Auralization of Process and Port Status Using
Program Binaries to Generate Semantically
Meaningful Non-Fatiguing Noise as Canonical
Sound Signals

Ronald Loui, University of Illinois Springfield
Dylan Dozier**, Lincoln Land College
Prem Bhamidipati*,
Evan Barber** and Jeswanth Harish*, University of Illinois Springfield
r.p.loui@gmail.com

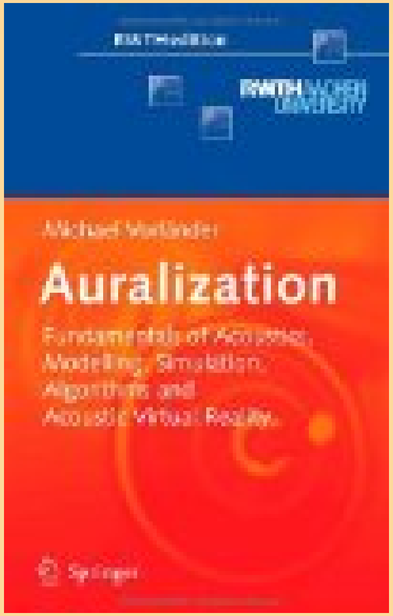
**undergraduates
*Msc students

IEEE EIT 2015 5/23/15
@NIU

UIS SPRINGFIELD CSC
R. LOUI

Auralization

- Using sound to report status
- Well used in some domains (medical monitoring, lab science, complex software)


RWTHedition
RWTH AACHEN UNIVERSITY
Michael Vorländer
Auralization
Fundamentals of Acoustics, Modelling, Simulation, Algorithms and Acoustic Virtual Reality
Springer

Sounds in Computing

- For alerts
  - Error bell
  - You've got mail
- For progress
  - Reboot beep
  - OS welcome
  - Shutdown farewell
- For data transmission
  - Acoustic modem / DSL

Why not for Monitoring?

- Computer rooms noisy
- Computer users like to play their music
- Tradition of visual monitors
  - http://uisacad5.uis.edu/cgi-bin/rloui2/ss.cgi
- Beeps are annoying
- It would require assigning sounds to events …
- Would help detect DDOS, malware, intruders, data loss, viruses, high loads ...

Windows 98 Sounds

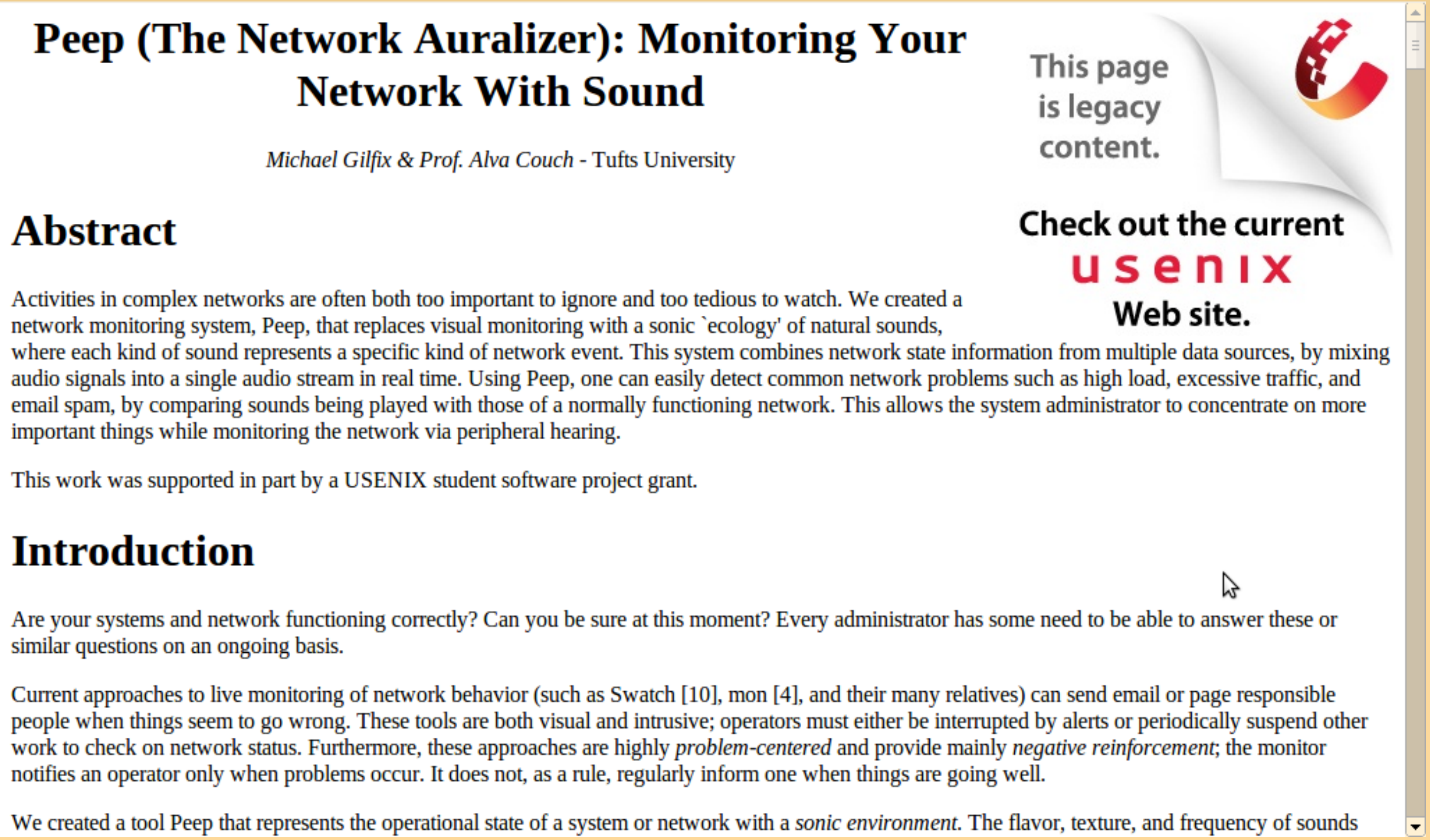
- Baseball
- Dangerous Creatures
- Inside Your Computer
- Jungle
- Leonardo Da Vinci
- Musica
- Mystery
- Nature
- Robotz
- Science
- Space
- Sports
- The 60's USA
- The Golden Era
- Travel
- Underwater
- Utopia
- Windows 98

Windows 7 Default Sounds

- chimes.wav
- chord.wav
- ding.wav
- ir_begin.wav
- ir_end.wav
- ir_inter.wav
- notify.wav
- recycle.wav
- ringout.wav
- Speech Off.wav
- Speech On.wav
- Speech Sleep.wav
- tada.wav

Prior Work in Computer Monitoring

- A student paper from 1999



Peep (The Network Auralizer): Monitoring Your Network With Sound

Michael Gilfix & Prof. Alva Couch - Tufts University

This page is legacy content.
Check out the current usenix Web site.

Abstract

Activities in complex networks are often both too important to ignore and too tedious to watch. We created a network monitoring system, Peep, that replaces visual monitoring with a sonic `ecology' of natural sounds, where each kind of sound represents a specific kind of network event. This system combines network state information from multiple data sources, by mixing audio signals into a single audio stream in real time. Using Peep, one can easily detect common network problems such as high load, excessive traffic, and email spam, by comparing sounds being played with those of a normally functioning network. This allows the system administrator to concentrate on more important things while monitoring the network via peripheral hearing.

This work was supported in part by a USENIX student software project grant.

Introduction

Are your systems and network functioning correctly? Can you be sure at this moment? Every administrator has some need to be able to answer these or similar questions on an ongoing basis.

Current approaches to live monitoring of network behavior (such as Swatch [10], mon [4], and their many relatives) can send email or page responsible people when things seem to go wrong. These tools are both visual and intrusive; operators must either be interrupted by alerts or periodically suspend other work to check on network status. Furthermore, these approaches are highly *problem-centered* and provide mainly *negative reinforcement*; the monitor notifies an operator only when problems occur. It does not, as a rule, regularly inform one when things are going well.

We created a tool Peep that represents the operational state of a system or network with a *sonic environment*. The flavor, texture, and frequency of sounds

Prior Work in Computer Monitoring

- Focused on event specification, not sounds
- Assumed sound clips such as beeps and chirps would work
- Many have had a musical background
- Asks sysadmins to be composers?
- Compare SPAMASSASSIN: would sysadmins have used it if they had had to write the regexps for detecting spam?

Major Advantages

- Visual attention is precious
- Audio can be processed passively
- Audio can overlay information
- Audio can report complex data efficiently
- Audio is excellent for habituation and anomaly detection
- *We just have to decide on the right sounds (and events)!*

Some Bad Ideas

- Use pure tones
  - Hard to habituate
- Count to convey measurements
  - Demands attention
- Use semantically meaningful sounds/music
  - Demands attention
  - Hard to habituate
  - http://uisacad5.uis.edu/~rloui2/out-1.ogv
  - http://uisacad5.uis.edu/~rloui2/out-2.ogv

Some Good Ideas

- Use the binaries themselves as sounds!
  - No disagreement over canonical assignment
  - Novel binaries have default assignments
  - Can detect corrupted or falsely named binaries (malware)
- Inject noise when doing counting
  - A warbled whistle
  - Report id-counts AFTER server status (so they can be ignored if status is not interesting)
  - http://uisacad5.uis.edu/~rloui2/ieee-eit.ogv

Conclusions

- If you want users to auralize, help them with sounds
- Monitoring/Alerting are two extremes, but monitors can be filtered and alerts can be regularized
- Noise can be better than ringtones
  - Less annoying
  - More informative
  - Appropriate and canonical

Future Work

- We would like to release this for gnu/linux and Win7/8 by end of Summer
- Questions?