Multimedia Basic Concepts - MIDI

Siddhartha Bhattacharyya*

1. Introduction

The advances in computer technology paved the way to the infrastructure for multimedia applications. With these developments the multimedia concepts have emerged. The word multimedia is made up of the two Latin words "multi" which means many and "media" which is the subtance through which something is transmitted. In this case multi is the multiple data types such as voice, video, image, animation ,text etc. and media is the computer environment used to transmit the information made up of this multiple data types.

The important thing is the computer environment, which involves the computer hardware such as the memory and the processing power, the operating system, the computer networks, and the storage media. Hence when we are dealing with the subject of multimedia all these issues listed above are of concern to us.

A multimedia product is a presentation, a tool or an interactive program, which is distributed in some kind of a storage medium to be viewed in a suitable computer environment. To prepare such a product, special tools are also needed. These are referred to as authoring tools. Multimedia deals with textual, audio and video data formats. These are generally handled in compressed forms.

2. Different Multimedia Data Streams

2.1 Text

Text constitutes the main part of a multimedia package. It is used to provide most of the information intended to be conveyed and it is even stated that other multimedia data types are used to enchace text.

2.2 Hypertext

Hypertext is usually defined as NON-LINEAR ACCESS TO TEXT using links embedded in the text. In this way going through material which is not of interest is avoided. Unfortunately, text is much harder to read from a screen than it is from paper, and our experience shows that people tend to print copies of information they are accessing from the computer. This may be due to habits or the effect the current displays have on the eyes. Using HYPERTEXT eases this problem to a certain extent in the sense that users navigating in the text relax their eyes occasionaly. If hypertext is not used carefully, however, it becomes difficult to follow the topics presented.

2.3 Hypermedia

HYPEMEDIA is the extension of the non-linearity to other media types. Using media other than text may require extra resouces compared to text but definitely makes it much easier to follow the material covered.

2.4 Images

Images improve the overall look of a presentation and they are useful to express informtion which text alone cannot vonvey. Using IMAGES and GRAPHICS can be very useful but it has to be noted that they also introduce extra load to the system both as storage and also as network traffic. Images come in different forms and resolution i.e. the number of picture elements used to represent them (pixels) and they are usually compressed using different techniques such as JPRG (Joint Photographic Expert Group). There are various image formats. Most widely used 2 formats are gif and jpg.

Lecturer, Information Technology

2.5 Video

A video clip or an animtion sequence can express an idea in a much better way than text or images. Video clips used in multimedia systems ar more flexible in the sense that they are EASIER TO EDIT and also EASIER TO ACCESS. DIGITAL VIDEO can be created from VCRs, using cameras or directly recorded from broadcasts. Either way they are captured stored and edited before they are used in multimedia presentations. Video is usually played at 25 or 30 frames per second and will have to be COMPRESSED in order to reduce it to acceptable sizes. Different compression techniques are used and MPEG (Motion Picture Expert Group) is one of these methods. Transferring this video over the networks especially the Internet is a hot research issue. Techniques are being proposed for this purpose and they will be dealt with in the coming sections. ANIMATION is similar to video, in that it uses the display of moving pictures to convey information. The pictures are constructed artifically, however, they can be very useful to explain abstract concepts. They also require huge storage space and packages used to produce them are usually difficult to learn. Video files are perhaps the most exciting and attrctive features of multimedia systems.

2.6 Sound

Sound is another data type used in multimedia applications. Sound requires more space than text but is better when compared to videp clips. People use it to express an idea in words ove the computer or it can also be used to introduce effects into a presentation. Either way it ENHANCES THE QUALITY of presentation and INCREASES THE EFFICIENCY of information transfer. Different audio fromats are in existence viz. wav, mid, mp3 to name a few. The next section deals with the basics of MIDI (Music Instrument Digital Interface) and the standard MIDI file format.

3. MIDI Basic Concepts

MIDI is an equipment between an electronic musical instrument and a computer, which can be plugged to the serial port. A MIDI interface has two components :

i) Hardware for connecting the equipment and enabling the MIDI port of the instrument and a cable for transmission.

ii) A data format for encoding MIDI stream. MIDI data allow encoding of 10 octaves and these correspond to 128 notes.

The MIDI data stream is in digital format and grouped into MIDI messages. Each measage communicates one musical event between machines. These events are actions such as keypress, slider movement etc.

When a key is pressed , a MIDI interface creates a MIDI meassage where the begining and intensity of the note is encoded. It is transmitted to another machine. When the key is released, the corresponding message is transmitted again.

A synthesizer is a MIDI device, which communiates with other machines through channels. There are 16 channels in all. There are several MIDI reception modes for tuning a device to one or more channels.

- Mode 1 : Omni On/Poly
- Mode 2 : Omni On/Mono
- Mode 3 : Omni Off/Poly
- Mode 4 : Omni Off/Mono

The first part of the reception mode determines as to how a device monitors the incoming channels. If Omni is On, a MIDi device monitors all channels. If Omni is Off, it responds only to a predetermined channel. The second part specifies as to whether to play one or several notes at a time.

3.1 MIDI Devices

There are five MIDI devices. These are :

i) Sound grenerators-to produce audio signals.

ii) A microprocesor-to communicate with keyboard as to which notes are being played and then sends them to the sound generators.

iii) Control panel-to control volume, tone and timbre of a MIDI note.

iv) Auxiliary Controls - to adjust pitch and modulation.

v) Memory module

3.2 MIDI File Format

A midi (.MID) file contains basically two things, Header chunks and Track chunks. A midi file contains ONE header chunk describing the file format, etc., and any number of track chunks. A track may be thought of in the same way as a track on a multi-track tape deck. One track can be assigned to each voice, each staff,and each instrument.

3.2.1 HEADER CHUNK

The header chunk appears at the beginning of the file, and describes the file in three ways. The header chunk always looks like :

4D 54 68 64 00 00 00 06 ff ff nn nn dd dd

The ASCII equivalent of the first 4 bytes is MThd. After MThd comes the 4 byte size of the header. This will always be 00 00 00 06, because the actual header information will always be 6 bytes. ff ff is the file format. There are 3 formats :

0 - single -track

1 - multiple tracks, synchronous

2 - multiple tracks, asynchronous

Single track is fairly self-explanatory - one track only. Synchronous multiple tracks refers to the tracks those are vertically synchronous, or in other words, they all start at the same time, and so can represent different parts in one song. Asynchronous multiple tracks do not necessarily start at the same time, and can be completely asynchronous.

nn nn is the number of tracks in the midi file.

dd dd is the number of delta-time ticks per quarter note,

3.2.2 TRACK CHUNKS

The remainder of the file after the header chunk consists of track chunks. Each track has one header and may contain as many midi commands as one likes. The header for a track is very similar to the one for the file :

4D 54 72 6b xx xx xx xx

As with the header, the first 4 bytes have an ASCII equivalent. This one is MTrk. The 4 bytes after MTrk give the length of the track (not including the track header) in bytes. Following the header are midi events. These events are identical to the actual data sent and reveived by MIDI ports on a synthesizer with one addition. A midi event is preceded by a delta-time. A delta time is the number of ticks after which the midi event is to be executed. The number of ticks per quarter note was defined previously in the file header chunk. This delta-time is a variable-length -encoded value. This format, while confusing, allows large numbers to use as many bytes as they need, without requiring small numbers to waste bytes by filling with zeros. The number is converted into 7-bit bytes, and the most-significant bit of each byte is 1 except for the last byte of the number, which has a MSB of 0. This allows the number to be read one byte at a time, and when there is a MSB of 0, it is the last (least significant) byte of the number. According to the MIDI spec, the entire delta-time should be at most 4 bytes long.

Following the delta-time is a midi event Each midi event (except a running midi event) has a command byte which will always have a MSB of 1 (the value will be >=128). Each command has different parameters and length, but the data

that follows the command will have a MSB of 0 (less than 128). The exception to this is a meta-event, which may contain data with a MSB of 1. However, meta-events require a length parameter, which alleviates confusion.

One subtlety, which can cause confusion, is running mode. This is where the actual midi command is omitted, and the last midi command issued is assumed. This means that the midi event will consist of a delta time and the parameters that would go to the command if it were included.

3.2.3 MIDI EVENT COMMANDS

Each command byte has two parts. The left nibble (4 bits) contains the actual command, and the right nibble contains the midi channel number on which the command will be executed. There are 16 midi channels, and 8 midi commands (the command nibble must have a MSB of 1). In Table 1, x indicates the midi channel number. Note that all data bytes will be <128 (MSB set to 0).

Table - 1

Hex	Binary	Data	Description
8x	1000xxxx	nn vv	Note off (key is released) nn=note number vv=velocity
9x	1001xxxx	nn vv	Note on (key is pressed)
Ax	1010xxxx	nn vv	Key after-touch
Bx	1011xxxx	cc vv	Control Change cc=controller number vv=new value
Cx	1100xxxx	pp	Program (patch) change pp=new program number
Dx	1101xxxx	cc	Channel after-touch cc=channel number
Ex	1110xxxx	bb tt	Pitch wheel change (2000H is normal or no change) bb=bottom (least sig) 7 bits of value tt=top (most sig) 7 bits of value

FF xx nn dd

All meta-events start with FF followed by the command (xx), the length, or number of bytes that will contain data (nn), and the actual data (dd).

Table - 2

Hex	Binary	Data	Description
00	00000000	nn ssss	Sets the track's sequence number. nn=02 (length of 2 byte sequence number) ssss=sequence number
01	00000001	nn tt ..	Text event-any text one wants nn=length in bytes of text tt=text characters
02	00000010	nn tt ..	Same as text event, but used for copyright info. nn tt=same as text event

03	00000011	nn tt ..	Sequence or track name nn tt=same as text event pressed)
04	00000100	nn tt ..	Track instrument name nn tt=same as text event
05	00000101	nn tt ..	Lyric nn tt=same as text event
06	00000110	nn tt ..	Marker nn tt=same as text event
07	00000111	nn tt ..	Cue point nn tt=same as text event
2F	00101111	00	This event must come at the end of each track
51	01010001	03 tttttt	Set tempo tttttt=microseconds/quarter note
58	01011000	04	nn dd ccbb Time Signature nn=numerator of time sig. dd=denominator of time sig. 2=quarter 3=eighth, etc. cc=number of ticks in metronome click bb=number of 32nd notes to the qurter note
59	01011001	02 sf mi	Key signature sf=sharp/flats (-7=7 flats, 0=key of C, 7=7 sharps) mi=major/minor (0=major, 1=minor)
7F	01111111	xx dd ..	Sequencer specific information xx=number of bytes to be sent dd=data

Table 3 lists system messages, which control the entire system. These have no midi chanel number. (These will generally only apply to controlling a midi keyboard, etc.)

Tabl - 3

Hex	Binary	Data
F8	11111000	Timing clock used when synchronozation is required
FA	11111010	Start current sequence
FB	11111011	Continue a stopped sequence where left Off
FC	11111100	Stop a sequence

Table 4 lists the number corresponding to notes for use in note on and note off commands.

Table - 4

Octave	Note Numbers											
	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
0	0	1	2	3	4	5	6	7	8	9	10	11
1	12	13	14	15	16	17	18	19	20	21	22	23
2	24	25	26	27	28	29	30	31	32	33	34	35
3	36	37	38	39	40	41	42	43	44	45	46	47
4	48	49	50	51	52	53	54	55	56	57	58	59
5	60	61	62	63	64	65	66	67	68	69	70	71
6	72	73	74	75	76	77	78	79	80	81	82	83
7	84	85	86	87	88	89	90	91	92	93	94	95
8	96	97	98	99	100	101	102	103	104	105	106	107
9	108	109	110	111	112	113	114	115	116	117	118	119
10	120	121	122	123	124	125	126	127				

4. Conclusion

The topic of multimedia also entails several other features viz. Multimedia Operating System, Multimedia Database and Multimedia Communications. Each feature in itself is a subject on its own. Moreover, the sphere of Multimedia Applications is also an upcoming area to work on. Those days are no more remote when people would be dealing with intelligent multimedia systems.

'But words are things, and a small drop of ink,
Falling like dew, upon a thought, produces
That which makes thousands, perhaps millions, think.'

- Lord Byron