Real-time Cross-adaptive Audio Effects

Queen Mary University of London

School of Electronic Engineering and Computer Science

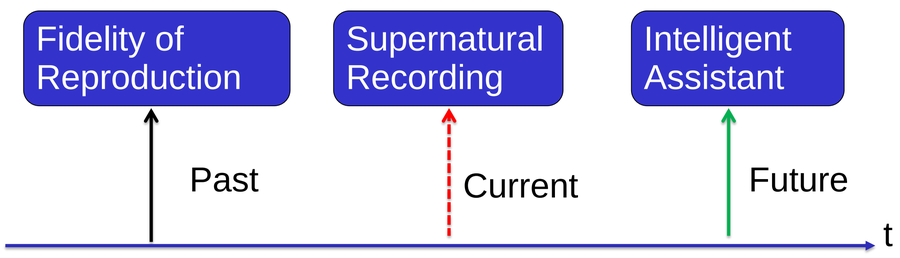
London E1 4NS, UK

Author: Yonghao Wang; Supervisor: Josh Reiss

yonghao.wang@eecs.qmul.ac.uk

centre for digital music

1 .: INTRODUCTION :.



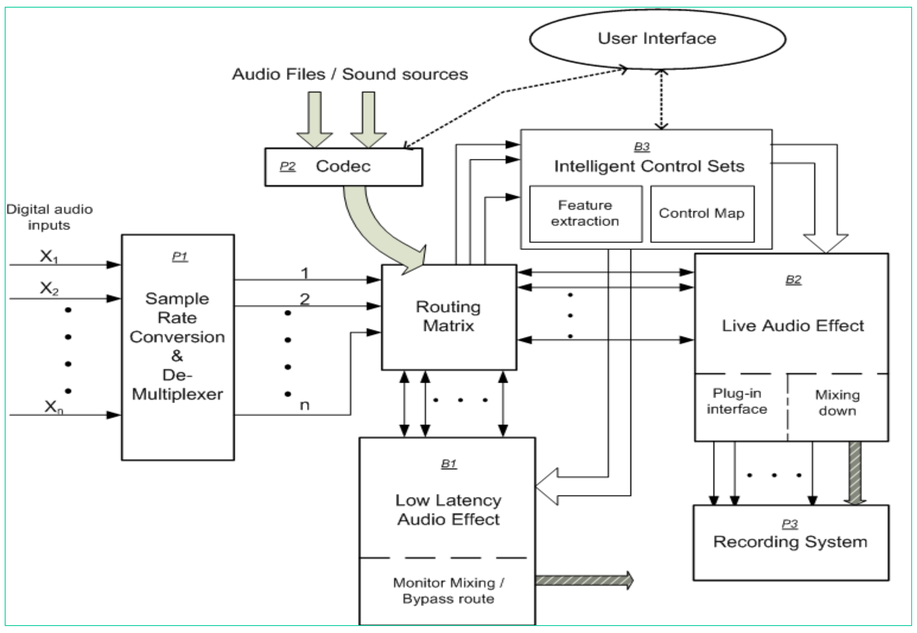
•James Moorer (2000) described three stages of Music Production which can be depicted as above diagram.

• "**Intelligent Assistant"** is his extrapolation for the future:

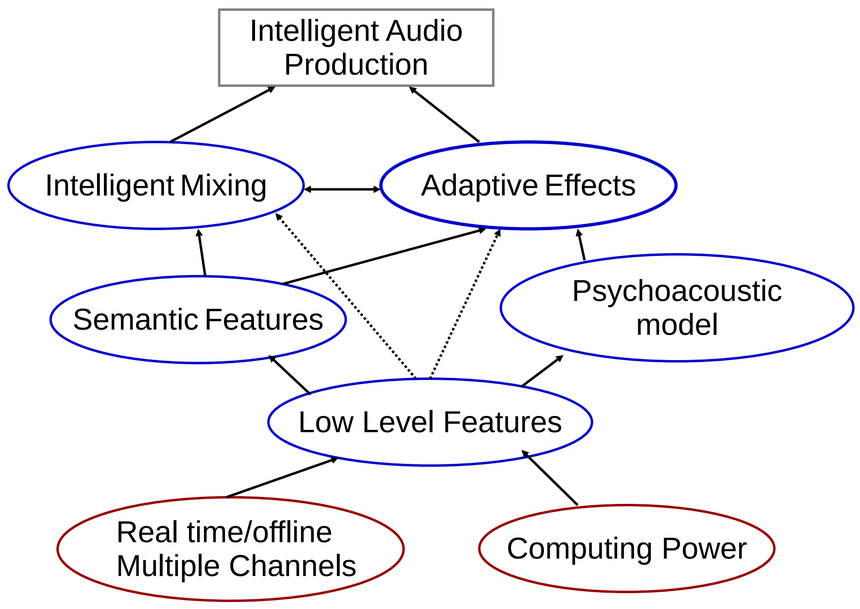
1. Processing 9000 Channel in Real-time.
2. Doing million points FFT.
3. Beyond capability of a proficient audio engineer.

2 .: THE PROBLEM :.

1. Our research focuses on the Real-time intelligent assistant based on cross-adaptive audio effects.
2. Develop the schemes of organising features over multi-channel.
3. Synchronise analysis path with audio processing stream.
   - Michael Lester et al.,(2007) indicates the low latency (<10ms) requirements for some music applications.
   - It is computational costly for feature extractions over multiple channels within real-time constraints.
4. Raw computing power has met power wall since 2006, to maintain the moore's law, the parallel structure needs to be considered.



3 .: OUR APPROACH :.



1. Multi channel feature extractions are computational costly under real-time constrains.
2. To gain the maximum performance and ensure the future scalability, the multi-level parallelism needs to be considered.
3. Hardware architecture/micro-arch level, memory/cache coherency, ILP e.g. SIMD.
4. Operating System level, distributed threads, processes scheduler.
5. Libraries: Math Library optimised for multi-cores such as FFT, Correlation.
6. Applications: algorithm sets, software architecture, re-entrant threading safe plug-in etc.

Three Steps approach towards Application level parallelism:

Step 1: Evaluate whether the current digital audio processing platform could maintain the steady audio processing latency regardless of CPU load.

Step 2: Identify the real-time characteristic of audio effects and feature extractions such as the latencies of real-time audio effects and feature extractions.

Step 3: Evaluate the parallelism software structure of preferred implementation using different parallel architecture.

4 .: RESULTS :.

- Latency measurements of popular operating systems with onboard soundcard, default audio API, common audio DAWs, and various CPU loads.
- The latency of common PC based DAW is comparable with audio consoles.
- CPU loads has little effects on latency!

Host	OS (API)	Without Load	With Outside load	With Audio load
2.b	WinXP (DirectX)	73ms (buffer 512)	81ms (buffer 512)	104ms (buffer 512)
2.a	OS X (CoreAudio)	4ms (buffer 14*2)	4 ms (buffer 14*2)	5.80ms (buffer 14*2)
4	Linux (ALSA)	3.31ms	3.31ms	error
		22ms (buffer 512*2)	22ms (buffer 512*2)	22ms (buffer 512*2)