CodeBLUE: A Bluetooth Interactive Dance Club System

Dennis Hromin, Michael Chladil, Natalie Vanatta, David Naumann, Susanne Wetzel
Dept of Computer Science,
Stevens Inst of Technology, Hoboken, NJ 07030

*Farooq Anjum, Ravi Jain**

Applied Research, Telcordia Technologies
Morristown, NJ 07960

Abstract. This paper examines the use of Bluetooth for a collaborative music creation system called codeBLUE where the low cost, low power and small dimensions of Bluetooth technology are critical. Dancers using the codeBLUE system wear clothing incorporating Bluetooth-enabled sensors that measure and transmit information about the dancers' movements to a Bluetooth access point positioned in the demonstration area, which in turn forwards the information to a control system. The system software transforms the simple dance movements into musical modifications in real time, altering the melodic, rhythmic, and dynamic properties of the music stream in terms of MIDI parameters. A configuration console allows the DJ to modify the effects that each type of sensor produces, providing him or her yet another channel of creativity and keeping the codeBLUE experience fresh for participants.

The paper describes the architecture, design, and hardware and software implementation of the codeBLUE proof-of-concept prototype. The paper also discusses our evaluation of the technology used for this application. The system has been successfully demonstrated to a live audience.

1 Introduction

Collaborative music creation can be one of the most powerful forms of human communication. Playing in private improvisational sessions or at public concerts exhilarates musicians due to the exchanges of emotional and intellectual information inherent in such creative settings. While many people express themselves musically by whistling and humming, relatively few have enough confidence in their musical abilities to collaborate openly with others.

This paper explores whether emerging Bluetooth technology [Bluespec, Bluebook01] can be used to provide a means for allowing untrained performers to share in the joy of musical collaboration and interaction. The idea is to develop a wireless interactive dance club environment that presents users with a simplified and intuitive musical interface. By transforming simple dance movements into musical modifications, untrained performers can collaborate in ways previously reserved for seasoned professionals. (A glance at the popularity of karaoke and sing-alongs demonstrates the powerful attraction of informal and amateur musical collaborations.) This paper presents the architecture, design, and hardware and software implementation of a prototype Bluetooth wireless dance club system, called codeBLUE. Our objective is to explore the practical viability of using Bluetooth wireless technology to expand and realize the opportunities for interactive musical collaboration. While previous research projects [e.g. Paradiso99] have also considered interactive dance club applications, we believe that Bluetooth technology, with its low cost, low power, and small dimensions, has the potential to render musical collaboration through dance for tens and possibly hundreds of participants a reality.

Example scenario. Ali, Bob and Carol are at *Club codeBLUE*, and have been outfitted with various wireless sensors. The DJ is playing a continuous selection of dance music. A motion sensor attached to Ali's sleeve allows him to modify the volume of the drumbeat on channel 1 when he moves his arm in a certain way; a pressure sensor attached to Bob's shoes allows him to raise or lower the pitch of certain instruments on channel 2 by tap dancing; and a force sensor in Carol's palm allows her to alter the ceiling lights near her by clapping her hands. Like other participants, Ali, Bob and Carol are provided detailed instructions about the effects of each sensor by the doormen or DJ; alternatively they are allowed to discover them completely by experimentation, or something in between. As they learn and interact the quality of their collaboration becomes more interesting. Participants wearing sensors that cause the same effects discover each other over the course of a session and experiment by synchronizing their movements. (A simple analog is the way spectators at a sporting event collaborate to form "the wave".) The DJ can modify the effects that each type of sensor produces, possibly in accordance with the characteristics of the music (disco, rock, new wave, etc.), or over the course of a session, providing him or her yet another channel of creativity and keeping the codeBLUE experience fresh for participants.

The codeBLUE scenario immediately raises several constraints. The weight and form factor of a wearable device has to be such as to make it unobtrusive during dancing or movement. The device has to be rugged enough to withstand use in a dance club. Battery life has to be long enough not only to last a dance session but also to minimize the cost and nuisance of replacement. Finally, the cost of the device itself has to be sufficiently low to warrant its use in a dance club environment. Sensors meeting these requirements are available in the market today. This fact along with the use of Bluetooth technology allows us to meet the requirements. To a large

extent, codeBLUE rests on the promise of light, low-power, inexpensive Bluetooth technology.

Our contributions in this paper are as follows. We investigate a novel application domain for Bluetooth wireless technology, namely human collaboration, and its instantiation for a wireless dance club environment. We also investigate the practical viability of applying Bluetooth for this class of applications. In the following section we describe the system model and architecture. In sections 3 and 4 we describe the hardware and software design respectively. In section 5 we briefly discuss the cost and usability features of codeBLUE, as well as related work. Finally we end with conclusions and suggestions for further work.

The codeBLUE prototype system has been implemented and successfully demonstrated before a live audience in an auditorium with state-of-the-art theatrical lighting and sound equipment to create an interactive musical environment.

2 System Model and Architecture

A block diagram of the system architecture is shown in Figure 1. Battery-operated wireless sensor modules are worn by participants and communicate sensor data (pressure, temperature, etc.) to wireless access points over Bluetooth links. Each sensor module consists of a body sensor, a microcontroller, and a Bluetooth module. An access point has power as well as wired LAN connections and is typically mounted on the dance club ceiling.

The access point relays the sensor data packet over an Ethernet LAN to the codeBLUE Controller computer. The Controller interfaces to a standard Musical Instrument Digital Interface (MIDI) format file player for sound effects.

It also interfaces via MIDI to a standard DMX lighting controller to produce visual lighting effects for stand-alone and intelligent lighting instruments.. The codeBLUE Controller application, called *CBPlayer*, contains a data filter that modifies the MIDI stream in response to sensor data from the access point. It also offers the DJ or system administrator a console and GUI to configure the music playlist as well as the effect that each sensor has, and other program parameters.

MIDI format files are binary data files containing instructions that tell electronic instruments how to play a song – from the notes and their durations to dynamic and pedal markings – much like musical scores. Standard MIDI files were chosen as best suited to the codeBLUE application, even though there are a number of standard audio file types to choose from (.wav, aiff, MP3, .au.) There are several advantages to generating sound with a MIDI synthesizer rather than using sampled audio from a disk or CD-ROM. The first advantage is storage space. Data files used to store digitally sampled audio in PCM format (such as .wav files) tend to be quite large. A four-minute song sampled at 44.1khz in stereo requires about forty megabytes of storage space, or ten megabytes per minute. MIDI files, on the other hand, typically consume ten kilobytes per minute. The second reason codeBLUE utilizes MIDI lies in the interactivity requirement. Since MIDI files are merely instructions about how to play a song, music stored in them can be easily modified in real time. Making these transformations in an audio file are extremely processor-intensive, and, in the case of rhythm, sometimes not even possible.

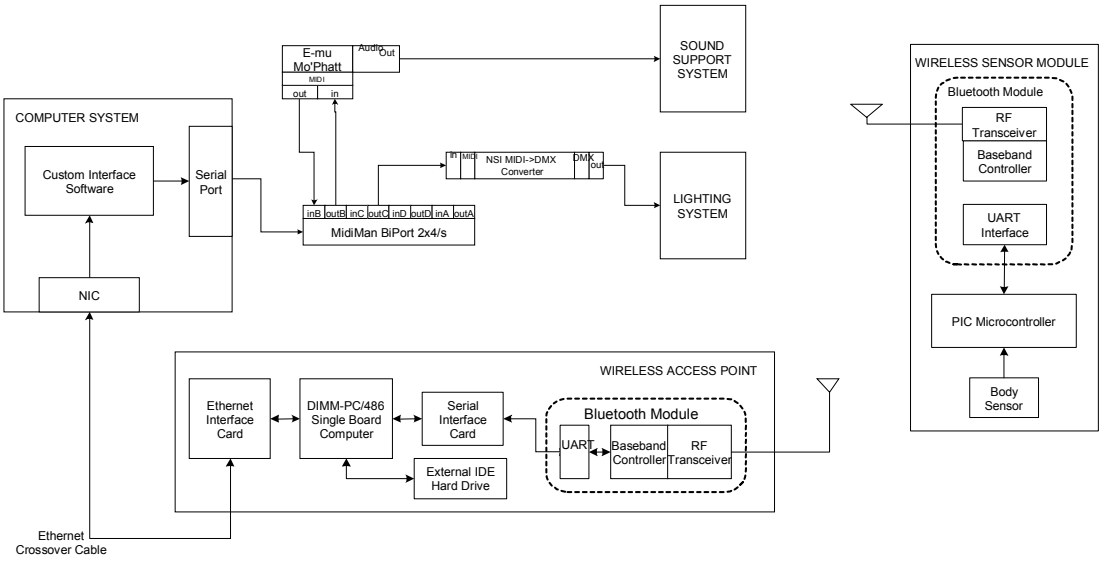

Figure 1: Block diagram of codeBLUE system architecture

3 Hardware design and implementation

Bluetooth is a low power, low cost short range wireless communication system operating in the 2.4GHz

ISM band. It enables small portable devices to connect to each other and communicate in an ad-hoc fashion with nominal speeds of up to 1 Mbps. We omit description of Bluetooth; details can be found in [Bluespec,Bluebook01]. Only Asynchronous Connectionless (ACL) Bluetooth links, designed for data packets, are used in the codeBLUE system.

The hardware consists largely of three components: the wireless access point, the codeBLUE controller interfaced with sound/lighting hardware, and the sensor module. The sensor module operates as a Bluetooth slave, and the wireless access point operates as a master.

For cost and form-factor reasons a wireless access point was custom built for codeBLUE, although in deployment we expect that with increasing commercialization of Bluetooth technology a suitable access point will become available.

The codeBLUE Controller is a Windows PC. The sound/light hardware consists of commercial off-the-shelf equipment such as synthesizers, mixers, etc as would be found in a high-end studio.

The wireless sensor module is a key part of the hardware design, as this is where form factor, weight, size, power and cost considerations are most severe. It contains a microcontroller, a Bluetooth wireless transceiver, and up to five analog sensors.

The sensors used in the codeBLUE prototype operate at 5V and output a variable voltage 0 - 5V. The *ReachClose* sensor measures infrared light reflections to measure a distance to an object and is insensitive to ambient light variations. The *GForce* sensor measures acceleration and deceleration along a single axis. To measure the flex angle of a specific body part such as the wrists, elbows, and knees, a *Bend* sensor is used. The *Flash* sensor is used to detect rapid lights changes or flashes (such as strobe lights at a club) using a phototransistor. The *Light* sensor senses variations in ambient light using a photoresistor. The temperature on an individual's body or the room can be measured using the *Hot* sensor. Finally, the *Touch* sensor provides a useful way to measure the pressure of a handclap or foot stomp.

An 8-bit microcontroller with 364 bytes of user RAM and 8 KB of program ROM is used on the sensor module; with careful programming this was found to be sufficient for our application. The sensor prototype module is sufficiently small and light to be worn comfortably on the belt.

4 Protocols and software

The protocol software stacks for the codeBLUE system are shown in **Figure 2**. The software components for implementing the codeBLUE system consist of two parts: the low-level software running on the Bluetooth wireless access network (i.e., wireless sensor modules and access points) and the *CBPlayer* application running on the codeBLUE Controller PC.

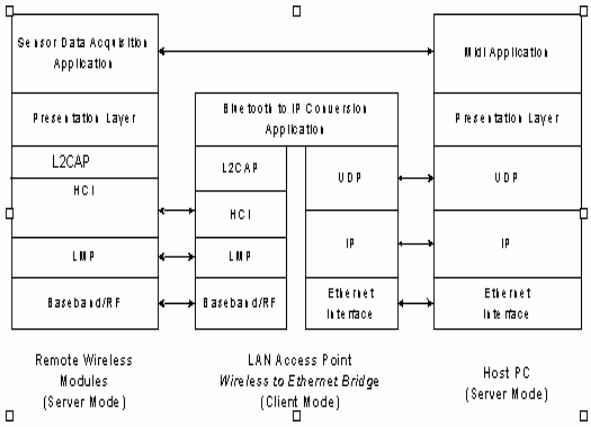

Figure 2: Protocol software stacks on the wireless sensor module (Remote wireless modules), the access point, and the codeBLUE Controller PC (Host PC)

4.1 Bluetooth network software

The sensor module application is a slave from the point of view of the Bluetooth protocol. After initialization the program instructs the Bluetooth module to enter the Pagescan mode where it listens to Page packets from the master, i.e., the access point. It waits until a connection is established with *BT2IP*, the Bluetooth-to-IP converter running on the access point, and periodically sends it sensor data. The sensor data is sent as a Bluetooth packet and includes a sensor ID value and the sensor data value consisting of the 7-bit digitized reading from the corresponding sensor.

The BT2IP application on the access point establishes connections with sensor modules by initializing the Bluetooth module and then instructing it to enter the Page mode. It subsequently waits for sensor data and relays it over the Ethernet using UDP.

To save memory and processing time we do not implement the IP protocol, or all the layers of the Bluetooth stack. Higher Layers like RFCOMM, OBEX, TCS etc. [Bluespec] are not necessary for transmitting simple sensor data packets and are not implemented. Thus the wireless sensor and the access point communicate using only L2CAP packets.

4.2 CBPlayer application

The CBPlayer application running on the codeBLUE Controller is the brain of the Interactive Dance Club System, connecting the dancers' movements to the music and essentially creating a virtual musical instrument that multiple dancers can simultaneously play to create music.

CBPlayer consists of a module to receive sensor data from access point(s), a standard MIDI file player, a *Data Filter* that modifies the MIDI stream in response to sensor data, and a DJ's configuration interface called the *Configurator* for defining the correspondence between

sensors and musical outputs along with other program parameters.

Data Filter. The Data Filter can insert MIDI events in the MIDI output stream, or modify existing MIDI events. The algorithms in the Data Filter must be such that the dancer-initiated modifications to the music are subtle enough to keep the music from becoming inaccessible, but at the same time responsive enough to keep the interest of the participants. We have developed a set of such heuristic algorithms; we omit details due to lack of space.

Configurator. The Configurator not only allows use of the system in different installations by specifying basic parameters appropriately, but also augments creative possibilities for new interactive performances. For instance, by including a configuration system, an elbow bend sensor doesn't always have to be mapped to rhythm modification. The Configurator GUI allows the DJ to both see and change how the mappings of sensor to algorithm to MIDI channel occur.

5 Discussion: Related Work & Lessons Learned

We first briefly mention related work, and then discuss some lessons learned. The key challenges of codeBLUE lie in two areas: (1) converting dance to music in a meaningful and aesthetically pleasing manner, and (2) system cost, usability and choice of wireless technology. Our discussion focuses on the latter.

5.1 Related Work

There have been a few projects and prototypes dealing with collaborative music systems. These include The Brain Opera [Machover96, Machover00], The Interactive Dance Club [Synesthesia98], Cybershoe [Paradiso98], CosTune [Nishimoto01] and Music in Motion [Ng00, Ng01].

While codeBLUE shares the goal of collaborative music with these projects, it differs primarily on account of the use of a frequency hopping, low power and low-cost wireless technology, namely Bluetooth. Brain Opera, and the Interactive Dance Club do not use wireless technology. Unlike codeBLUE, in CosTune there is no notion of a shared dance club environment that is playing a single song that all users interact upon, so it is likely to be harder to use with many participants. Also, the ad hoc nature of the network may make collaboration harder due to connectivity, delay and hidden terminal problems, and implementation using 802.11 technology may raise the cost. Unlike Cybershoe, the codeBLUE system is not limited by the fixed-frequency allocation so can scale to more participants, and the use of standard Bluetooth low-cost technology is preferable. The codeBLUE prototype also incorporates different types of sensors and affects sound as well as light, and allows the DJ to modify the dance-to-music mapping.

5.2 Cost and usability

The cost of the system can be broken down into four main components: wireless sensor modules, access points, the codeBLUE controller (a desktop PC), and the sound/light equipment. In our prototype these cost about $500, $1000, $3000 and $4000 respectively. We estimate a commercial implementation would drop the cost of the sensor and access point to about $10 (or less) and $200 respectively.

The system implementation took roughly 24 person-months of effort over about 6 months.

The key component of usability is the wireless sensor module. The prototype module measures about 7 cm x 5 cm x 3 cm (without the sensor itself) and weighs about 100 g. Using surface-mount components, a one-chip Bluetooth module and a Li-Ion battery will reduce the size by about a factor of four, making the module unobtrusive.

Thus our experimental experience indicates that the cost and usability of the system is satisfactory.

5.3 Choice of wireless technology

One of our conclusions from the codeBLUE implementation experience is that while Bluetooth technology is adequate for this type of application, it is not ideal.

The codeBLUE application has a particular set of requirements that differs from those often considered in the research literature, e.g. for wireless voice or web browsing. The amount of data from each sensor is very small, leading to bandwidth requirements (a few kbps, depending on the sensor) that are much lower than voice. However, end-to-end delay bounds of about 100 ms must be met to enable interactive modifications to music while it is playing. Low power and low cost are also key requirements. It is worth pointing out that critical components of the total cost of the system are not the sensor modules alone but the codeBLUE software, as well as installation and maintenance of the hardware, including power and LAN cables. We observe that these requirements may be representative of other ubiquitous computing applications in general.

We considered the use of IEEE 802.11b technology, currently a fashionable choice for wireless data as well as voice applications. However, 802.11b is likely to be even a worse choice than Bluetooth in some respects, since both its cost and its power consumption is generally higher and it currently provides no delay guarantees. In addition, although bandwidth is not a bottleneck, the bandwidth efficiency of 802.11b for such small data packets (~10 bytes) is low.

Bluetooth has its own set of pros and cons. Bluetooth can provide delay bounds and has low cost and power consumption. While it has relatively low bandwidth per cell (sometimes pointed out as a disadvantage compared to 802.11b), this is not a problem for codeBLUE. In addition, bandwidth can be utilized efficiently using Bluetooth for the small sized codeBLUE packets; if the Bluetooth specification allowed smaller

slots this efficiency would increase still further. These factors make Bluetooth adequate for codeBLUE, as we have stated.

However, Bluetooth is limited in that each piconet (of 10-30 m radius) can support at most seven simultaneously active slaves. This is far less than the person density in a popular dance club! Smart scheduling techniques which place slaves into standby modes (PARK, HOLD, etc) could expand this limit [Adamou03], but the required algorithms are quite complex. An alternative would be to use many access points; while the cost of the access point itself would be low, the cost of installation, including power and LAN cables, would be high. In principle LAN cables can be avoided if the access points form a scatternet, but scheduling transmissions in a scatternet to meet delay bounds seems overly complex, particularly for an application such as codeBLUE. Similarly, power cables can be avoided by using batteries, but maintenance is then an issue in any real-world deployment, especially when a large number of access points have to be deployed due to active-slave capacity constraints.

In general we believe Bluetooth is needlessly complex for an application such as codeBLUE. It is likely IP (perhaps with header compression) will be used for many applications such as codeBLUE mainly because of the simplicity and ubiquity of the IP protocol. However, Bluetooth does not support IP easily, and a fairly complex protocol stack is required. (Recent progress in the Bluetooth standards group may make support for IP easier, but it is not likely to be as simple as, say, 802.11b.) Also while the slotted operation of Bluetooth provides good bandwidth efficiency and delay guarantees, it does lead to more complex protocols (reflected in more expensive firmware).

Finally, while not a critical concern for codeBLUE itself, neither Bluetooth nor 802.11b offer satisfactory security solutions.

Our experience with codeBLUE suggests that there may be need for a different radio protocol design for applications such as codeBLUE, one that makes different trade-offs from those made by current commercially popular wireless LAN protocols.

6 Conclusion

We have proposed, designed and prototyped a novel application of Bluetooth technology. The prototype is end-to-end complete and has been successfully demonstrated.

Our experience demonstrates that Bluetooth can is adequate to develop innovative applications like codeBLUE while meeting cost, usability, power, and performance criteria. However, as discussed in sec. 5.2, an open research issue is whether a different radio protocol can be developed for such applications in order to reduce cost and complexity still further, to provide security, and yet meet ease of programming and delay requirements.

The codeBLUE system is being further developed as a case study for research in Bluetooth security [Jakobsson01] and high assurance of containment policies for security by static analysis to minimize performance costs of run-time access controls [Banerjee02,Banerjee02a].

REFERENCES

[Adamou03] M. Adamou, F. Anjum, and R. Jain, Helpers and Sleepers: Master assisted auxiliary piconet formation in Bluetooth, submitted for publication, 2003.

[Banerjee02] A. Banerjee and D. Naumann: Representation independence, confinement and access control, *Proc. ACM Symp. on Principles of Programming Languages (POPL)*, pp. 166-177, 2002.

[Banerjee02a] A. Banerjee and D. Naumann: Secure information flow and pointer confinement in a Java-like language, *Proc. 15th IEEE Computer Security Foundations Workshop (CSFW)*, pp. 253-270, 2002.

[Bluespec] Bluetooth Specification, http://www.Bluetooth.com

[Bluebook01] B. Miller and C. Bisdikian, *Bluetooth Revealed*, Prentice Hall, NJ 2001.

[Gong99] Li Gong: *Inside Java 2 Platform Security*, Addison-Wesley 1999.

[Machover96] T. Machover, Brain Opera. http://brainop.media.mit.edu

[Machover00] T. Machover, Hyperinstruments, http://www.media.mit.edu/hyperins/projects

[Nishimoto01] K. Nishimoto, T. Maekawa, Y. Tada, K. Mase and R. Nakatsu: Networked Wearable Musical Instruments Will Bring A New Musical Culture, *Proc. 5th Intl. Symp. on Wearable Computers (ISWC2001),* pp.55-62, 2001.

[Ng01] K.C. Ng., Music via Motion: Trans-Domain Mapping of Motion and Sound, *Proc. Intl. Workshop on Human Supervision and Control in Engineering and Music*, Kassel, Germany, September 2001.

[Ng00] K. C. Ng, S. Popat, E. Stefani, B. Ong, D. Cooper, and J. Smith-Autard, Music via Motion: Interactions between Choreography and Music, *Proc. 10th Intl. Symp. on Electronic Art (ISEA2000),* Paris, France Dec 2000.

[Paradiso99] J. Paradiso, E. Hu and K. Hsiao: The CyberShoe: A Wireless Multisensor Interface for a Dancer's Feet, *Proc. Intl. Dance and Technology 99*, Tempe AZ, Feb 26-28 1999.

[Synesthesia98] Interactive Dance Club. *SIGGRAPH*, June 1998, http://www.elkabong.com/IDC/idc.html.

[Jakobsson01] M. Jakobsson and S. Wetzel: Security Weaknesses in Bluetooth, Proc. Progress in Cryptology, RSA Conference (CT-RSA) pp. 176-191, 2001.