THE SHOW MUST GO ON: DEPENDABLE AUDIO COMPUTING

Jörn Loviscach
Fachhochschule Bielefeld (University of Applied Sciences)
Fachbereich Ingenieurwissenschaften und Mathematik, Bielefeld, Germany
joern.loviscach@fh-bielefeld.de

ABSTRACT
Dependable computing has originated in domains such as aeronautics and medicine in which computer failure can lead to fatalities. No human lives are at stake in audio and music computing. However, a malfunction of today’s computerized music gear can easily ruin a live performance. This Unconference session looks into minimizing the risk of computer failure in unrecoverable moments of audio and music performances or recording sessions. The objective of this session is to share best practices as well as eclectic ideas, to inspire novel approaches, and to fuel future research.

1. INTRODUCTION
This session intends to assemble an audience of electronic musicians and computer scientists, ranging from novices to weathered experts. It starts off with a presentation of different aspects of reliability and approaches to it. Conventional and less conventional solutions as well as weird ideas will be introduced to spark contributions from the audience. Applying Unconference techniques, this session aims at building a Web-based repository, possibly in the form of a Wiki, preferably even before ICMC 2010. This repository stores general advice as well as specific tips and tricks—and experiences on what works and what does not.

The following sections illustrate topics each of which can form the basis for a breakout session, depending on the audience’s preferences. Some topics address overarching strategies, others address general but underrated technical issues. The aim is to not get lost in details but to spark fruitful discussions that extend long beyond the conference.

2. TOPICS: OVERARCHING STRATEGY
Sophisticated or reliable: Pick one? The simpler a system is, the easier it can be guarded against failure. To safely perform with a violin, you only need to bring an extra set of strings, just in case. Are there general principles—more concrete than “reduce to the max”—for striking the right balance between sophistication and reliability in electronic music performances?

Learn from rocket science? Airplanes can cross an ocean without a mid-flight reboot. Which of the development approaches employed to create such hardware and software can be carried over to audio and music? For instance: May a musician bring three computers onto stage, each doing the same task in parallel, but with a different version of some audio software on a different operating system?

Make it foolproof? To a large extent, computer failure results from user error. How to design user interfaces that an artist can handle in critical situations close to mental overload? This affects aspects such as graphics design and tangible controllers, but may also lead to rethink the technology behind the interface. For instance, a recording tool may record all of the time, so that one can’t forget to press a “Start” button. Human-computer interfaces based on image processing and/or pattern recognition incur a huge potential for disaster; this could be kept in check by deeper layers.

3. TOPICS: TECHNICAL ISSUES
Kill those daemons? On a standard computer, too much is going on. Malware protection may put a vital file into quarantine. The firewall may block a program that never before tried to fetch updates from the Internet. The wireless LAN card may disturb some critical interrupt. How can we learn what’s actually running on the machine? What is indispensable? How do we get rid of the potentially harmful items?

Run in parallel? Now that computers regularly come equipped with four or eight CPU cores, more and more developers have to tackle the intricacies of concurrency. Are there specific tools and/or specific guidelines to write multithreaded audio software? How can audio and music software be dissected into units with minimum dependencies?

Can we run, say, sequencer software as a collection of different processes so that one audio track may crash, but the others continue playing?

Dodge the wires? Inadvertently pulling a plug suffices to stop the show. Can you insert an audio plug again without devastating noise from the P. A.? Do you need to restart the audio software if you unplug a USB or Firewire audio device and then reconnect it? Should we go wireless? What would that mean in terms of latency and in terms of safety and security?