The Show Must Go On: Dependable Audio Computing

Jörn Loviscach

Fachhochschule Bielefeld, Germany
(University of Applied Sciences)

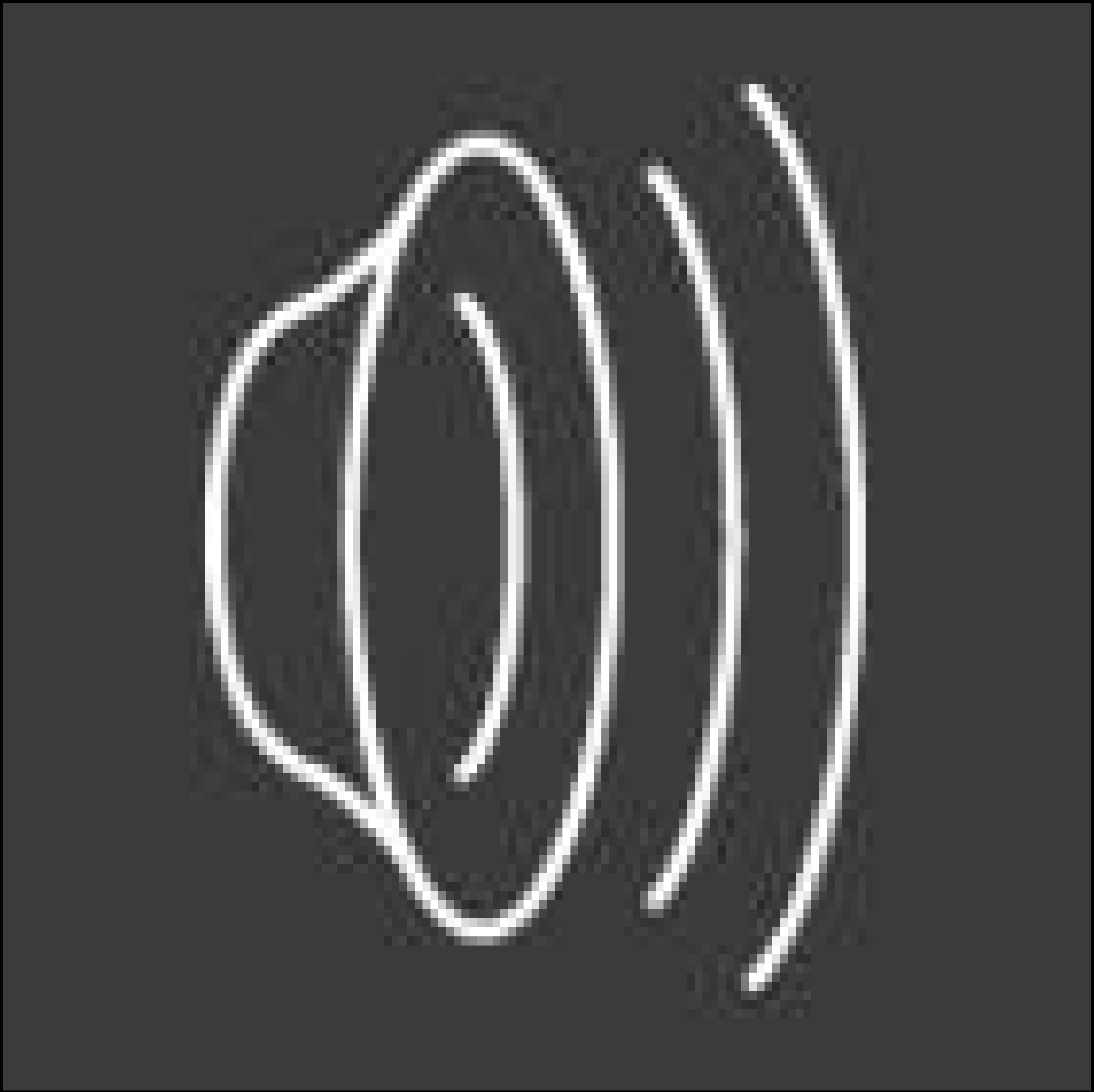
An invitation to a virtual unconference

bit.ly/9kIvlT

URL will be repeated at the end

What is this about?

SYSTEM ERROR



Computers are dangerous!

Looking for best practices
& inventing new ones:

- Live music and multimedia performances
- Recording unique moments

Applications of off-the-shelf products
& development of proprietary solutions

Computers are dangerous!

Looking for best practices
& inventing new ones:

Support for artists and engineers

- Live music and multimedia performances
- Recording unique moments

Applications of off-the-shelf products
& development of proprietary solutions

Computers are dangerous!

Looking for best practices
& inventing new ones:

Avenues
for research

- Live music and multimedia performances
- Recording unique moments

Applications of off-the-shelf products
& development of proprietary solutions

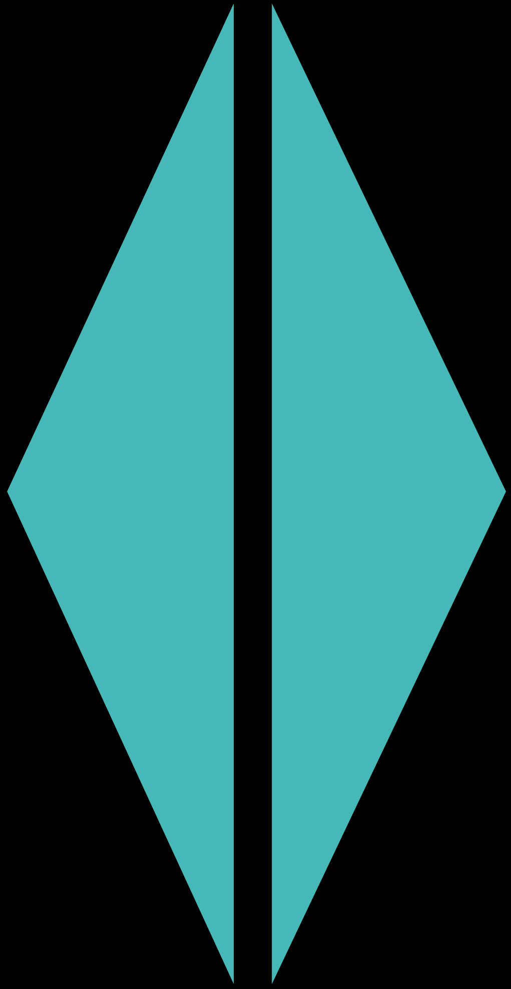
Tons of stuff to cover …

Preliminary structure

- Overarching strategy
- Hardware
  - Computers
  - Sensors
  - Audio interfaces
  - Connections
  - …
- Software
  - Operating systems
  - Parallel processing
  - User Interfaces
  - …

Preliminary structure

- Overarching strategy
- Hardware
  - Computers
  - Sensors
  - Audio interfaces
  - Connections
  - …
- Software
  - Operating systems
  - Parallel processing
  - User Interfaces
  - …



Art

Some topics to look into

Overarching strategy (1)

Look at reliability engineering:

- Fault prevention by design
- Fault detection and removal
- Fault tolerance
- Fault forecasting

Languages?
Systems?
Algorithms?

cf. M.R. Lyu (ed.), Handbook of Software Reliability Engineering

Overarching strategy (1)

Look at reliability engineering:

- Fault prevention by design
- Fault detection and removal
- Fault tolerance
- Fault forecasting

Art & science of testing (incl. instrumentation)

cf. M.R. Lyu (ed.), Handbook of Software Reliability Engineering

Overarching strategy (1)

Look at reliability engineering:

- Fault prevention by design
- Fault detection and removal
- Fault tolerance
- Fault forecasting

Bring three computers? Fallbacks in software?

cf. M.R. Lyu (ed.), Handbook of Software Reliability Engineering

Overarching strategy (1)

Look at reliability engineering:

- Fault prevention by design
- Fault detection and removal
- Fault tolerance
- Fault forecasting

Models for music software?

cf. M.R. Lyu (ed.), Handbook of Software Reliability Engineering

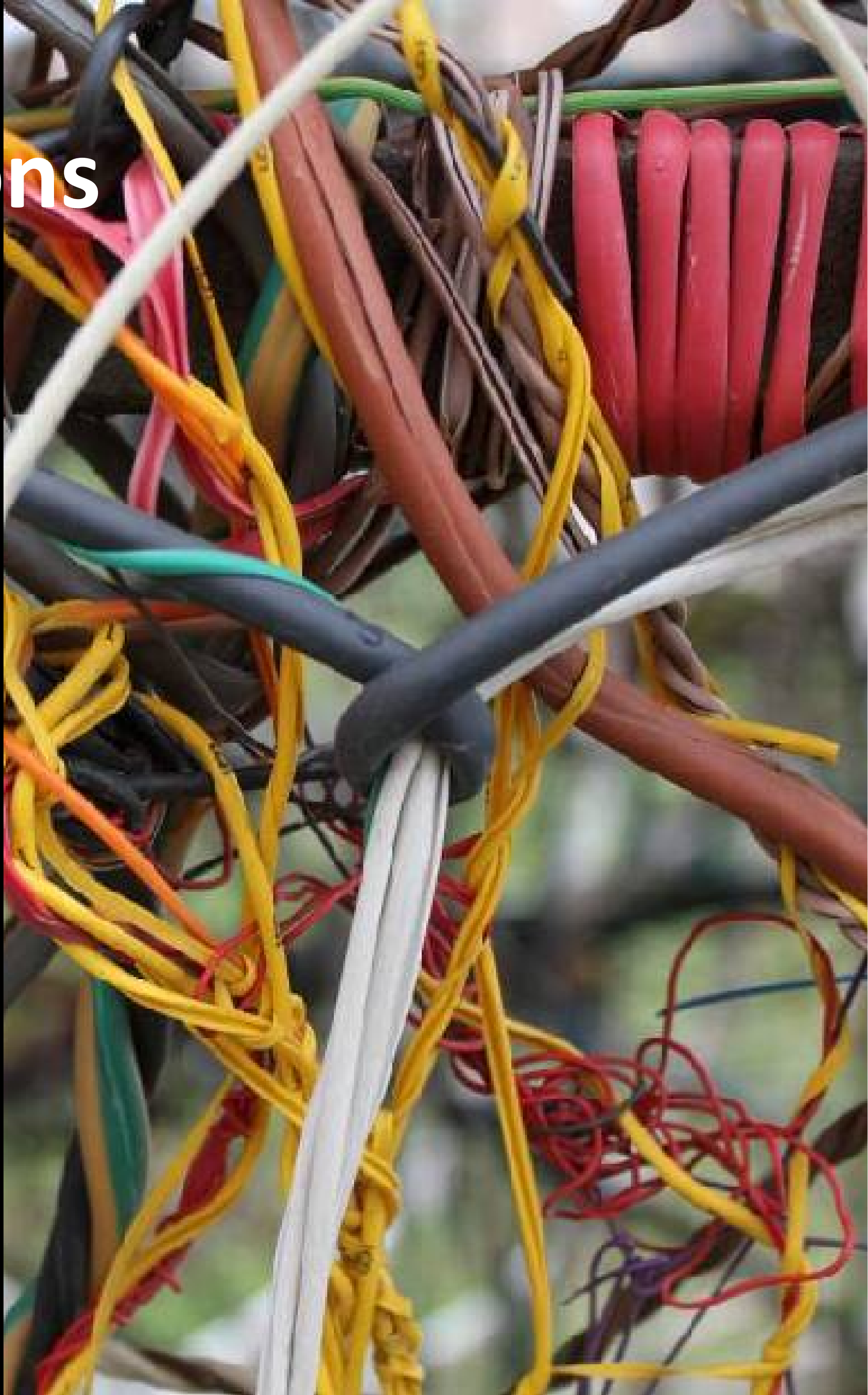
Overarching strategy (2)

Sophisticated or reliable: Pick one?

- a.k.a. “reduce to the max”
- Needs to be fleshed out in practice!
- Can we achieve a similar artistic effect with less technology?

Hardware: Connections

- Wires may break
  and may be unplugged
- Can you reinsert them?
  (noise, software)
- Wireless connections
  may be flaky
  or prone to intrusion

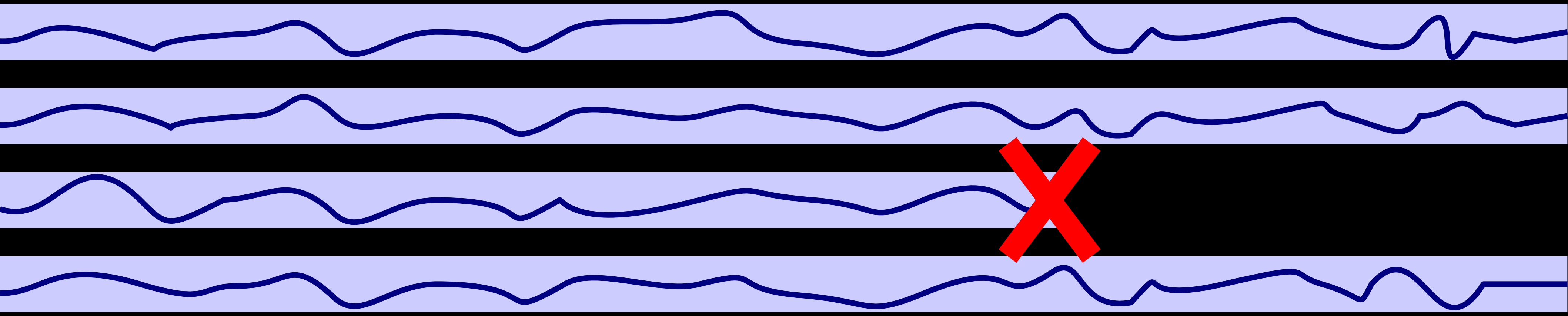


Software: operating system

- Task switching time?
- Memory management? (disk swapping; heap size)
- Background processes? (malware scanners, updaters, ...) Can you momentarily disable them?

Software: parallel processing

- How to debug multithreaded or multi-process applications?
- Parallel tracks as individual processes, so that no track can crash the others? (cf. Google Chrome)



Software: User Interfaces

- Foolproof interfaces?
  - No record button
  - Never overwrite files
  - Save automatically
- Indicate safe margins for recognition
  - Image
  - Audio
  - Tangible interfaces

Call to action

Let’s build a collection of:

- Tested methods
- Tips & tricks
- Pitfalls to be aware of
- Lessons painfully learned

And let’s come up with great new research!

Meet you at:

bit.ly/9kIvlT

Jörn Loviscach

www.j3L7h.de