IMusic: a Bluetooth music application

Angel Aguilera Manglano, Borja Sierra Espín

***Abstract* — nowadays, we can see a huge amount of customized services in lots of applications. A more customized experience get a real full satisfaction to the customers.**

The project was born with the aim of providing this kind of service. Thus a client takes part in the decision- making of the local songs, creating a playlist with songs ordered as the customer's data preferences.

The way is easy: a customer who enters the premises, install the software that local provides and then he has to indicate their customer likes and get it transmitted via Bluetooth. This information is received by the customer attention server and it generates a playlist with personalized songs.

Keywords— AI system, Bluetooth connection, customized music application, mobile interface.

XIII. Introduction

Music is found in every known culture since ancient times. The "oldest known song" was dated 4,000 years ago. The music has been varying wildly between times and places and it can be divided into genres in many different ways. The goal of the application we want to develop is to serve customized music to places such as pubs, restaurants, hotels to help these places to offer a better service for their customers. To make this possible we are going to use: technologies as Bluetooth to set up the connection between customers and the music server, genie and smile library to pair each customer with the musics genre he fits better, and a .net application to receive all the information for the creation of the playlist

XIV. First Step: How to receive the customer's likes?

The first problem we face is how to get the application closer to the customers. All of us are living in the communication era so there are lots of ways to make the connection between the application and clients:

The first idea was to set up a point of connection at the local -a few computers linked to the main server for example- so clients go and fill in a simple form with their musical likes.

Another possibility was to allow a WI-FI access point which makes possible links of customers with their laptops.

Once connected, they could be redirected at the local main page, and all they have to do is fill in the form provided.

But those ideas are quite hardworking or uncomfortable to the clients, in one hand have to stay in line –on the case there is no free computers on the local- to get the likes transmitted in the other hand with the second idea they have to carry with the computer on and it's a nonsense to go party with all of these gadgets.

So if we want the application was successful we need to find the technology that fits those problems, and we found it on Bluetooth technology.

The most of the cell phones nowadays works with Bluetooth, and if we can use this mobile devices, to get an access to the local main server we could solve the connection problem.

Due to the fact that Bluetooth technology is a standardized wireless technology and it is designed to be compatible across a range of very different operating systems and devices, makes us to choice this alternative.

XV. Second Step: How to do an application from customers?

A. Initial decisions

We want a customer's application that allows sending their musical likes and getting it received by the customers dedicated server.

At first we choose to create a .NET technology server for attending all the Bluetooth connections from the customers. All Bluetooth libraries that we found on .Net are under subscription, and those libraries such "IntheHand.Net.Personal" free of charge doesn't work properly because their implementation and options are very limited and do not provide many possible cases of communication in Bluetooth.

Because of that we decided to develop a server with Java technology. The functionality of this server is basically listen Bluetooth connections and to store the content of these connections (saving them into a database).

This java library is in the API JSR-82 and defines two packages that depend on the CLDC javax.microedition.io package:

- javax.bluetooth: core Bluetooth API
- javax.obex: APIs for the Object Exchange (OBEX) protocol

These two packages are totally independent:

- The first one defines classes and basic interfaces for the discovery of devices, service discovery, connection and communication. The communication through javax.bluetooth is at low level: through data streams or by providing arrays of bytes.
- The second one handles the high-level OBEX protocol, this protocol is very similar to HTTP and is mainly used for exchanging files. The OBEX protocol is a standard developed by IrDA and is also used on other wireless technologies than Bluetooth.

The main development platform for the API JSR-82 is J2ME. The API is designed to rely on the CLDC configuration,

however there are implementations to use this API in J2SE. Regarding the client side we did not have any problem at all as most of the mobile work perfectly with MIDlets.

"A MIDlet is a Java program for embedded devices (dedicated to a single activity), more specifically for the Java Virtual Machine MicroEdition (JavaME). Generally MIDlets are games and applications that runs on a mobile phone.)."

But regarding the customers dedicated server side we found the following problems:

- In the detection phase, our own Bluetooth device which we will send and receive it doesn't be detected by the javax.bluetooth library. After several days of searching we discovered the problem, it was because our Bluetooth device is using a Bluetooth stack protocol that is not included in the library we are working with.
  The USB Bluetooth device that we had was running with the Microsoft protocol stack and it wasn't compatible with the library we used. Thus we had to remove the J2ME library and replace it with "bluecove-2.0.2.jar" which accepted the Microsoft protocol stack. This library contains all the protocols the other library has and many others protocols, so the coding was the same but this time it works.
- In the discovery phase we don't find any problems because both server and client java use the same Bluetooth technology for communication so the bluetooth profiles are the same, if there is no correlation between these profiles could not be exchanging information with another device. It is worth to comment that we had to alternatives for the communication due to the API javax.bluetooth makes possible two different ways of connection: SPP and L2CAP:
  - With SPP we get an InputStream and an OutputStream.
  - with L2CAP we get an array of object that we could send or receive

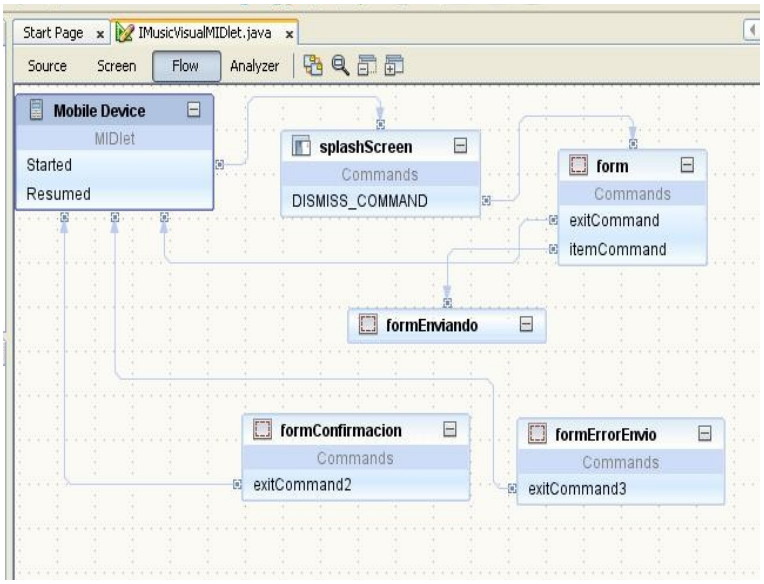
  We choose SPP.

Once these 2 problems were solved, everything related to Bluetooth was swimmingly.

B.Developing the customer application

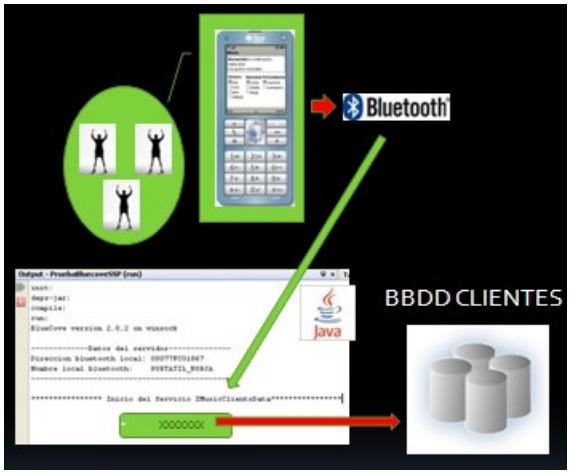
Programming on J2ME, and using the Bluetooth API (JSR 82), which provides a standardized API for using Bluetooth networking, we made the customer application possible.
After the implementation the result was that we have a functionally interface which allows clients to connect their own phones with a java application that performs the task of storing the client's form.

The basic navigation structure of the mobile application is:



It is just a simple form with 3 lists of radio Buttons on which the customer has to select his preferences of songs
"Genero": Genre
"Duracion": Length (short, medium, long)
"Procedencia": Origin (national, foreign)
Once selected by pushing the send button the form is sent to the customers dedicated server, and it's stored properly.
The interface looks as follows:

And finally the structure we want to build is:



Third Step: Build the Bayesian Network

At the point of view of the local, in order to set the best playlist, the application needs to access the customer's data forms to helps the application choose the songs that fits the best. All we have to do is perform an application that arranges the songs on the right position of the playlist. This is possible with the help of Genie (Artificial Intelligence) application that compares all songs on the database with the statistics music likes of all the customers at the local. The result is a preference ordered playlist which begins to play.

C.Genie Application

The GeNIe (Graphical Network Interface) software package can be used to create decision theoretic models intuitively using the graphical click-and-drop interface. GeNIe is the graphical interface to SMILE, a fully portable Bayesian inference engine developed by the Decision Systems Laboratory and thoroughly tested in the field since 1998. GeNIe 2.0 is the latest version of GeNIe. GeNIe 1.0, released to the community in 1998, has received a wide acceptance within both academia and industry. Users of the programs have shared with us their experiences and their suggestions have led to the development of GeNIe 2.0. GeNIe 2.0 has a refreshingly new modern interface, and is even more intuitive and easier to use than GeNIe 1.0. In addition to aesthetics, GeNIe 2.0 has many more new features to offer.

Primary Features:

- Graphical editor to create and modify network models
- Uses the SMILE Engine. You may develop models in GeNIe and create a custom interface for them using SMILE
- Supports chance nodes with General, Noisy OR/MAX and Noisy AND distribution
- Open multiple networks and cut and paste sections of models between them
- Complete integration with MS. Excel, cut and paste data into internal spreadsheet view of GeNIe
- Cross compatibility with other software. Supports all major file types (e.g. Hugin, Netica, Ergo)
- Support for handling observation costs of nodes
- Support for diagnostic case management

D.SMILE LIBRARY

SMILE (Structural Modeling, Inference, and Learning Engine) is a fully platform independent library of C++ classes implementing graphical probabilistic and decision-theoretic models, such as Bayesian networks, influence diagrams, and structural equation models. Their individual class, defined in SMILE API (Application Programming Interface), allows you create, edit, save, and load graphical models, and use them for probabilistic reasoning and decision making under uncertainty. SMILE supports directly object-oriented methodology. SMILE is implemented in C++ in a platform independent fashion. Individual classes of SMILE are accessible from C++ or (as functions) from C programming languages. As most implementations of programming languages define a C interface, this makes SMILE accessible from practically any language on any system. SMILE can be embedded in programs that use graphical probabilistic models as their reasoning engines. Models developed in GeNIe can be equipped with a user interface which utilizes SMILE as the backend engine. SMILE is released as a dynamic link library (DLL). There are also several SMILE wrappers, such as SMILE.NET (.NET interface), SMILEX (Active X), jSMILE (Java interface), etc.

Primary Features:

- Graphical editor to create and modify network models.
- Platform independent, versions available for Windows, Unix (Solaris), Linux, Mac, Pocket PC, etc.
- SMILE.NET available for use with .NET framework. Compatible with all .NET languages, including C# and VB.NET. May be used to create web-based applications of Bayesian networks.
- Thorough and complete documentation.
- Robust and running successfully in the field since 1997.
- Responsive development team support, we will compile SMILE for your platform on demand.

E.Let's build it!

We've got 12 nodes on which we make the comparison between the received statistics likes of customer and the current songs stored on the database of the local

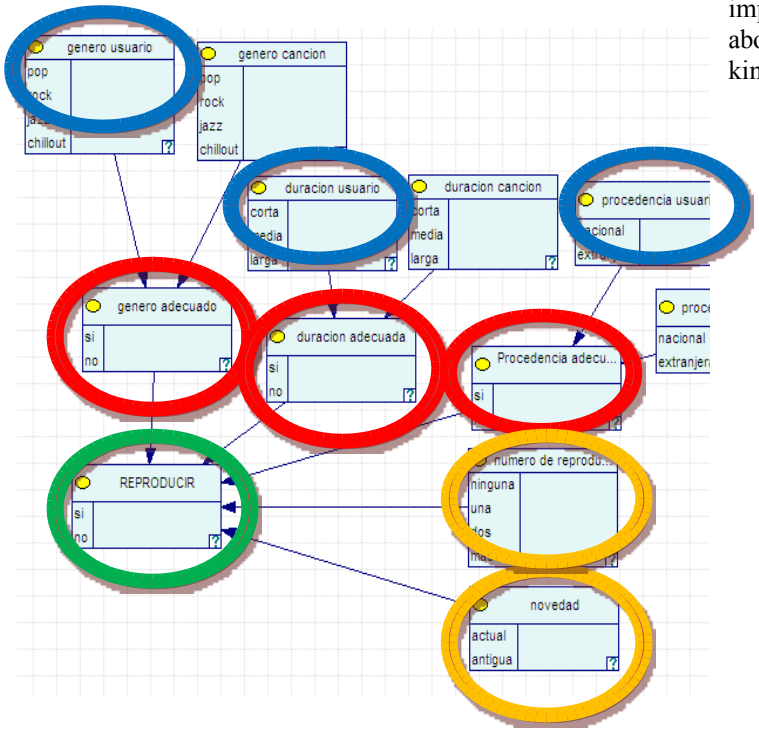
We are going to explain the meaning of this tree structure:

- The nodes with no circle surrounded means the attributes of the songs, such attributes as genre ("genero de cancion"), length("duracion de cancion") and origin ("procedencia de cancion"). This information is set on the nodes by "setEvidence" method, a method of the library that SMILE provides us. All the songs of the database have to get this attributes.
- The nodes surrounded with blue circles mean the statistic information of all the customers who send the form with the Bluetooth customer application.

The joint of these nodes ends on nodes surrounded with red circles called: appropriate genre ("genero apropiado"),

appropriate length ("duracion apropiada") and appropriate origin ("procedencia apropiada") which includes the value of the result of evaluate the previous nodes.

- The nodes surrounded with orange circles mean the information about number of reproductions the song has and if the song is recent. The first one penalized the songs with more number of reproductions; the second one generates more value to the song with this attribute.
- All of those nodes ends on the node with the green circle; it has a value for each song on the database taking into consideration all the previous nodes.



Once we get evaluated all the songs finally, it only remains to arrange the songs into a list and began the reproduction of this playlist.

XVI. Forth Step: How to create the comparison?

We are going to explain where we find our conclusion about decision rates selected on the different styles of music, this site is a metadata database of music called:

"http://www.allmusic.com/".

The database consists of:

- Basic Metadata: names, genres, credits, copyright info, product numbers, etc.
- Descriptive Content: styles, tones, moods, themes, nationalities, etc.
- Relational Content: similar artists and albums, influences, etc.
- Editorial Content: biographies, reviews, rankings, etc.

Allmusic also claims to have the world's largest digital archive of music, including approximately six million songs fully digitized.

The Allmusic database is also used by several generations of Windows Media Player and Musicmatch Jukebox to identify and organize music collections. Windows Media Player 11 and the integrated MTV Urge music store have expanded the use of Allmusic data to include related artists, biographies, reviews, playlists and other metadata.

A.Related researches

Since IMUSIC is a customer likes oriented application it's important to know what studies have been taken on this topic about criteria, needs, likes in order to set the decision of what kind of music they want:

- Young Spaniards between 15-35 years old, lives surrounded of technologies since all of them have a mobile phone, about a 95% have personal computer, 84% have digital camera, 73% mp3 player and a 62% have a videogame-consol under a first trends research by Nokia.
  The first conclusion is that differences between men and women are quite few on use of technologies and the big group on use of new technologies is the young people. Nowadays young people want a comprehensive mobile phone with Bluetooth, USB connection, digital camera, mp3 player, external memory. And talking about design the preferred format for mobile is the opening slide, which is preferred by 31 per cent of young people, followed by monoblock with 26 percent.
  The research highlights the importance of music among young people who listen to an average of 23 hours per week and consider it to be more important than the sport, leaving march, politics or religion and that lie behind family and friends.

- Another study on the people musical likes conducted by Renaud Lambiotte and Marcel Ausloos Statistical Physics experts contend that the musical likes of people are more diverse from what we know.
  These physicists have analyzed a website for exchanging music files and have found that people likes a wide variety of music for a mix of several predefined categories. The results could give a new way of classifying musical genres and be useful for analyzing market trends music. The study is based on the Audioscrobbler site, where users can upload their favorite music to a personal library on a central server. The site is designed to encourage their users to discover new types of music through the suggestions that other people do in their personal libraries.
  Using data from January 2005 researchers were able to examine the music heard by a total of 35,916 people together "possess" 617900 musical groups in their personal libraries. At the top of the list were

Radiohead, Nirvana, Coldplay, Metallica and the Beatles. To analyze the list of users and music listeners used methods of the theory of complex networks in which people and music are two kinds of nodes in a network. Physicists began with a network "absolutely connected" in which two people are "connected" if at least share a song from the same artist.

After applying an adjustable filter to the result network which took into account the correlations between entire collections of music rather than among individual musical groups it could see that as increasing the value of the filter structures found disconnected or "branches" formed along the overall structure of the original network, thus revealing trends and patterns in the form of a map of different music genres.

This method of analysis allows study how music and new trends appears and it quantifies the musical signatures of a large number of individuals and visualizes their collective behavior that eventually leads to the emergence of new sociological communities.

- An investigation of Injuve has done a study about musical preferences, which indicates the percentage of young people who say they like "Fairly" or "much" each of the following musical genres:

	TOTAL (%)
Pop, pop-rock	50.1
Dance	40.2
Latina, salsa	36.4
Cantautores	30.7
Electrónica	29.6
Baladistas, canción melódica	29.5
Rumba	21.9
Flamenco	21.2
Rock'n'roll, rock clásico	20.0
Hip-hop, rap	18.0
Rock alternativo, *grunge, indie-rock*	17.6
Indie-pop, power-pop, pop alternativo	16.7
Popular o típica de su región o comunidad autónoma	16.3
Reggae, ska	16.2
Músicas del mundo, étnicas	15.3
Rythm&blues, soul, funk	14.9
Heavy, hard rock, metal	14.9
Clásica	13.7
Rock progresivo, psicodelia	11.7
Punk, hardcore	11.6
New age	10.3
Jazz	9.5
Folk, country, blues	7.3
Total	1.900

NS: p>.05.

Thus, among the styles like "fairly" or "a lot", highlights the pop-rock or pop (likes to half of those interviewed), dance music (40%), Latin music and salsa (36 %), Songwriters (31%), electronic music (30%) and the trivial and melodic song (30%). The rest of styles presented acceptances below (all less than 22%).

The classification is usually done in musical genres along the following criteria:

1. Features melodic, harmonic and rhythmic.
2. Instrumentation
3. Geographic place where it's developed mainly.
4. Historical sources.
5. Structure of works (songs, movements, etc.).
6. Standards and techniques of composition and interpretation.
7. Ways and methods of broadcasting

B.Decisions making

Conclusions on previous researches:

- Referring to the first research we could deduce that it could be better to take the advantages of the current mobile technology for connect our application with the customers likes. Instead of use terminal input devices and all that it entails as the problem of setting up in each local, this solution is cheaper and better. But even so using mobile technology there are many options you can use instead, those options are: WiFi, infrared…

- The second research gives us information on all genres that exist today in order to catalog all the music likes of customers.

  Through this research we find that we may be expanding and improving the Bayesian network that it will help us to decide what music is appropriate and what not as the likes of customers.

  Because there are so many musical genres, in our first version we will introduce only a few genres.

  In the future versions we will expand the application to get a better align with the likes of customers. Then we could establish the musical genre as an essential attribute to decide what kind of music we want.

- With the third research we figure out what is the music likes globally. One of the objectives of our application is that any customer can find a way to express their musical tendency and to make it possible we have to develop an easy and friendly user interface that allows each user whatever likes he has can become an important build-part of the songs decision making of the local.

  Because the interface developed has to be simple we couldn't include so many musical genres, so for a first version we have chosen the following genres:

  - Pop
  - Rock
  - Jazz

- Chillout

After getting information about several researches we found we have reached the following conclusion about their influence on each other genres:

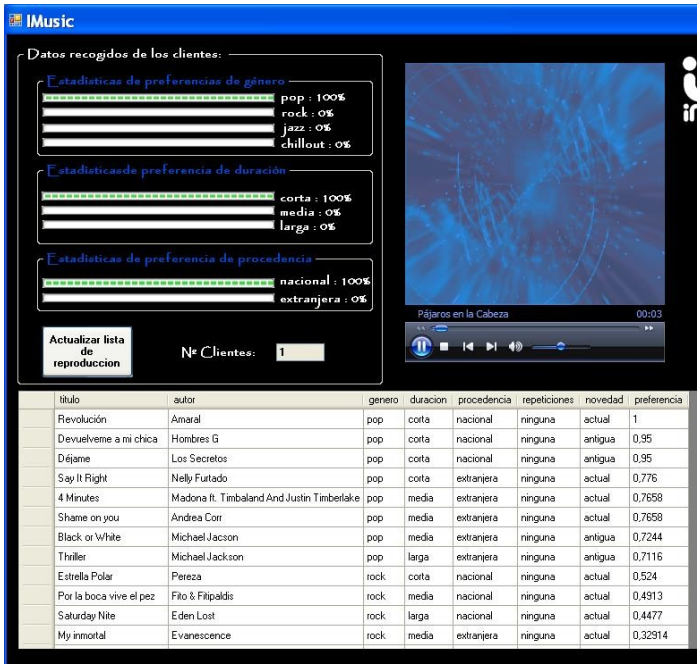
Influence of	over	%
Pop	Rock	30%
	Jazz	10%
	Chillout	5%
Rock	Pop	50%
	Jazz	20%
	Chillout	10%
Jazz	Pop	10%
	Rock	36%
	Chillout	60%
Chillout	Pop	15%
	Rock	20%
	Jazz	50%

We also found information about different styles and trends of music so the next step was quite easy: pair each musical genre at the Genie application with the others genres to make a intelligence decision making based on the origin of this style, trends, music speed (bits per second).

Once make this step, we have to make decision of the results given for the customers as well, this decision consist on the form they had filled in. The fields with the relevant decision making weight was:

- Genre
- Nationality
- Length

This process is done for each song of our Database so at the end we have a "CUSTOMIZED PLAYLIST" ready to use into our local music application designed to play songs:



XVII. Conclusions

In the last few weeks we have spend a lot of time working on development of IMusic prototype, thinking if this project could get ahead. Sometimes we have the feeling that we can't do this, but eventually, the reward of all this hard work is worth it.

Technology talking, we have to learn a lot of packages and libraries to make possible the communication between a simple mobile phone, with a computer, which it comes to mind that there is a lot of applications that could make the daily life better. We even learnt, how to make a free chat using the phones with Bluetooth, and many others applications, that could fit well with our own interest (a little help on test evaluations, for instance). Talking about the IMUSIC application, we found very interesting the idea of being able to send, the kind of music you wants more, and this idea of business can be tempting for those locals that want to experiment a new idea of service for offering the customers. The next step that the application could overtake is to learn about new music tendencies and include them in our application. These procedures probably require a huge change on the application intelligence actions, but it is quite interesting to be developed in the future.

References

[1] Blue Cove [http://sourceforge.net/projects/bluecove/]
is a JSR-82 J2SE implementation that currently interfaces with the Mac OS X, WIDCOMM, BlueSoleil and Microsoft Bluetooth stack found in Windows XP SP2 and newer

[2] javaBluetooth [http://www.javabluetooth.org/]
The JavaBluetooth Stack is a 100% (no native) Java implementation of the Bluetooth Specifications Version 1.1. It currently supports HCI, L2CAP and SDP.

[3] Neofronteras [http://neofronteras.com/?p=199]
Análisis de gustos musicales en Internet

[4] Injuve Instituto de la juventud
[http://www.injuve.mtas.es/injuve/contenidos.item.action?id=314621908&menuId=428194837]
Jóvenes entre sonidos: Hábitos, gustos y referentes musicales (FAD).
Ignacio Megías Quirós, Elena Rodríguez San Julián (2003).

[5] Madrid, 2 de abril de 2008. Instituto de investigación social y de mercado
[http://www.nokia.es/A4213038?newsid=-12015]

[6] [http://generosmusicales.supaw.com/rock.htm]

[7] Sun Mobile Device Technology
[http://developers.sun.com/mobility/getstart/]