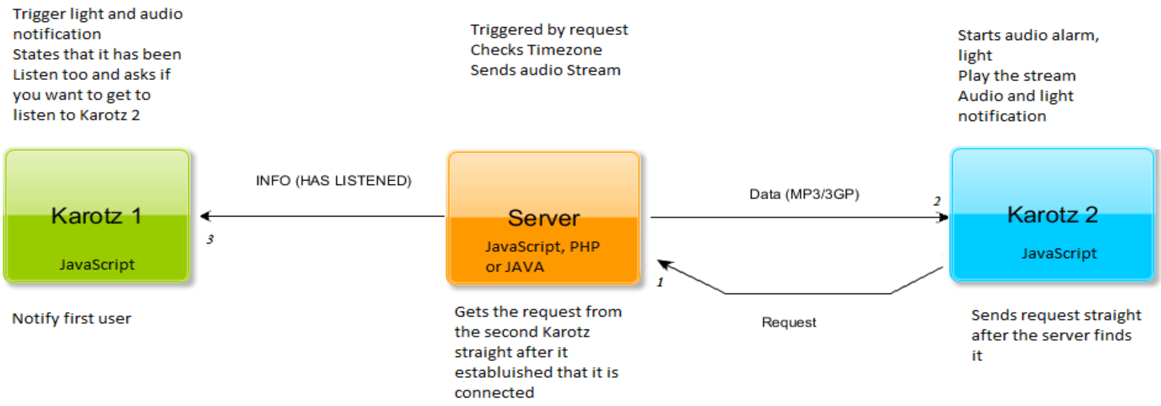

Fig. 3 JavaScript getting the data across the network diagram

The Second Karotz knows that the server has something for it so it requests it to be played. The server sends the data and the second Karotz plays it. While this is happening it also sends information to the first user to let him know that the second Karotz is listening to the data. There is a lot of information going back and forth to make sure that everything runs smoothly and there is no problem with the connection. The information is going to be a simple massage so it will not use a lot of bandwidth and it will also arrive really fast even though it gets sent over the internet.

API REST (on the Violet Server)

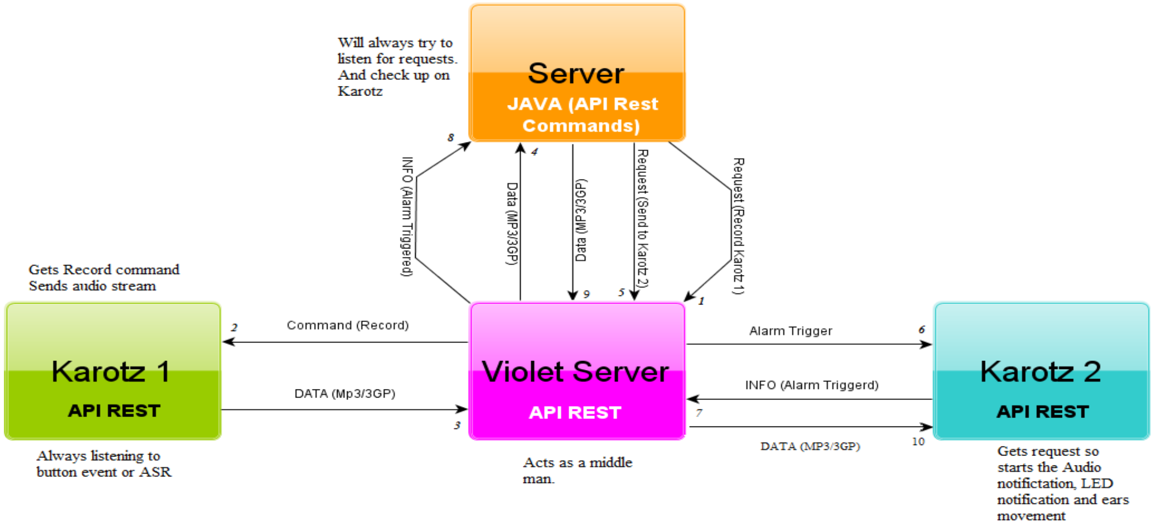

Fig. 4 API REST method communication diagram

The server has to constantly check if the first Karotz is connected and ask it to get the data from the first Karotz. This command gets sent to the Violet server and then gets transmitted to the first Karotz. After that the data gets sent to the Violet server and then the Java one. Once the data gets stored in the Java server, it sends a check to the violet server to see if the second Karotz is available. The violet server receives the information and sends it forward to the second Karotz. The second Karotz sends an Ok to the Violet server that sends it to the java server. Once this is done the java server will send the data to the Violet Server and then send it to the second Karotz.

4. Developing and designing the application

We are first going to represent interactions that are happening with Karotz in each of the scenarios that we have presented in the Introduction (1.1):

Domestic connection. For this type of connection we will have one of the Karotz sending an audio diary to the server. The server will store and process it and will than send it forward to the second Karotz. To initiate recording process the top button will simply need to be pressed, Karotz will use its internal microphone to record audio and using Wi-Fi it will send the data to a java or PHP server. The other Karotz will receive this audio stream and play it back automatically. This stream can go either way so anybody can record at any time.

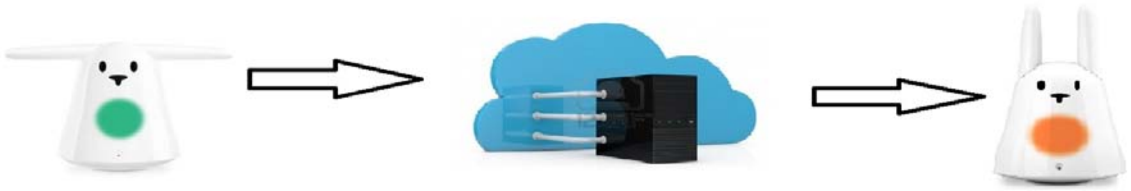

Fig. 5 The data gets sent to the server and then it gets sent to other Karotz

Audio diaries can go back and forth through the server enabling both user to listen to each other’s audio diaries.

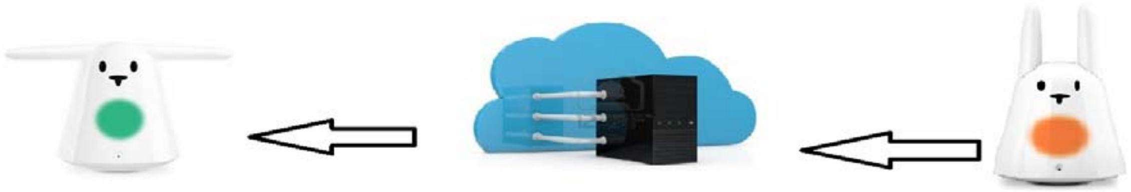

Fig. 6 The other Karotz responds with a call to the server

International connection. Because the people using this will be located in different countries maybe even different continents, there has to be a way to get the audio to play, taking in account the time delay. So let us say one person asks Karotz to record. Karotz will send the data to the server and the server will start to process it but notices that the second Karotz is in different time zone, so it will not play the file because it is too late.

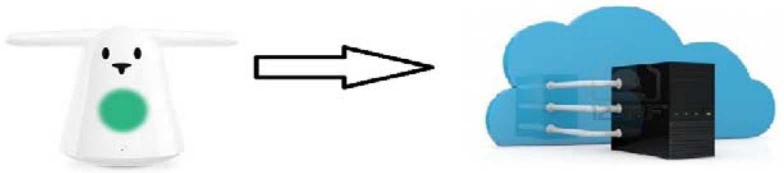

Fig. 7 The data gets sent to the server

The server will store the data and wait for an appropriate time once there is an appropriate time then the server will send the audio stream to the other Karotz. When the Karotz will receive the data it will start playing it automatically.

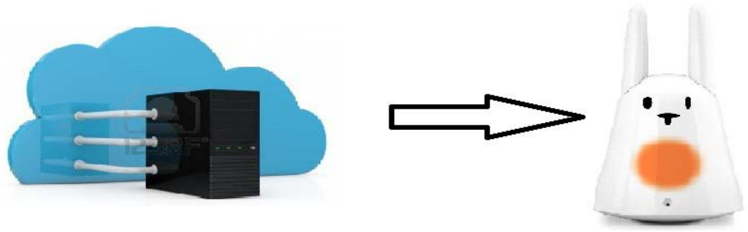

Fig. 8 The server found the Karotz in a different country and waits for the correct time to send the warning message

Local connection with specific interaction. Now this has the same interaction as the first local connection but it allows the user to remote trigger a recording , this will automatically make the second Karotz record and will get it to send data straight to the first Karotz and vice versa. Once the data is received the first Karotz will play it on its loud speaker and indicate to the user that it is playing. If the remote trigger is activated it will let both user know that a remote action is happening giving the possibility to stop it on both sides.

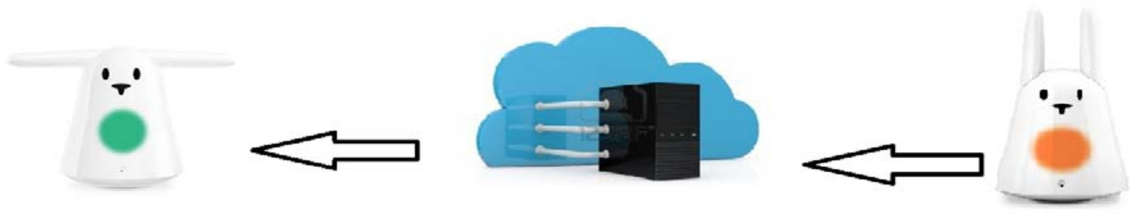

Fig. 9 The data gets sent to the server and the server gets Karotz to play automatically

International connection with specific interaction. This method interacts the same as the International connection but the user may want to record at the same time as the other user. So the server will accept both record commands and will get the audio stream from both Karotz. After the recording it is finished both Karotz will get the appropriate stream to play back. Once each of the Karotz receives the audio stream they will directly play on the loudspeaker.

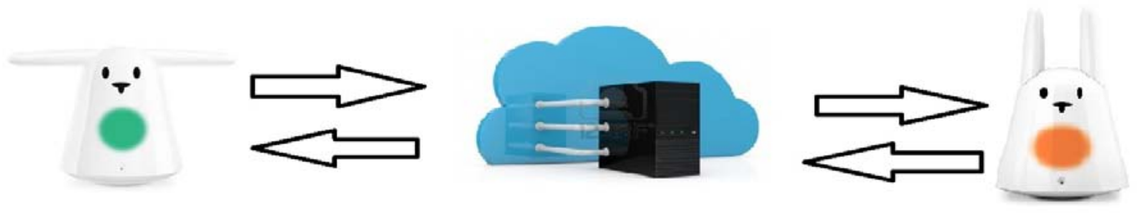

Fig. 10 The Server will let them both record at the same time and playback at the same time

Now that we have planned everything it is time to develop the application. For the approach we have chosen the local JavaScript method in coding the application. This method was chosen because although the API REST method did not seem that hard even if the record function was deprecated. We also needed to take network lag into consideration. API Rest uses a lot more network transmits and receives so it would have severally crippled the system.

My initial attempt in coding the application was creating a java socket and then making a simple JavaScript that uses the record/play function already available in Karotz, after which it connects to the socket to transfer the data. This seemed like a simple procedure but it did not work because the Karotz OS does not allow you to keep a socket connection active. It allows you to connect to a socket but then blocks the connection. So the application that we have attempted has forced my java server to go in an infinite loop. It was trying to connect, it gets connected but then it would go in an infinite loop, so the timeout would kick in and it will try to connect again and open another connection. This happened for about 2 minutes till my ram got full and my whole Linux operating system decided that it had enough.

While doing this initial build and using some of the test applications we have discovered some major problems in the Karotz OS:

- It does have a log file but it gets stored on the device. This would not have been a problem however the drivers that allow you to connect to the pc will not work on anything newer than Windows XP.
- There is no warning if an application crashes. In case of a crash the application gets forced closed and the OS Resumes as normal without any warning.

Because of these major problems we had to made a way to dump the errors from the server directly in Karotz's speech functions (making a simple debugger). We will talk about this later on.

Now that we know of some of the major problems of the OS we have recreated the server using PHP.

4.1 The PHP Server and Database (SERVER SIDE)

The database is made up of two tables, one that holds registered devices and another that holds recorded diary entries. This is the structure of the two database tables:

Table 1 Device id table

device_id
22-15-76
77-63-20

Table 2 Diary entry tables

file_name	recorded_by	played_by
1	22-15-76	77-63-20
2	77-63-20	22-15-76

In order to stop the device from playing back the same audio diary that it has recorded, we had to find a way to distinguish between each device connected to the server. To solve the problem we have decided to generate an id for each device, by making a PHP script that handles registering. The script generates a random id made up of 3 groups of numbers separated by hi fins e.g. 10-89-16.

Because the id needs to be unique the script checks if the id already exists in the database. Once the unique id has been generated the script will add it to the database table that contains registered devices. Finally the script will echo the id to the page. One major problem in the Karotz operating system is that it does not have a way to change the server address. It requires a hard written IP address in each compiled version of an application. To overcome this problem we first created a PHP page that contains 3 fields username, password and server address and a submit button. By accessing this page and filling in all the fields it will enable you to change the server address. The server address field contains the root location of all the PHP scripts and recordings that make the application function.

The script checks if it receives the form data and if it does tests if the password and username are correct. If they are not correct it will output an error message. If they are correct it updates the server address text file. The address of this PHP script and server address file is in a static location so it can always be accessed by Karotz.

4.1.1 Server side record script using PHP

This PHP script needs the recording device id posted to it as a URL variable ($_GET). The script then checks in the database if the device is registered, if it isn't it will kill the script giving an error message. The next step is to insert the id of the device into the diary entry table in recorded_by column. Because the filename is automatically generated in the table, the script then gets the last file_name from the table where the recorded by id matches the device_id. After this is done the script then moves the received .flac file into the designated location and renames it to the appropriate file name returned from the database. If there was an error receiving the .flac file the script gets killed and an error message is displayed.

4.1.2 Server side play script using PHP

This PHP script needs the recording device id posted as a URL variable. After that it returns the entire table values were the recorded by does not equal the device_id from the URL variable. Once this is done the script splits (explode command) each row in the played_by column at the character ",". Next the script loops through each splitted result and checks if the device_id is found. If the device_id is not present in the current row the script will then update the row to include the device_id and echo the play command (play:filename). If the device_id is found in every row then the script echoes none to indicate that there are no files to play.

4.2 The Karotz side application (Client side)

After doing all of this to the server and doing some simple tests we have moved on coding the client side application on Karotz so we can use the PHP scripts. The client side application will be composed of three features register, record and play. Because we need to assign the server address before any action can be performed, the application will test if the server address text file located in the application root directory has a value in it. If it does not it accessed the server address text file, returns the address from this file and writes it to the local address server file. If the file is written the application will get the value written in the file and store it in a variable. So each time Karotz accesses a PHP script page the URL is made up of, the server address and the script name. The application checks if id.txt is found in the device. It then writes the id of the device to the PHP page. If the file is empty the register function is called. The register function gives an audio warning to the user that the device is being registered and then using the Karotz get function which accesses the register.php page returning the result of the page and storing it in variable. If the return result is error the user is notified by audible message and the LED is changed to red. If no errors occur it then saves the id to the id text file.

4.2.1 The client side record feature using JavaScript

The record function in the application calls the Karotz API record command and records 1.5 minute block of sound. Once the recording is done the application will submit the recorded file along with its id to the record.php page. Because Karotz does not allow you to record more than 1.5 minute blocks we had to build a loop that will keep on calling the record function once the recording is finished. So this will make Karotz continuously record until the stop function is triggered by the user by pressing Karotz top button.

After testing this feature we have come across a major problem and that is although Karotz should just send the audio recording and also saves it to memory .After exactly 25 minutes of recording we had to make way for Karotz to free up the memory. If we would not have done this the application would simply crash after each 25 minutes block. To overcome this problem we have made a text file that contains a Boolean variable, to let the application know that it was recording. Then at about 20 minutes we have made the application close itself and reopen in a minute after closing. This is the only way we can assure that Karotz clears its memory and can record constantly for a long time.

4.2.2 The client side play feature using JavaScript

Most of its time the application checks if a file is available to play. In order to do this we have set up a timer that checks every 30 seconds if a file is available to play. The play function accesses the play.php page and passes the id of the device in a URL variable returning the result of the page. If the result is an error message the user will be warned by audible message and the Karotz Led will light up red. If the return result is "play:filename" syntax , the device will then get the name of the file it needs to play from the value that has been return from the PHP page. It will then change the LED colour to blue and will call the built in multimedia play function to play the file.

Once the file has done playing it will instantly recall the play function to see if another file is available to play. We have used this method to make sure that the audio track is smooth and is not interrupted. If the play function returns none from the PHP page the led changes colour to pink to show the user that there is no file available to play. It then returns to the normal application timer that checks every 30 seconds if there is a file to play.

4.2.3 Changes and improvements in the application

We have decided to implement a new remote record feature. My next problem was how to get Karotz to perform a remote action during each operation. We needed to find a way to test what the record status was. In order to do this we have created a new PHP scripts called remote record which received the device ID as a URL variable and then returns the record status from the database for that specific device. This script will then output the status. In order to check the record status the device will access this page and submit its ID as a URL variable and then test if the record status is false or true.

The problem now is that we have to find a way to stop the application from recording or playing or to make it start recording based on this returned value. My approach to solve this problem is to include the check record status after each action performed. For example once the recording has stopped, before actually starting to record again we call the check record function which tests if the record status variable is true, if it is we carry on with the recording procedure otherwise we stop the recording and simply go back to the default check play timer functions.

The problem now is how to start the recording if it is stopped. To do this, after each check play function is executed we call the check record status function which gets what the device record status is. If the device record status is true, instead of running the play function again we simply run the record function. If it is false everything simply will carry on as normal.

We had to also create our own error checking and dumping feature since the Karotz OS no longer is compatible with current windows drivers. We am making the server dump any error it can pick up sending it to the device. The errors are always accompanied by the LED on Karotz turning red.

4.2.4 Button interaction on the client side application

For a single press action the application will toggle between recordings. Once the button is pressed the device checks if the recording Boolean is true, if it is it will stop the recording, if it is not it will initiate the recording by calling the record function.

For a double press action the application will exit, in other words if a double press of the button is detected the device will tell the user that the application is being closed and then it will run the exit command which closes the application.

To perform remote action the button has to be held in i.e. a long press. Once a long press is detected the user will hear an audio message telling them that a remote action is being performed. After this the application will access the change remote action PHP script and passes the device ID as a URL variable. The remote action PHP script checks if the device ID is set and if it is registered in the database. If the device is not registered an error message will be displayed and the script will be killed.

To hold the remote record value we have altered the Karotz device id table. Now it also contains a new column with an enum (enumeration) value of either false or true. This value represents if the device is currently recording. The next step of the remote action PHP script is to return all of the record statuses of the devices which do not have the same ID as the ID received from the URL variable. The next thing that the application does is toggles between statuses, in other words if the enum is false it will update that database row and make it true or vice-versa. So now the database Device id tables has the following structure:

Table 3 New device id table

device_id	is_recording
22-15-76	true
77-63-20	false

Once the user presses the record button before executing the record command a function is called which automatically updates the device recording status. This function opens the update record status page and posts the devices ID and also the record status as URL variable (if the application is not recording the status will be false, if it is the status will be true). By doing this we am maintaining an accurate recording status in the database allowing for far more superior interaction with each device.

The update status script first checks if the received ID is valid and if the device is registered. If the device is registered it checks if the status is either true or false. If everything is okay the script will then update the database row where the device ID is equal to the device received from the URL variable so that it includes the new status. If any errors occur or the device is not registered or the status is invalid an error message will be displayed and the script will be killed.

4.3 Changes from my initial plan

There had to be some changes done to the initial plan because they are impossible on the device. So the actual Karotz OS has some huge limitations. One of the major things that are not possible is the ability to directly connect two Karotz together without using a server. This was discovered very fast when developing because Karotz can only communicate using URL posts and gets. It can stream but it cannot keep a connection.

One of the main changed that we had to make is to stop the user from exiting the application. This had to be done because of the limitation on the way it records and stores to memory making the device run out of memory after 20 minutes of recording. We had to make the application close and reopen itself, it being the only way to clear out the temporary memory in Karotz.

5. Evaluation

After finishing up the programming and doing some simple tests to make sure everything works as it should. We have decided to do some tests on people to see if the application that we have developed is actually working as nice as we have planned. The things that we will be looking for is how the participants are going to interact with the device. We will be looking to see if the server can handle the stress of constant audio recordings. We will also like to see if there are any different interactions that the users can suggest after they have used Karotz and of course we would also like to see if Karotz did actually help the users feel closer to each other.

5.1 Ethics approval preparation

We prepared a participant information sheet to explain what the purpose is to the participant and the consent agreement. After doing this we had to make up the outline of the whole project to present it to the ethics committee so they can analyse it and give me back an answer if we am allowed to do the tests or not.

The next thing that we had to do is create a questionnaire so we can get some feedback from the participants. This way we am able to fix up the application if needed and also get really useful feedback. The last thing that we needed to attach is a task sheet with some diary entries to help me monitor how many times the participant has used the application and how did they interact with Karotz.

5.2 The user study

Once we have received my ethics approval. We have setup the test and gave the participants some simple instructions on how to use Karotz. We say simple instructions because we have made Karotz really user friendly. To record or stop recording one must only press the top button once and to remote record or remote stop a long press is required. We have also made up light themes for playing the led is blue, for recording it is orange and for standby it is pink. we have also included my email and phone number to make sure that we can intervene in case there is a major problem with the system, so the participants will not get annoyed using it.

Because there is a limited number of Karotz devices and a limited time frame we have decided to only use 4 participants in the study. To try and vary the study as much as possible, we am using participants with quite a different age variation and of course very different sets of skill. By doing this we am slightly improving the accuracy of such a small set to work with. We have also set the timeframe to be 1-4 days this way we can start analysing the data as soon as possible without having to rush through it. So after the 1-4 days the users should have had the diary entries filled in. Now we will go and ask them to fill in the questionnaire and also ask for some verbal feedback to see how did they feel during the test and if they have any major problems.

5.2.1 Results from the user study

Now that the user study is over we will be presenting my results and we will be talking about how it went and what feedback we have received from the participants. We have done two users study done as mentioned in the User Study section of this report (5.2).

First user study

The first user study was done at the participant's private home. One of the participants was a student and the other was lab technician.

The test started on Wednesday and it finished on Friday, so it only lasted 3 days. The interesting part of the study on these participants is that they communicating during the evening towards night-time.

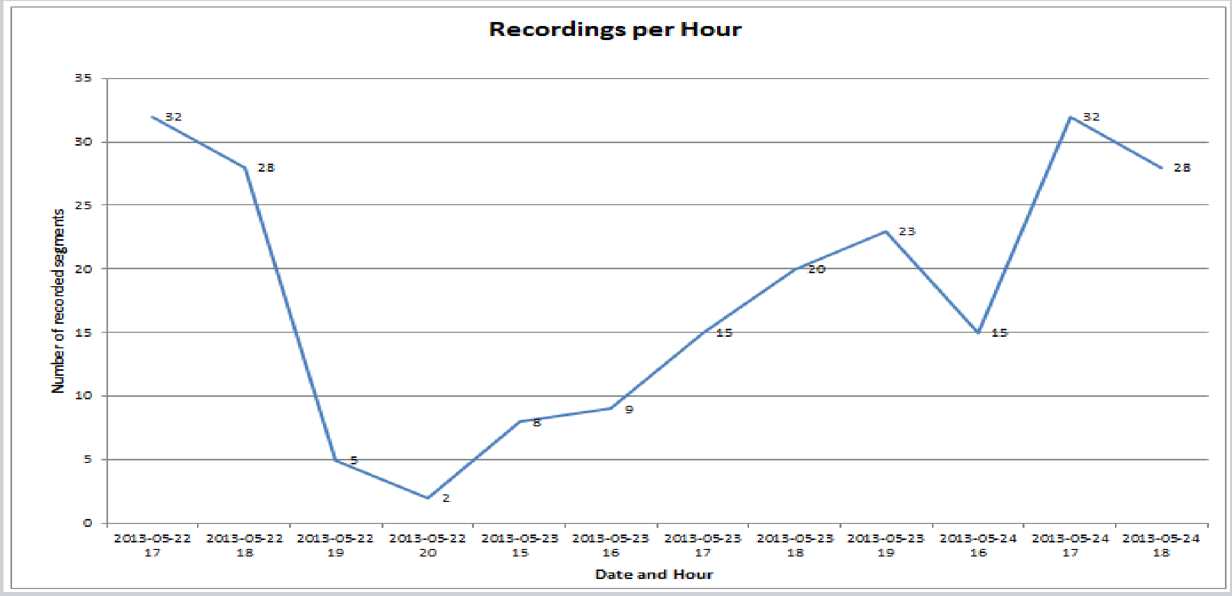

Fig. 11 Graph of first user study recording chunks

Because we have no access on the actual audio recording, we have made a logger that will give me all of the chunk amounts, the time and date when they were generated. The vertical column shows is the chunk amount and on the horizontal there is the time and date. So it is chunks vs time date, this way we can show the amount of recordings at a certain time. The beginning of the user study was really interesting because the 2 participants actually stared using the device quite a bit having 32 chunks , this being the whole time slot (1 hour and 4 minutes). It shows that the participants probably were curious, it being something new also the graph went slightly down because the time was getting late. Next the behaviour is quite interesting because it seems that 6 to 7 is a peek time for the participants. One thing to note the participants first felt somewhat surprised and pleased. The most used feature from these participants was the remote record, and it was stated that although at first it feels like a weird thing to do once you know that it is a private location, it made them feel better.

From the Diary entries it is shown that the participants in this first study uses Skype and phone calls for most of their conversations, However they are not often, their communication pattern being only once or twice a week. Karotz in this study has proven to be quite a useful tool, they have had some interesting experience. Although the user study only went for 3 days they both stated that the Karotz was a really useful tool. Both loved the fact that it was really easy to use, the application only having five commands.

The thing they have found difficult was not being able to tell when the person has the device on, because the application was just either playing or recording if it was on. When asked if they felt a tendency to use Karotz after using it both said yes, this shows that even if they did not have Karotz for a long time, it does have a tendency to help with them feeling closer to each other. It also helped them imagine they were there near the person because Karotz only recreates sound giving the participants the ability to use imagination. The first day it made the participants feel slightly more homesickness but to my surprise by the third day they felt closer.

Second user study

The second user study was done for 4 days, it started on Tuesday and finished on Friday and we have also got a usage graph for this one like for the first one.

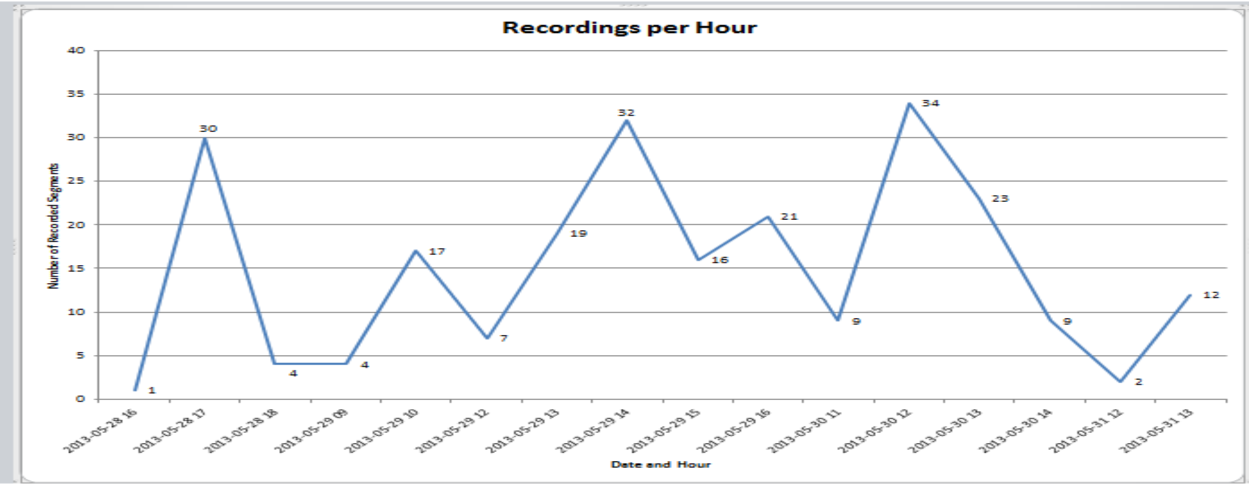

Fig. 12 Graph of second user study recording chunks

This way we can compare them, we say this because the environment where they were recorded is quite different. This user study was undertaken in a work environment so a private office and an open space office. For this user study it is obvious that the time was morning towards afternoon, having even some slight recordings done in the evening. Another difference between this participants and the first is the amount of communication. From the graph we can see the same pattern as the first getting a huge spike within the first hour of usage because it is something new. The other times there was some spikes were around lunch time. This has to do of course with the fact that both participants were working so they were busy. But as we can see on the graph, the first and second user study came back with a lot of data because the participants did all use Karotz on that short timeframe in a decent amount about at least 5 hours a day.

One emotion that a participant did have was it being unpleased that the second participant did not remote record when using the device for the first time. Another interesting issue was that, the two participants were in their offices. One of the participants was feeling like they were eavesdropping the other participant. The whole different environment had a huge impact on this study we feel. If at least one of the participants would have been at home or in a relaxing environment like the participants in the first user study then the outcome might have been slightly different. The similar feature noted was that they liked the ease of use in the Karotz application.

They had the same problems that the first participants did have. So it was a feature that we never thought of since we had both electronic rabbits in the same room or two different rooms but we could always go check on them. From the diary entries, we did notice that these participants have a lot less time spent in the same environment as Karotz. This has slightly proven to be a problem because my plan for Karotz was it to play seamless audio feedback, so it does not give any warning. We have received negative feedback on things because they sometimes missed half of the information because they were absent. Also we got an interesting request from one of the participant he would of like some sort of visual feedback, maybe even to be able to connect to see video using the PC.

The other participant would have like some sort of icon or maybe even to be able to connect using their mobile phone. By having this it will help when they are not in the office or they are away from the place where Karotz is in. It has also been suggested, to have a way to move Karotz from one room to the other. This limitation can be easily overcome by buying a battery pack for Karotz unfortunately we did not have any batteries packs at the time.

The participants in this user study did state that Karotz did help them get closer and that they did use it as a tool along phone calls and Skype. They have both noted on how this is an interesting new way of communicating. It is not like a phone call or Skype because Karotz can blend in and because it plays back audio capturing the environment it helps one imagine the location. The most predominant feedback that we got is one of the participants was listening and the other one is not there, but there still was that feeling that he was there. The results from the users study and diary entries can be viewed from the University of Waikato in the FCMS data archive and will be kept till the end of 2018. All of the data that we received is positive so we will say that the application does do its purpose and it does it really well.

6. Conclusion

For this project we started by researching related works. We needed to do a lot of tests because Karotz is poorly documented. Even the example code that is supplied by the developer has a lot of mistakes in the code. We built an application that will do the purpose of audio recording diaries. Once that was done and we have done a small amount of user tests to get results and see if the device does help people be closer together and help them overcome feeling homesick. Although this project needs a much bigger timeframe to fully analyse the results are really promising.

All of the feedback that we have receive points to the fact that the application, although it might need some improvement like a warning system (having an audible tone that is noticeable and ears to move), was a success.

This has been proven in the results that we have received from my user studies. To my surprise it was obvious even in a short time of usage. Karotz has helped participants feel slightly closer and because it allows you to imagine yourself in the place you hear sounds from, we say that it does help with overcoming the feeling of being homesick.
An interesting thing to note is that the participants really had no problem in leaving Karotz on for hours at a time and once they have got use to using it everybody stated that it was a nice experience.
It was also mentioned in a conversation that we had with the participants that all of them can see the application as being a really useful security device or baby monitor.
The most used feature was the remote record or remote stop feature. Another thing that we believe that made this application a success is the fact that you can leave it on and forget about it as stated in my introduction.

We suggest to continue this project into a more in depth analysis of user behaviour. This is why we have also made a list of suggestions that will help improve the application (6.1.1).
We achieved all goals set out in the initial part of this project. We found out that a person needs more of a warning system than just the LED. But the LED was a feature that everybody enjoyed because it gave accurate information. We have discovered that the simulated presence of a person is really helpful and does help users feel much better after using Karotz.

We discovered that the direct approach making Karotz play data directly is really efficient. We have implemented a scheduler but could never use it because the test was done in the country. The press a button trigger that we have created for Karotz has seemed to be efficient and easy to use. And the remote recording has proven to be a great automated feature because it enables users to start the Karotz that was far away.

6.1 Future work suggestions

There are a few bugs that the participants noticed and these are the suggestions on how to fix these problems. We will start with creating a better warning system that will make it more obvious if a user is present or if a Karotz is turned on. Allow the Karotz to have an audible alarm that will draw the attention of the user, maybe even make the ears move. The audible alarm is an easy fix because there used to be a debug system implemented.

The next thing will be to make a better interaction with Karotz, instead of having it just play and listen have it to listen to user commands and playback data from a specific time and date. Implement using the voice recognition a way for Karotz to listen to a command and use the database to track down a certain audio diary entry. This is an implementation that is easily done because the data is already split up in chunks and is loaded in a database.

Another thing will be to allow Karotz to send data to different devices maybe mobile phone/computers. Use the server PHP script to send audio logs to a mobile phone or computer. It will be also interesting if you can use the PHP script to make Karotz remotely record diary entries. This is possible because the PHP script can accept commands from different devices.

You can also do audio analysis to get rid of not important empty room recordings. Try to do audio analysis to get rid of static sound and only bring out the important sounds. Maybe even develop a way for automatically generating titles to the recordings.

References

1. Baharin , Hanif and Mühlberger , Ralf (2009) “Living with The sound of the Past: Experiencing Sonic Atomic Interaction Using the Sound Diary” The University of Queensland School of Information Technology and Electrical Engineering QLD 4072 Australia
2. Baharin , Hanif and Mühlberger , Ralf (2009) “Utilising the Open Channel Created by Telecare: Acoustic Communication Analysis of Radio Sounds at Home” The University of Queensland School of Information Technology & Electrical Engineering, QLD Australia 4067
3. Kaminsky, Michael , Dourish, Paul , Edwards, W. Keith , LaMarca , Anthony , Salisbury Michael and Smith, Ian (1999) “SWEETPEA: Software Tools for Programmable Embodied Agents” Laboratory for Computer Science Massachusetts Institute of Technology 545 Technology Square Cambridge MA 02139
4. Kysoh society, Accueil (2009) “Tux Droid” Online webpage, Available at http://wiki.tuxdroid-community.org/tdc/index.php?title=Accueil (4 Jun. 2013)
5. Avocado Software Inc “Avocado” Android/IOS application, Available at https://itunes.apple.com/nz/app/avocado/id510134530?mt=8 (4 Jun. 2013)
6. Kaye , Joseph ‘Jofish’ , Levitt , Mariah K. , Nevis , Jeffrey , Golden , Jessica and Schmidt , Vanessa (2005) “Communicating Intimacy One Bit at a Time” Cornell University
   301 College Ave, Ithaca NY 14850
7. Lottridge , Danielle , Masson , Nicolas and Mackay , Wendy (2009) “Sharing Empty Moments: Design for Remote Couples” Interactive Media Lab, Mechanical & Industrial Engineering, University of Toronto
8. Violet “ Karotz” the electronic device used (2011) Manufactures website, Available at http://store.karotz.com/en_US/ (4.Jun.2013)