Mobile Audio Technology

Report and Recommendations

Prepared by the
Mobile Audio Working Group of the
Interactive Audio Special Interest Group

September 2007

Published By:
MIDI Manufacturers Association
Los Angeles CA

Disclaimer:

The MIDI Manufacturers Association (MMA), Interactive Audio Special Interest Group (IASIG), and their members and/or participants in the Mobile Audio Working Group shall not be held liable to any person or entity for any reason related to the adoption or implementation of, nor adherence to the recommendations in, nor any other use of this document nor any accompanying software.

Copyright © 2005-2007 MIDI Manufacturers Association Incorporated

ALL RIGHTS RESERVED. NO PART OF THIS DOCUMENT MAY BE REPRODUCED IN ANY FORM OR BY ANY MEANS, ELECTRONIC OR MECHANICAL, INCLUDING INFORMATION STORAGE AND RETRIEVAL SYSTEMS, WITHOUT PERMISSION IN WRITING FROM THE MIDI MANUFACTURERS ASSOCIATION INCORPORATED.

PRINTED 2007

MMA
POB 3173
La Habra CA 90632-3173

Table of Contents

THIS PAGE IS INTENTIONALLY BLANK

1. Executive Summary

This report looks at the task of developing and producing mobile audio, and, where possible, makes specific recommendations for simplifying development, improving quality and performance, and encouraging market growth through widespread adoption of minimum recommended levels of audio support across multiple mobile platforms.

1.1 Background

The market for mobile devices and content is exploding. Ringtones, ringbacks, videoringers, wallpaper, games, and music all present significant business opportunities for audio content developers and carriers, but fragmentation is adding significant cost and confusion to the market, much of which may be unnecessary.

Just a few years ago, audio on mobile devices was limited to simple “monophonic” ringtones (basically a modulated sine or square wave). In 2007, the capability of these devices varies widely depending on a number of factors such as CPU power, operating system and production date. Device manufacturers offer dozens of handset models based on standard and proprietary technologies, and there are now a number of differing approaches to playing sounds, including polyphonic MIDI (such as SP-MIDI and General MIDI) and a variety of digital audio recording formats (such as ADPCM and MP3.) One content developer estimates over 400 different audio implementations on mobile devices. The plethora of file formats and platform variations requires content providers to maintain multiple tools, and requires a substantial investment in products testing. The investment in education alone is a significant barrier to entry, which further restricts market growth.

Furthermore, inconsistent implementations and inadequate audio designs increase the cost of audio production by forcing developers to work harder just to make their content sound as good as possible. It is reasonable to expect that the audio capabilities of mobile devices will increase over time, and additional complexity will follow unless action is taken now to stabilize the market.

We also feel it is important to note that despite rapid increases in power and complexity, mobile phones will remain significantly less powerful and contain much less storage capacity than dedicated portable gaming platforms (e.g. PSP) for a few more years. And given the size of a typical mobile game (and the bandwidth required to download it to a phone), streaming audio game soundtracks are not a viable option today. Therefore, mobile game music will continue to be rendered in a resource-constrained environment, suggesting the need for MIDI and/or Mobile DLS sampled instrument tracks.

As professionals working with audio for interactive entertainment, this document is our attempt to describe how audio development for mobile devices falls short of our expectations. We provide recommendations that are a framework for future standardization of audio in the wireless space. As a group, we are aware of the value carriers and handset makers may place on their own IP, but we feel that non-proprietary standards are possible and necessary in this market.

1.2 Issues

The following factors contribute to lower productivity, higher development cost, and reduced quality of audio content on mobile devices:

- Too many platforms/technologies
- Handsets with partial implementations and/or incomplete documentation for existing formats (both hardware & software)

- Handsets that only play one sound at a time, with no ability to mix sounds that are triggered "on the fly" by user interaction.
- Missing, inadequate and/or incomplete development tools
- Lack of reliable and affordable device emulation and/or access to handsets for testing.

1.3 Recommendations

We urge manufacturers to prevent further segmentation of mobile audio platforms by agreeing on a minimum base level of audio support for devices. Such an agreement would foster greater availability of applications, and provide a better audio experience for consumers using mobile handsets:

- SP-MIDI, Full GM Level 1 recommended
- Minimum of 4 simultaneous mono digital audio streams
- Minimum mDLS and mXMF support (future "iXMF" support recommended)

We also recommend:

- Full implementation of the JAVA JSR 135 and JSR 234 specs, or where JAVA is not used, then provide access to file system and audio streams
- Open and accurate documentation for current hardware & software
- Accurate and affordable audio emulation tools
- API access to <u>all</u> audio features in the device.

Finally, we recommend handset makers and carriers participate in the IASIG to help foster productive dialog between hardware, software, systems, and content designers.

IASIG Mobile Audio Working Group
September 2007

2. Audio for Mobile Games

Gaming on mobile devices is a growing industry. However, mobile phone interactive soundtracks are usually not as robust as soundtracks on dedicated gaming platforms like Gameboy and PSP. Granted, audio resources on mobile phones are currently extremely tight due to delivery bandwidth and power constraints, but as these limitations decrease, it will be important to have high-quality game audio on such devices.

2.1 Background

There are a number of technologies for delivering interactive audio available to developers of mobile phone games, each supporting a variety of audio formats. The following technologies are presented as a snapshot of the industry as of this writing and may not contain all available technologies. The listing of these companies and their products does not constitute an endorsement by the IASIG or MMA management or membership.

2.1.1 Platforms

The following are development platforms or operating systems (OS) that are often integrated by mobile device manufacturers, or for which mobile application developers write their applications. Mobile applications can be written to run natively within an OS or to run in another technology (e.g. Java), which in turn can run within an OS. Usually, a mobile device will have support for only one of these platforms present but that is not always the case. For example, the US carrier Verizon almost exclusively uses the BREW operating system, but Java can run inside of BREW.

WindowsCE is an operating system from Microsoft developed for embedded environments and is the core of Windows Mobile devices.

Windows Mobile is an operating system component of WindowsCE with the ability to be tailored for specific types of devices.

Symbian is an operating system designed specifically for smartphones.

Linux is an operating system used on both desktop PCs and embedded devices.

Qualcomm's BREW APIs can also provide a number of useful game audio functions. The technology can handle several types of audio and video objects, and play up to 4 WAV or QCP (Qualcomm's PureVoice compressed audio) files, and 4 MIDI tracks simultaneously. Program changes and other MIDI commands can be sent via API calls to dynamically modify the soundtrack during game play. The platform also supports DLS sounds and instruments, so that multiple MIDI files can access the same custom instruments in order to save space.

The Java 2 Platform, Micro Edition (J2ME) provides an application environment for mobile devices that game developers can use to produce “write once, play anywhere” content. The MultiMedia API allows access to numerous basic audio functions (i.e. sample playback) and is designed to be content agnostic and extensible. More importantly, the platform also integrates easily and efficiently with other audio systems.

2.1.2 Technologies

These technologies often enable additional audio capabilities that may not be present natively or that may be functionally limited in a platform or OS. The products and companies are listed alphabetically by company, shown here for example, not meant to be an exhaustive list, and not endorsed by IASIG/MMA. Any and all trademarks belong to their owners and may not be indicated herein.

AM3D offers three mobile audio solutions for 3D audio: **Zirene** for stereo widening using HRTF filters; **Diesel Power FX Player**, which allows spatial positioning of existing ringtones; **Diesel Power Mobile**, which is a 3D audio game engine.

Beatnik offers **mobileBAE**, a comprehensive standards-based software solution for enhanced mobile audio, designed and highly optimized for mobile phones. Highly configurable, mobile BAE has been deployed in half a billion phones from over a dozen leading manufacturers for playing ringtone, UI sounds, recorded music, and 3D sounds in many formats including MIDI, SMAF, MFi, SP-MIDI, Mobile DLS, Mobile XMF, MP3, and AAC. Some platforms have elected to also expose the mobileBAE API to game developers, enabling rich control over complex, interactive game soundtracks.

Faith, Inc. offers a mobile solution that has currently been implemented in more than 80% of ringtone-capable mobile phones in Japan. Faith's solution includes both proprietary and standard format support including MIDI, SP-MIDI, Mobile XMF, Mobile DLS, SMAF, and MFi, wavetable creation, and integrated support for digital audio, LED, vibration, 3D audio, image and text data. Faith has also partnered with QUALCOMM in development of their mobile audio solution. http://www.faith-inc.com

Focusbyte offers **Tuny Engine**, a cross-platform wavetable synthesizer with 3D positioning, Doppler and other features for WindowsCE, Windows Mobile and Symbian platforms.

MobileSynth by **Wave Arts** is a lightweight ANSI C software synthesizer for delivering ringtones and interactive sound effects on mobile devices with low cost 16- or 32-bit processors. It's compatible with the MMA's standardized formats (SMF, SP-MIDI, Mobile DLS/XMF) and can include optional processing such as **MobileMax** loudness maximization, **MobileVerb** and **MobileSurround** 3-D expansion.

Mobileer offers **ME3000**, a ‘C’ code based Mobile XMF, SP-MIDI, DLS compatible wavetable synthesizer for mobile devices.

QSound microQ is a compact, modular and highly efficient digital audio engine consisting of 3D positional audio (mQ3D), 3D stereo expansion (mQXpander), digital effects processing (mQFX), and a wavetable synthesizer (mQSynth).

Rohm offers the following mobile audio solutions: **Melody Sound Source** LSi is a wavetable synthesizer for mobile devices and supports 16-64 voice polyphony with EQ, Reverb, Chorus, and 3D Surround options; **Sound Path Selector** LSI Series is an audio mixer and selector solution; **Audio Decoder** LSI is an MP3/AAC/HE-AAC decoder.

Silansys’ MoMedia is a 64 voice polyphonic synthesizer for mobile devices and offers Doppler, Occulusion and 3D Sound spatialization.

Sonaptic offers **Sonaptic** Sound for 3D spatialization of stereo sources on mobile devices

SRS Labs offers two mobile audio solutions: **MobileHD** virtualizes (using HRTF filters) pre-rendered 5.1 or 3D positional material over two channels for mobile devices; **SRS XSpace 3D** enables placement of audio objects in a 3D space in the absence of an actual 3D positional engine.

Tao’s Intent Audio is a suite of mobile audio tools that include a wavetable synthesizer with SP-MIDI, DLS support, JSR135 compliance, a modular framework permitting 3rd party modules, and a codec framework for multiple formats.

2.1.3 Content Formats

The following are listed in no particular order, shown here for example, and not meant to be an exhaustive list nor an indication of any endorsement by IASIG/MMA. Any and all trademarks belong to their owners and may not be indicated herein.

Multimedia Messaging Service (MMS) - Popular in Europe, MMS is the 3GPP's multimedia version of the ubiquitous SMS text messaging services. While SMS only sends and receives short strings of text, an MMS message can contain text, graphics, audio, and even video, with playback controlled by the Synchronized Multimedia Integration Language (SMIL). MMS messages can sometimes be installed as multimedia ringtones, but are mostly shared by users as eCards, love notes, or other entertainment. Implementations of the MMS specification vary widely from phone to phone, making them somewhat unreliable and limited in usefulness.

Flash Lite - A scaled-down version of the ubiquitous Flash vector graphics Web plug-in, this format has been popular for multimedia ringtones and simple games on a variety of Japanese phones and carriers, including KDDI au service, DoCoMo imode, and Vodafone KK. As is frequently the case in the chaotic mobile environment, audio implementations of Flash Lite vary from platform to platform. Some devices support Flash's built-in audio engine (referred to as "native audio") to support WAV and MP3 formats, while others rely upon the phone's OS components (called "device audio") to play MIDI, SMAF or MFi files. The choice to enable native or device audio depends on requirements of the OEM or the carrier. Consequently, not all Flash Lite enabled devices support the same sound capabilities. Flash Lite provides simple "trigger sync" for sound and animation, meaning multiple media types can be fired off at the same time. This is adequate for, say, an on-screen explosion or short animated ringtone but not for "lock to picture" type synchronization over an extended period of time. Future versions of the software (2.0) will contain enhanced audio capabilities that utilize a scripting language to sync device sound with forced (standardized) frame rates.

Qualcomm's CMX (Compact Media Extensions) multimedia software enables developers to create customized ringtones and screensavers as well as audio and graphics for mobile applications and games. Through an authoring tool provided by Qualcomm, content developers can create content that enables time-synchronization of MIDI-based voice, text, music, graphics and animation. Features 72-polyphony, 44 kHz sampling rate wavetable and support for formats including MIDI, SP-MIDI, SMAF, MFi, Mobile DLS, Mobile XMF, MP3, AAC, AAC+. [http://www.cdmatech.com/products/cmx.jsp]

SMAF (.mmf) files – Yamaha's Synthetic music Mobile Application Format (SMAF) files are used with Yamaha hardware for ringtones containing low-resolution digital audio and/or MIDI data. A tool called the Synchronous Contents Authoring System (SCAS) facilitates creation of simple "banner style" graphics that play along with digital audio files, for multimedia ringtones, mobile greeting cards, karaoke applications, and other entertainments playable on phones using KDDI's "au" network since 2005.

XMF Meta File Format - The MMA's eXtensible Music File (XMF) format supports bundling of MIDI and audio data with other associated content and playback information in a standard container. There are different types of XMF files for different applications, including some for ringtones and MIDI playback on mobile phones. Many of the major audio engines support the format, but at this moment there are few tools available for creating XMF files but – a partial list is available on www.midi.org/xmf. Since XMF is an open standard and free for anyone to use, it holds great promise for mobile gaming. The following XMF Types were discussed in the creation of this report:

- **Mobile XMF**: The bundling of "Mobile DLS" content with SP-MIDI content as a "Mobile XMF" file is a good basis for delivering game sound, since it allows for the efficient packaging of sample-based custom instruments, sound effects, dialog, synchronized music premixes, and more, all using open standard formats. Mobile XMF has also received endorsement by 3GPP and has become a standard feature of many mobile phones.
- **Interactive XMF (iXMF)**: The (IASIG) is authoring a specification based on XMF ("iXMF") for bundling all of the content assets of a game along with all of the elements of an interactive score, to make it possible for content-interchange among a variety of platforms. Moreover, since iXMF will be an open standard, proprietary authoring tools would not be needed, and content authoring for interactive applications could become an opportunity for many more people than it is today.

“**chaku-motion”** video ringtones - Based on Windows Media or MPEG4 video, these popular items consist of MTV-style video clips, and have been available on DoCoMo imode phones since 2004. It seems likely that this type of format will eventually dominate the market as phones become powerful enough to support reasonable frame rates, and hi-speed networks plus removable media allow faster download and storage of large files. Creating “chaku-motion” (as well as “Chaku-uta” digital audio ringtones) will generally be a simple conversion / compression process from a hi-resolution original, allowing lots of content to be produced quickly and easily. Audio is simply the soundtrack of the video clip, and is created using the standard tools of the trade.

2.2 Problem Statements

2.2.1 Platforms & Technologies

- According to one developer, there are 400 different implementations of audio in mobile devices. This often means different versions of content must be produced to adequately meet consumer demand, with substantial additional cost of production and testing. Its also practically impossible to test on 400 different handsets..
- Providers expose their technologies differently at the API level. As a result, platform and game developers can’t effectively leverage lessons previously learned from other vendor’s offerings.
- In surprisingly many cases, good interactive audio technology that is already present within the phone platform is not exposed to application/game developers at all (the Symbian/Beatnik implementation is a rare exception). This illustrates that the mobile platform and handset companies apparently fail to grasp the importance of exposing full audio functionality to games and other applications.
- Exacerbating the problem, some of the mobile content format standards are limited in scope, and thus don’t begin to fully solve the problems addressed in this paper.

2.2.2 File Formats

- The dizzying array of file formats alone is a major concern: The IASIG has received reports of over 50 different file formats. Such a wide range of audio formats makes it difficult to produce and manage audio content, especially ringtones, diminishing profitability for content developers.
- For the ringtone market, both monophonic and polyphonic MIDI ringtone formats appear to be losing popularity to actual recordings of popular songs. But, polyphonic MIDI is still important for mobile games, just as it was on PCs until the system bandwidth and CPU power became sufficient to support realtime digital audio mixing.
- Future devices may need to have support for “legacy” formats to extend the life of existing content. MP3 ringtones in particular are expected to remain popular (which is of some concern because the MPEG specification encompasses numerous bit-rates and encoding schemes, not all of which are necessarily supported on all devices.) Ideally, a solution would be for device makers to agree to a standard set of supported encoding schemes so that content developers can be assured that their files will play correctly. However, given the astonishing array of manufacturers that would need to be involved, it is important to stress to manufacturers the long term value and savings that standardization enables for those porting to a variety of different handsets. Accepting a minimum level of audio capability benefits developers, handset manufacturers, and carriers by speeding time to market for mobile applications.

2.3 Recommendations

Adoption and inclusion of standardized inter-operable file formats, delivery platforms and audio APIs, is recommended. This group recommends the following actions to encourage more widespread adoption of these specific technologies:

- Games require additional data types that are not currently supported in Mobile XMF. A new variation on Mobile XMF should be standardized that allows ambitious game developers to store whatever resources they need within an XMF file.
- The minimum sound RAM requirement for Mobile DLS (7k / 15k) is insufficient. The IASIG recommends manufacturers provide at least as much RAM as is specified for audio clips in MMA's "Audio Clips for Mobile XMF Specification".
- Robust and inter-operable APIs across devices are required. For instance, capabilities that were optional in the MMAPI should be standard features on all future handsets. An example of this would be short MIDI message support which is sporadic by manufacturer. Another would be multiple sound emitter support such as two audio files playing at the same time within a J2ME environment. Some phones support MIDI and AMR (Adaptive Multi-Rate codecs) in separate audio objects, others don't.
- Device makers should strive to use standardized and open standard audio formats (e.g. WAV/AIF, SP-MIDI, MP3, AAC, XMF) whenever possible. The trend to include the open-standard SP-MIDI in newer devices is a good example to strive toward (SP-MIDI has been endorsed by the 3GPP organization.)
- Adopting standardized or open- standard audio formats reduces the need for content developers to produce multiple version of their content. It also facilitates the creative process because of the wide variety of audio production tools available for standardized formats.
- Proprietary, closed, non-standard formats should be avoided for the following reasons:
    1. They can create extra work for developers since multiple versions must be created, tested and deployed.
    2. Proprietary formats require proprietary tools, complicating the production process and adding cost.
    3. They flood the market with a variety of formats which may confuse consumers by presenting them with too many choices (this format but not that one, for that phone but not this one) or not enough choices (you can't get what you want because it isn't available in the correct format).
- XMF's flexibility and scalability makes it perfect for gaming. We strongly recommend that the mobile industry adopt the XMF standard.

3. Authoring Tools and Environments

3.1 Device Emulation

The ability to audition using a desktop workstation with the same sound bank as on the phone is an important factor for success. It is also important to know the general frequency response and range of typical phone built-in speakers, because a wide range of speakers are used by manufacturers.

Products such as Awave (a popular DLS authoring tool that can produce MobileDLS sampled instrument banks) can be driven by a MIDI routing utility like MIDI Yoke (on Windows systems), allowing game soundtrack authors to play MIDI files (using their favorite Windows MIDI sequencer) to access a DLS sound bank that is identical to the one on the target device.

Yamaha and some of the iMode phone producers make hardware emulators (tone modules that contain the same audio chipset and speaker as phone), but such solutions are relatively costly, especially when you consider how many different devices a developer has to support. Virtual synthesizers (e.g. VST, AU, DXi, plug-ins, etc.) would be ideal for content developers because they would be less expensive (in theory) and less cumbersome, while allowing the developer to work with their preferred DAW or MIDI sequencer.

Nokia Audio Suite 2.0 is a VST plug-in that emulates the wavetable synth and speaker qualities of many Nokia devices. The plug-in also imports DLS instruments, allowing for realtime previewing. [http://sw.nokia.com/id/8c5a4c9c-5b00-4116-b6e9-905d24db0b73/DS_Nokia_AudioSuiteV2.0.pdf]

QUALCOMM and Yamaha both include emulators in their authoring tools for their respective proprietary formats (CMX and SMAF), which emulate the various versions of their wavetable synths.

3.2 Scalability Testing for SP-MIDI

Designers who author SP-MIDI ringtones need the ability to audition the file at all polyphony levels. While this could be part of a specific device emulator, it might also be a feature that a sequencer developer could implement, or perhaps someone will develop a ‘generic’ SP-MIDI virtual synthesizer.

In some cases these tools are part of development packages as is the case with the aforementioned Nokia Audio Suite 2.0, and Faith’s SP-MIDI MIP Message Edit Tool which allows one to import an SMF, edit the MIP message, set the max polyphony of the virtual synth and preview the behavior.

3.3 Development Tools for Apple Computers

IASIG members have lamented the lack of mobile audio authoring tools available for the Mac OS. This is particularly true for authoring XMF files. With the exception of a forthcoming Faith/Moderati DLS+XMF packaging tool for the PC, authoring XMF files requires no less than three pieces of PC software (DLS editor, MIDI sequencer and Beatnik's Mobile Sound Builder) plus one or more MIDI utility programs. Sound designers and musicians have often used the Mac OS, so the fact that XMF files cannot be created on the Mac OS means that authors are forced to maintain PC hardware and tools. The extra expense and complications could hinder adoption of XMF by Mac OS users.

3.3.1 Problem Statements

- Reliable emulation tools don’t exist for all devices. Without large, continuous investments in handsets and testing hardware, it’s impossible to know how the audio sounds, or if the audio implementation works.

- Difficulty in locating tools and updates. Tools are spread among a multitude of handset and chipset manufacturers. Some are easily accessible while others are buried in labyrinthine web sites, or behind walls accessible only to industry partners, shutting out independent developers.
- Legacy support. Emerging markets in developing nations can sometimes require authoring for formats that are four to five years old. Tracking these down can be difficult if not impossible.
- Supporting non-standard audio formats. Digital audio formats are spread across an astounding range of sample rates, bit rates and compression standards. Authoring tools must support as many of these formats as possible to simplify the job of content creation.

3.3.2 Recommendations

- Create standardized device emulators in plug-in formats common among standard MIDI composition tools (VST, AU, DXi, RTAS, TDM, etc.)
- Coordinate (perhaps under the auspices of the IASIG) an effort to publish links to tools, specifications and important developer information in a central location.
- As part of the coordinated effort described above, work with hardware developers to standardize formats for digitized audio.
- Device manufacturers partner with plug-in and music sequencer developers to offer SP-MIDI VST, AU, RTAS, TDM and DXi plug-ins.
- Device manufacturers should provide emulation tools that accurately emulate the phones audio capabilities. Minimum capabilities include polyphony and instrument set.
- Mobile audio software developers should offer Mac OS based development tools.

4. Looping of MIDI

The ability to loop (seamlessly repeat) segments of music data is advantageous to mobile devices. It means files can be smaller, which makes for faster downloads and more efficient storage. Such files can also play seamlessly for long periods of time, and the muting and un-muting of instruments (tracks) in the files can create a great deal of musical variation. A single minute-long MIDI file containing multiple, complimentary, percussion, chord, and melody tracks can play for a very long time without ever repeating the same exact performance.

The lack of standardization for looping of MIDI data for ringtones and game audio is unfortunate. The MMA does not have a standard for defining loop points in standard MIDI files (SMF), so manufacturers that use SMF or similar formats (typically based on SMF) have had to implement proprietary schemes for marking loop points and controlling them during playback.

SMAF, MFi and CMX files are MIDI-like formats with custom loop instructions. (SMAF stores the loop data outside the sequence data, while MFi / CMX stores it as special events within the sequence data.) Other solutions depend on proprietary “loopstart” and “loopend” markers in the MIDI header track.

Java mobile developers have been known to simulate looping by reopening the player on completion of sound playback, which is almost guaranteed to create a hiccup at the loop point.

Developers that play MIDI using the Multimedia Messaging Service (MMS) have been stymied by varying implementations on different devices. For example, some phones allow the developer to loop a MIDI file accompanying an MMS slide show a specified number of times, while others will only play the MIDI file once, making it impossible to utilize this feature effectively (since you can’t predict whether the file will play looped or not).

There is an obvious and growing need for looping of MIDI data, and that since a number of the component manufacturers such as Yamaha, Faith Technologies and Beatnik are members of the MMA, it should be possible for the MMA, by way of an IASIG submission, to standardize MIDI looping within Mobile XMF or even within a MIDI file itself. Unfortunately, the MMA has considered this before, but has never received sufficient interest to produce results and the subject is now in the hands of the IASIG. Certainly, looping MIDI has been a mainstay feature of games developed for consoles and PCs in the 1980’s (though largely using proprietary markers and engines again) and to a lesser extent with Web-based games in the 1990’s, so perhaps the problem is simply that handset and platform companies have not heard from the mobile game developer community loudly nor clearly enough.

The IASIG currently has a proposal for a Working Group to develop a standardized method for producing seamless looping of MIDI files. This feature is considered invaluable to developers of resource constrained mobile game audio and MIDI-based ringtones (including XMF files). Pauses or hiccups between the end and start loop points are simply unacceptable. Seamless looping is also important when developing “remixer” applications, or when varying a game soundtrack by muting and unmuting tracks.

4.1 Recommendations

The IASIG or MMA must produce a formal standard for looping of SMF data. This standard must address the following issues:

- Looping must be seamless from end point to start point. It will be assumed that the sequence author will provide any and all program changes and controller data within the looping data so that playback will be as intended.
- Nested loops (loops within loops) should be implemented and the specification should adhere to the criteria set above (seamless from point to point, etc.)

- Conditional and Unconditional Branching (similar to logic programming branching) should be implemented.
- The method employed must be flexible to allow other looping considerations in future versions. These features might include loop iteration (the number of times the loop should be played before continuing past the loopend marker), multiple loops within one file, nested looping, branching to specific points within the file, et al.
- The loop standard must not break existing MIDI players (they should be able to play the content without looping).
- Evangelization of the Looping standard is key in order to gain the "critical mass" needed to adopt the standard.

(For more detailed descriptions of these features, please refer to the IASIG Interactive Loop & Branch for SMF Working Group Goals Document and the MMA's XMF Working Group draft specification for SMF Loop & Branch).

5. Synchronizing Audio, Vibra and LED Tracks

Most phones are equipped with vibration motors, LEDs and/or color screens so that users can be alerted to incoming calls by buzz and/or flashing lights. Embedding control of those features in a ringtone can significantly increase the “oh, cool” factor… for example, a “xmas carol” ringtone that flashes red and green with the music. Such “vibra/LED tracks” can also promote device sales: the Danger Hiptop (aka T-Mobile Sidekick) is frequently cited in the press for the “psychedelic light show” produced by its color LED display. Vibra/LED response can also be unique to each ringtone, thereby allowing users to identify callers by feel or sight alone.

When a ringtone is a MIDI file, turning LEDs on and off in sync with the music – even having different light intensities and/or different lights for different instrument -- is easy, because every musical event is already represented in the MIDI file as individual Note-on/Note-off instructions. But digital audio files (MP3, wav, etc.) don’t contain such instructions, so creating cool Vibra/LED displays for those ringtones isn’t as feasible. It’s possible to auto-generate vibra/LED information from peaks in the overall waveform – e.g. for every snare hit there will be an accompanying thump and/or LED flash – but this is a very limited level of expression compared to what a developer can do with MIDI data. So as polyphonic MIDI ringtones give way to digital audio ringtones, composers will have a hard time using vibra/LED features in creative ways, unless new solutions are developed.

One solution is to bundle and deliver Vibra/LED control data along with the audio data in some sort of standard file container. One company has produced a library of data for controlling Vibra/LED information (MIDI), which they can stream to 3G phones as MP4 files (MIDI and audio). But their solution is specific to certain devices and their use of MIDI for Vibra/LED control is not standardized, so this is not a solution that will drive content developers to embrace custom Vibra/LED tracks.

5.1 Standard Solutions

The MMA’s "Scalable Polyphony MIDI Device 5-to-24 Note Profile for 3GPP" Telephone Ring Vibrator Control specifies a Program Change to be used in a MIDI track to control the vibration motor, but implementation in devices is optional so its presence cannot always be depended upon. Moreover, this approach requires one MIDI channel be used solely for vibration control, such that syncing the vibrator to both the kick and snare drum would consume two of only 16 available MIDI channels.

In response to input from this working group, the MMA has recently developed and adopted a much more flexible and powerful standard for Vibra/LED control using a MIDI System Exclusive message. The “Mobile Phone Control Specification” [MMA/AMEI RP-046] provides a method of producing non-audio alerts that sync with ringtones. The method is flexible and extensible enough to cover a wide range of non-audio alert types (even those yet to be invented) including but not limited to:

- Vibration motor toggle [on, off, or Follow MIDI Channel]
- Vibration motor intensity [128 levels]
- Backlight toggle [on, off, or Follow MIDI Channel]
- Backlight color [RGB numbers]
- Backlight variable [128 levels]
- LED toggle [on, off, or Follow MIDI Channel]
- LED color [RGB numbers]
- LED brightness [128 levels]
- Multiple color LED displays

The standard supports simple “toggled alerts” (i.e. vibration motor on or off) as well as more complex alerts that respond to variable parameters. The method does not require dedicating a MIDI channel or MIDI controllers to non-audio alert control, so not only does it allow the full range of MIDI channels to be used for synthesis, but it also facilitates time alignment of alert controls by means of MIDI notes within a track (i.e. when syncing buzzers to kick drums).

The standard is also not limited to use with polyphonic MIDI ringtones. Using the “Audio Clips for Mobile XMF” format [MMA/AMEI RP-045] a ringtone can include a long audio clip in the form of a Mobile DLS instrument, plus a single Note On message to trigger said audio playback, plus MIDI data just for Vibra/LED (not MIDI notes).

Some platforms may want to auto-generate controller data for synced vibra/LED tracks based on analysis of the digital sample (tracking wave peaks, for example). For example, this feature would also be useful for mobile games when using the vibration motor as a force-feedback device or creating a vibra/LED in conjunction with explosions. While the actual algorithm used may be outside the scope of the MMA standard, the MMA method supports this purpose.

5.2 Recommendations

The IASIG recommends mobile handset and platform companies adopt the new MMA Mobile Phone Control Universal System Exclusive standard because it supports rich interoperable LED and Vibra functionality now, as well as providing for vendor-specific extensions where necessary, and new standardized features in the future. This standard can be used both with MIDI and, in container-based formats, digital audio content.

[N.B.: The IASIG technology recommendations are made without specific knowledge or consideration of intellectual property ownership in the technologies recommended, and implementers are advised that patents have been applied-for regarding some usages of MIDI.]

6. Future Directions

It is inevitable that cell phones will become more capable over time, as processor power increases (along with battery life) and devices are able to include more memory, better screens, and maybe even better speakers. Future phones may look more like today's PDAs, and offering similar programmability. Everyone agrees that no matter what they look like, they will have more and better audio features.

6.1 Digital Signal Processing

One future feature that we expect to see is the ability to improve the quality of audio using Digital Signal Processing (DSP.) This would typically be applied to the audio output stream to compensate for the specific deficiencies of the phone's speaker system. Thus, content providers could (within reason) use a wider variety of audio content on phones with some sort of assurance that it will be heard as accurately as possible.

EQ is one of the most useful DSP tools for mobile platforms. Some mobile audio solutions include an EQ that is permanently adjusted when implemented for a specific device's speakers. When such an EQ is present, it is not always taken advantage of by handset manufacturers. In some implementations, it could be difficult to affect enough simultaneous frequencies without using too many MIPS – but even marginal improvements would be welcome. Anything done on the phone itself would be preferred over expecting content developers to equalize content for different phone models. Other DSP tasks such as normalization, limiting, gain control would also be a good investment of CPU resources.

It was suggested that the IASIG should work with platform designers and content providers to determine a minimum recommended design for a suitable EQ subsystem. However, the size, shape and porting of the enclosure containing the speaker within the mobile device must also be carefully designed for best results, so it may not be possible to design a one-size-fits-all solution. In any case, an experienced audio engineer should determine the correct settings for each model and speaker.

6.2 3D Audio

3D audio is already gaining interest as a feature on mobile platforms for music listening enhancement and game audio. The presence of 3D audio, either for listening to music in surround sound or positional audio in mobile gaming, expands the user's experience beyond the physical dimensions of the device. While the small screen of phones limits the scale of the visual experience, 3D can compensate for this and deliver varying degrees of immersive audio experiences on speakers and especially on earbuds.

There are actually three distinct technologies that are encompassed by "3D Audio":

- Stereo Expansion a.k.a Stereo Widening – Used with music playback to create a "wider stereo" effect on stereo speakers, or "outside the head" (as opposed to within the head) effects on headsets. No API is required because the technology works on existing content (passive algorithm). Some expansion algorithms support the generation of pseudo 3D stereo from mono content.
- Virtual Surround Sound - Used to emulate multi-channel audio playback systems over stereo speakers or headsets, which may be interesting for mobile devices which playback multi-channel video content or MP3Surround audio.
- Realtime 3D Positioning - Used for interactive placement of sound effects in games. Works on mono sound effects (and changes stereo signals) to move sounds independently to arbitrary positions both within and beyond the normal stereo sound stage, i.e. outside the area bounded by the speakers, or "outside the head" on headsets. (More likely headsets for games on mobile

devices). Content developers must use an API to assign positions -- APIs are available with various platforms such as Vodafone (VFX Specification), NTT Docomo, QUALCOMM (Brew), JAVA JSR-234, and Open SL ES.

Because of the small size of mobile phones, stereo expansion is likely to be the most popular form of 3D for the near term. But as more handsets include stereo headphone jacks, realtime 3D and virtual surround will become more prevalent. Some recent trends include:

- 3D audio capabilities for mobile games is now a required handset feature for some carriers.
- Docomo has added 3D audio APIs to their specs and Vodafone has a 3D positioning API, Vodafone "VFX" (for which QSound Q3D is the reference implementation).
- QUALCOMM has also included 3D audio in their chip sets (using QSound technology) and enables the API through Brew.
- Major chipset manufacturers NEC (using FueTrek technology), Oki (using SRS technology) and Yamaha (MA-7) are incorporating 3D surround sound and sound positioning into melody chipsets intended for phones.
- A growing number of software mobile audio engines, such as Beatnik, QSound microQ, Wave Arts, Sonaptic, SRS XSpace 3D, support 3D sound positioning and expansion.
- The Java Community Process approved the final draft of JSR-234 "Advanced Media Supplements" in June of 2005. The audio portion of JSR-234 API gives J2ME midlets on compatible phones access to advanced audio processing capabilities like equalizer, audio effects, artificial reverberation and positional 3D audio.

6.3 Standards

In 2005 the Khronos Group established the OpenSL ES working group. OpenSL ES will be a royalty-free, cross-platform open standard, C language API for standardized access to features such as 3D positional audio, MIDI playback, and DSP effects intended for phones and consumer electronics devices. For more information visit http://khronos.org/opensles/.

6.4 The Ringtone Market

We expect some significant changes to the ringtone market over the next few years. Ringtones are an expression of personal style and taste, so we expect consumers to gravitate towards handsets and services that offer access to more content and provide more flexibility in how that content can be used.

One of the biggest drivers of change is the ability to use the cell phone as a portable music player. Phones that play files in the MP3/WMA/AAC formats are becoming more and more popular, as are services that provide songs for $.99 each or less. As a result, consumers will become reluctant to pay higher prices for lower-resolution, thirty-second ringtones … especially when they already have that song loaded on their phones. Carriers who are selling songs over the air (Verizon's V-Cast service, Sprint's Music Store, etc.), don't currently allow downloaded music to be used as ringtones — and we understand that carriers may preclude this functionality for now for licensing reasons and to protect their ringtone revenues — but we believe enabling the repurposing of music as ringtone content will become a market advantage. We also don't yet see many handsets that allow using stored-music files to be used as ringtones, but we expect that will change over time.

People often identify songs by the "hook" or some other identifiable section, and in terms of personalization nearly any 30-second portion of a song could be desirable as a ringtone. This leads us to believe there is potential for services and/or tools that provide ringtone customization. Something that would make development of this more practical and economical is the addition of “ringtone start” markers in recorded music files.

While we expect that the creation of “basic” music ringtones will eventually be in the hands of consumers — perhaps primarily in the form of “click here to use this song as a ringtone” — we believe there are still revenue opportunities for content producers, such as in celebrity voice-over ringtones, movie tie-ins, contests with ringtones as prizes, multimedia ringtones, etc.

Finally, if we haven’t said it enough times already, technical standards will help enable these new features and services, by making it more efficient and affordable for content providers to produce content for a large variety of phones.

7. References and Links

- "Could Ringtones BE More Annoying?!" (Peter Drescher). A transcript of the Mobile Audio presentation given to the 2004 Project Bar-B-Q
  http://digitalmedia.oreilly.com/2004/11/10/drescher_bbq04_ringtone.html
- Project Bar-B-Q Interactive Audio Conference
  http://www.projectbarbq.com/
- The MIDI Manufacturer's Association (MMA)
  http://www.MIDI.org
- Interactive Audio Special Interest Group (IASIG)
  http://www.iasig.org
- Additional information about companies offering audio engines and solutions for mobile phones can be found at: http://sonify.org/tutorials/mobile_audio/engines/.

8. Definitions

8.1 “Mobile Audio”

By 'Mobile Audio' we mean sounds produced by the operating system of a hand held device that communicates voice/data information wirelessly. Such devices include cell phones and “smart” phones (such as Treo, Blackberry, Ngage, and Hiptop devices), but not portable audio and gaming devices like the Sony PSP, Apple iPod, and XM radio terminals.

8.2 “Interactive” Audio on Mobile Devices

By 'Interactive’ we mean sounds that play in response to system events or due to user interaction (e.g. game audio and/or user-interface sounds).

8.3 “Ringtone”

For the purposes of this report, a 'Ringtone' is any alert (audible or not) produced by a mobile device to signal the arrival of an incoming event (e.g. phone call, text message, voice mail alert). This alert usually takes the form of audio (in the form of a piezo ringer, MIDI file rendering, or digital audio file), but can also include vibration motor buzzing, flashing lights, video or other signals.

8.4 “Handset”, “Phone”

Used interchangeably herein, to refer to the thing you hold in your hand for mobile communications.

9. Mobile Audio Working Group Participants

Peter Drescher (Chair)	Danger, Inc.
Preben Nielsen	AM3D
Chris Mitchell	Anglia Polytechnic University
Rob Cairns	Associated Production Music
Jeff Essex	Audiosyncrasy
Hayden Porter	Aviarts
Chris Grigg	Beatnik
Andrew Rostaing	Beatnik
Chris Burke	Bong and Dern
Ron Kuper	Cakewalk
Jean-Marc Jot	Creative Labs
Peter Clare	Creative Labs/Sensaura
Jack Buser	Dolby Laboratories
Mike Leahy	EGR
Mitsuo Matsumoto	FueTrek Co Ltd
Jim Rippie	Independent
Matt Levine	Independent
Devin Maxwell	Loud, Louder, Loudest
Alain Georges	MadWaves
Mike Curtes	Moderati
Steve Horowitz	Nickelodeon Online
Pascal Darre	Philips Semiconductors
Scott Willing	QSound Labs, Inc.
John Mortimer	QSound Labs, Inc.
Michael Klinowski	Shinnyo Interactive Audio
Brad Fuller	Sonaural
Kurt Heiden	SONiVOX
Rob Light	Spherex Inc
Dan Jochelson	Texas Instruments
Martin Wilde	Wildeworx
Luke Holden	Yamaha