



## Auditory Displays for Exploratory Programming

Christoph Thiede | [christoph.thiede@student.hpi.de](mailto:christoph.thiede@student.hpi.de)

2022-03-05

Making Things Audible | ACUD Berlin

## Software- Architekturen

Fachgebiet | Hasso-Plattner-Institut  
Universität Potsdam



## Squeak/ Smalltalk

## Neurodesign

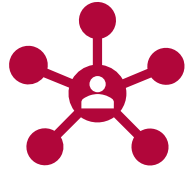
Forschungsgruppe | Hasso-Plattner-Institut  
Universität Potsdam



**Auditory Displays for  
Exploratory Programming**

Christoph Thiede  
2022-03-05

Chart 3



**save visual  
channel**



**background  
perception**



**attention**



**alternative  
perspectives**

**Auditory Displays for  
Exploratory Programming**

Christoph Thiede  
2022-03-05

Slide 4



## Related Work: Program auralization



**Auditory Displays for  
Exploratory Programming**

Christoph Thiede  
2022-03-05

Slide 6

## Related Work: Sound techniques



- Learnings from prior work:
  - don't replace visuals but augment them [Hussein et al., 2009; Vickers et al., 2006]
  - avoid fatigue/discomfort in the listener [Vickers, 2004, 2011]
  - use sound for characterization, not for identification [Vickers, 2011]
  - often helpful: playback/variable speed [Berman et al., 2017]



**speech** to describe information using **natural language**



**earcon**: “**abstract, synthetic tones** that can be used in **structured combinations** to create sound messages to represent parts of an interface.” [Brewster et al., 1993]



**auditory icon**: “**everyday sounds** that convey **information about events** in the computer or in remote environments by analogy” [Gaver, 1994; Gaver et al., 1995]

# Approach

- **a general-purpose toolkit to explore software systems through sonification**
- empower developers to **rapidly listen** to **aspects of interest**
  - code artifacts (packages, interfaces, methods, ...)
  - domain entities
  - conditionals
- define **custom parametrized sound mappings**
- further possible features:
  - sound aggregation
  - activation scope of triggers, debugger integration
  - sound watcher window (supports explorability)
  - scripted sound mappings
- advanced feature: sonify dynamic software metrics (EOC/IOC, LCOM, ...; Chhabra, 2010)

## No goals:

- complete/musical coverage (scalability)
- educational tool for beginners
- replace visual representations/build an accessibility product

## USPs:

- liveness (no precompilation/prerecording)
- reusable general-purpose solution
- trace individual entities
- configurability

**Auditory Displays for  
Exploratory Programming**

Christoph Thiede  
2022-03-05

Slide 8

## Sound-based tOols for uNderstanding of software sYstems through eXploration

### LinqLover/**sonyx**

A toolkit to explore software systems through sonification in Squeak/Smaltalk



1 Contributor 1 Issue 0 Stars 0 Forks



- a **general-purpose toolkit** to **explore software systems** through **sonification**
- empower developers to **rapidly listen** to **aspects of interest**
- define **custom parametrized sound mappings**



save visual  
channel



background  
perception



attention



alternative  
perspectives

Auditory Displays for  
Exploratory Programming

Christoph Thiede  
2022-03-05

Chart 9



< Demo >

```

at: anInteger
add example | add script example
0.2 seconds wait.
^ super at: anInteger
SonyxSound new
  squeakSound: #squeak;
  pitch: (anInteger from: 1 to: self size) into: 220 to: 880);
  synchronous: send
  balance: send
  duration: send
  pitch: send
  squeakSound: send
  volume: send
    
```

custom sound mappings

scriptable sound language

`anInteger from: 1 to: self size) into: 220 to: 880);`

**Babylonian Browser**

- Sandblocks-CustomJavas
- Sandblocks-Simulation
- Sandblocks-Plugin-StLive
- RuntimeValueVisualization
- Sandblocks-Ohm
- Sandblocks-Scheme
- Sandblocks-Scheme-Tests
- Unknown
- Sonyx-Study-Base
- Sonyx-Study-RVV
- Sonyx-StudyTasks-Sorting
- Sonyx-StudyTasks-Demo
- Sonyx-StudyTasks-Server
- Sonyx-StudyTasks-Regex
- Sonyx-Study-Demo
- Sonyx-Study-Solutions
- Sonyx-Study

instance class ?

browse senders implementors versions inheritance hierarchy vars source

**issues**

```

add example | add script example
| results |
results := ...
SonyxSound new
  squeakSound: #chirp;
  yourself
  chirp audicon
    
```

client httpGet: self baseUrl, /issue request contentType: self accept self authorizeRequest: request ifError: [:error | ^ error signal].

^ results collect: [:result | self parseIssue

ct 2/7/2022 11:28 · requests · 5 implemento

sound probe

`SonyxSound new`  
`squeakSound: #chirp;`  
`yourself`  
`chirp audicon`

sound monitor

**SonyxMonitor**

activate sonyx      fixup (in case of trouble)

Collection talkInsertionSort: (on)  
 RxMatcher matchesOnStream:do: (on)  
 SonyxDemoContext noteInstVarAccessTo:at: (on)  
 SonyxDemoMorph mouseDown: (on)  
**SonyxDemoMorph mouseMove: (on)**  
 SonyxDemoMorph mouseUp: (on)  
 SonyxSimpleDemo class griffle: (off)  
 SonyxSimpleDemo class griffle: (off)  
 SonyxSimpleDemo class griffleAll: (off)  
 SonyxSimpleDemo class nurb (on)  
 SonyxSimpleDemo class nurb (on)

browse    disable    remove    play demo

SonyxSound new  
 balance: 0;  
 squeakSound: PluckedSound default;  
 pitch: ((anEvent position y from: self top to: self bottom) into: 220 to: 440);  
 synchronous: false;  
 balance: (anEvent position x from: self left to: self right);  
 yourself

Auditory Displays for Exploratory Programming

Christoph Thiede  
2022-03-05

Chart 11

**typical programming tasks**      **operationalization**

**within-subjects randomization**

**participants (N = 6)**      **study script**

**control conditions**

**one-to-one sessions**

**recruitment**      **Zoom screen share**

**Auditory Displays for  
Exploratory Programming**

Christoph Thiede  
2022-03-05

Chart 12

# Evaluation: Operationalization

measure  
condition



task:

- sorting
- server
- regex

questionnaire: 5/6 participants would use a fully developed tool weekly

**Auditory Displays for Exploratory Programming**

Christoph Thiede  
2022-03-05

Chart 13

## Future Work

---

- Can programmers benefit from **monitoring state changes** via sound probes?
- How can we **aggregate probe sound events**?
- How can we make **listening to probe sound more interactive**?
- How can we improve the **convenience of the sound probe editor**?
- Can we improve the **understanding of software architectures** by **sonifying dynamic software metrics**?

# Summary

## Motivation



save visual channel



background perception



attention

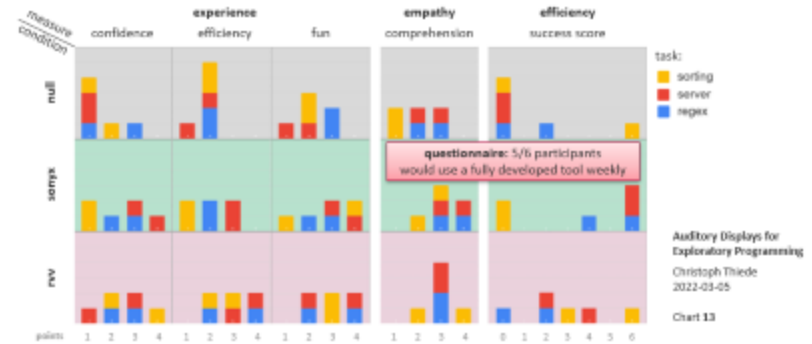


alternative perspectives

The screenshot shows the Sonyx IDE interface. A window titled 'scriptable sound language' displays a list of custom sound mappings. Another window titled 'sound probe' shows a configuration for a sound probe. The interface includes various panels and a terminal window at the bottom.

**GitHub:**  
<https://github.com/LinqLover/sonyx>

## Evaluation: Operationalization



Auditory Displays for Exploratory Programming  
 Christoph Thiede  
 2022-03-05  
 Chart 13

## Auditory Displays for Exploratory Programming

Christoph Thiede  
 2022-03-05

Chart 16

Auditory Displays for Exploratory Programming  
 Christoph Thiede  
 2022-03-05  
 Chart 11



## Auditory Displays for Exploratory Programming

Christoph Thiede | [christoph.thiede@student.hpi.de](mailto:christoph.thiede@student.hpi.de)

2022-03-05

Making Things Audible | ACUD Berlin