

# The Earconizer - A Tool for constructing hierarchical Earcons

Maximilian Hanke

Lars Engeln

Toni Auerswald

Felix Kallenbach

Rainer Groh

firstname.lastname@mailbox.tu-dresden.de  
Chair of MediaDesign, Technische Universität Dresden  
Dresden, Germany

## ABSTRACT

Earcons have become an important part of the human-computer interaction. In addition to the visual presentation of a user interface, the use of the auditory channel enriches software-side entities with information. Additionally, earcons can significantly improve interaction and effectively communicate complex information. Often Earcons are created with the expertise of a sound designer. How can the creation be simplified so that earcons can be created and organized by ordinary users? This paper shows of a tool for creating earcons based on the hierarchical structure of compound earcons to positively enhance the user experience in an interactive environment.

## CCS CONCEPTS

• **Human-centered computing** → **Auditory feedback**;  
*Sound-based input / output*; User interface design.

## KEYWORDS

Earcons, Auditory Displays, Sonification, Toolbox

## 1 INTRODUCTION

Interactive systems are primarily governed by visual stimuli. Accordingly, user interfaces are visually interactive: Elements are represented by icons and pictograms that can be touched, manipulated, moved, copied or cut out; they lie on a virtual desktop, are organized and edited in folder structures; processes are visualized by progress bars and percentage displays. The user receives direct visual feedback on his actions

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

*MuC'19 Workshops, Hamburg, Germany*

© Proceedings of the Mensch und Computer 2019 Workshop on Innovative Computer Music-Interfaces (ICMI). Copyright held by the owner/author(s).  
<https://doi.org/10.18420/muc2019-ws-436>

at all times. If this produces many visual stimuli close to each other, this can lead to a *cognitive overload*, which, among other things, impairs the user's decision-making speed. Due to the continuously increasing flood of information in everyday life, visual communication should be cognitively reduced (cf. [5]). Hereby, the auditory channel plays a crucial part, enriching system entities with information and communicating through the use of *auditory icons* [6] or *earcons* [1, 7].

Earcons are used to reduce information overload [2], and their general effectiveness has been shown [4]. In addition, Earcons can be designed in such a way, so that they can be perceived and identified at almost the same time [8]. Also, the use of earcons for hierarchies was discussed [3].

In the semiotics a subject is described with its *pragmatic*, *semantic* and *syntactic* function. Pragmatic describes the effect of a subject, i.e. how a user thinks and feels about a subject. The semantic describes the meaning of the subject and the syntax the general structure of the subject. Thereby, the syntactic structure influences the semantic meaning and the semantic meaning the pragmatic effect. The semantic is differentiated into the *symbolic*, *indexical* and *iconic* meaning. The semiotic classification of the semantic can be compared to the classification of earcons by Gaver [6] (see Table 1).

semiotics [9]	meaning	influence	earcons [6]
symbolic	assignment	society	symbolic
indexical	reference		metaphorical
iconic	identity	self / thing	nomic

**Table 1: Comparison of classification by semiotics and for the earcons.**

Three basic categories of earcons (*symbolic*, *metaphoric* and *nomic*), differing in mapping between visual information and auditory representation, are distinguished [6]. Compared to the field of semiotics [9] there are small differences.

*Symbolic* refers to earcons that are characterized by social conventions. For example, applause is associated with appreciation or acceptance.

*Metaphorical* earcons refer to a key characteristic (a indexical reference) of the thing to be represented. The earcon uses at least one dimension of the sound and changes it according to the meaning to be represented. For example, an ascending chord could represent the opening of an app and a symmetrically descending chord the closing of the app. The class of metaphorical earcons by Gaver [6] can not be directly transferred to the indexical meaning. In the semiotics the metaphoric meaning is more like a subclass of symbolic. Here the indexical references of atomic audio characteristic are creating the metaphoric reference to a sound.

*Nomic* earcons accentuate the (physical, iconic) source of the representing information. The sound of the earcon is taken from the reality, resulting in a direct association to the corresponding meaning in the user interface. In contrast to symbolic and metaphorical earcons, the learning process of nomic earcons is very short or even non-existent; in most cases, the association can be directly understood by the user. An example for this is emptying the trash bin on a desktop computer; the sound of crumpled paper represents the whole action and additionally the amount of emptied data.

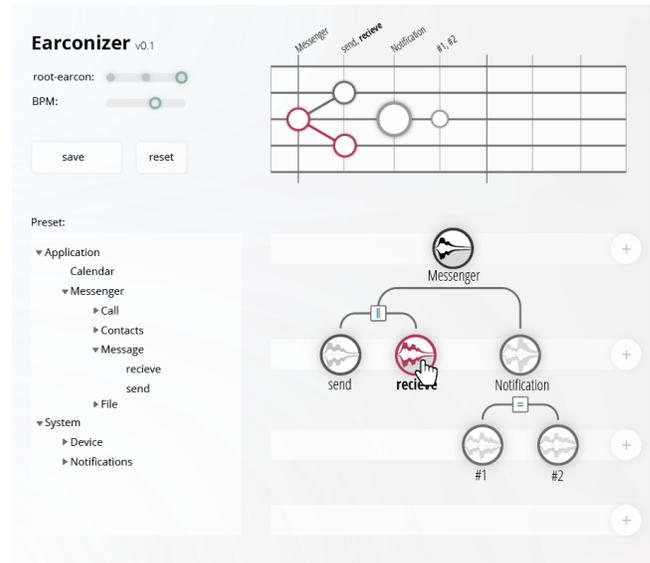
Meanwhile, verbal elements are also considered as a separate category of earcons [11]. For instance, these earcons, known as *spearcons*, can be used, to navigate in menu structures. It was shown that the use of spearcons results in an increase in decision performance and accuracy [10, 11].

To achieve a consistent sound design in an application, the Earconizer for ordinary users (non-experts in terms of sound design) is introduced here. The Earconizer relies on an semantic tree to construct compound earcons. Thereby, the focus is on metaphorical earcons. The input could also be nomic, but it is then treated as metaphoric. Therefore, the general construction of earcons is described first and afterwards the Earconizer is introduced.

## 2 EARCON CONSTRUCTION

The syntactical construction of an earcon is done either as a *one-element* earcon or as a *compound* earcon [1]. One-element earcons consist of a (synthesized) sound. This sound can be used individually as *single pitch* or in combination with others or itself as *single motifs* [1]. Single pitch earcons contain a maximum of one note with its pitch and is characterized by the timbre, length and intensity. The simple structure of single pitch earcons is reflected in their sound character, which is why these earcons should be used for correspondingly simple states and processes. Single motif earcons consist of the dimensions rhythm, pitch (melody), register, timbre and intensity. They are suitable for simple but relevant processes.

Compound earcons are divided into the three subtypes: *combined*, *inherited* and *transformed* [1].



**Figure 1: Interface of the Earconizer-App: general settings (upper left), compound earcon composer (upper right), tree-list with presets for templates and examples (lower left), earcon hierarchy manager (lower right)**

*Inherited* earcons follow a hierarchy in correspondence to their structure. Each Earcon acts as a node in a tree graph. The root represents the associated information class. Each hierarchical layer of earcons inherits the auditory structure of their parent-earcon. The received parent-earcon is combined with a new sound (note), resulting in the new parent-earcon for the next layer.

In contrast, *transformed* earcons do not insert new elements into the earcon itself, but modify existing properties of the sound over time.

The length of the earcon should be short, so that the information is presented in the shortest possible time and the easiest way to understand. An earcon should not last longer than 4 notes to give the user enough space for the recognition [1]. If this is respected, the learning effect is kept small in the interest of the user.

In order to achieve a meaningful representation of a larger hierarchy, the different types of compound earcons can be combined (like in [3]). This is why the Earconizer-App uses compound earcons.

## 3 THE EARCONIZER-APP

The Earconizer is a single-page web application based on Typescript. Vue.js is used for the component-based development of the application. Furthermore, p5.js is used for visualization and Tone.js for sound generation and modification. Being a webapp, everyone can use the Earconizer instantaneously.

The interface consists of a preset browser (see Figure 1, lower left) for an easy and fast start, a hierarchy manager (lower right) to create or modify the way the compound earcons are inherited (compare *inherited earcons*, Section 2), and an earcon composer (Figure 1, upper left) to influence the combination in terms of pitch and rhythm (compare *combined earcons*, Section 2). Both the hierarchy manager and the composer are modifying the syntactic structure and thereby the semantic meaning. The composer is more focused to the actual structure of one earcon, but may influence other earcons in their hierarchy. The hierarchy manager is more focused to the semantic of groups of earcons.

The Earconizer starts with a root-earcon, which is combined and/or transformed to a new earcon, which is then inherited to the next lower layer. In this early version, the root-earcon can be chosen from three predefined samples. The rhythm speed can be adjusted by a BPM slider.

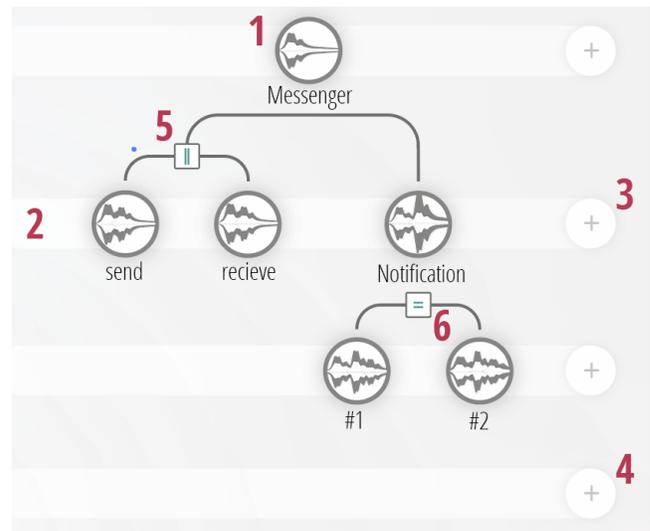
### Preset Browser

The Earconizer provides various presets for the user (see Figure 1, lower left). Depending on the application to be configured, the user can select a suitable template, such as a preset for earcons of a typical messenger. The main component of a preset is a tree graph, whose internal node logic represents the semantics and functions of the application. Therefore, two nodes can be defined as *symmetrical* or *similar*. An action or a subtree from the browser can be dragged to the hierarchy manager.

### Hierarchy Manager

The hierarchy manager is a tree. Thereby, the user gets a comprehensive overview of the composed earcons and their inheritance. Earcons at the same layer can differ in melody (pitch), rhythm (dynamic), and/or transformation (effects, i.e. reverberation, distortion, etc.). The next lower layer differs in the amount of used notes (pitched and rhythmized combination of the root-earcon) or a transformation. The tree also contains semantic features for symmetric earcons (|| – only modified in pitch, e.g. ascending and descending melody for receive/send message) or earcons that should be very similar (= – only transformed by an effect). Those features are created by dragging and dropping an earcon representation on another (see Figure 2). Rhythm, pitch and transformation is inherited to the next layer.

Internally the root node of the graph stores the root-earcon and is then passed on to the other nodes for the sound generation. According to the (semantically, auditive) transformation and modulation stored in the nodes, the root-earcon is manipulated node by node and then saved, as well as visualized in the circle representation.



**Figure 2: Hierarchy manager: root-earcon (1), layer of earcons with the same parent (2), add earcon to layer (3), add earcon to new layer (4), earcons are grouped as symmetric (5), earcons are grouped as similar (6)**

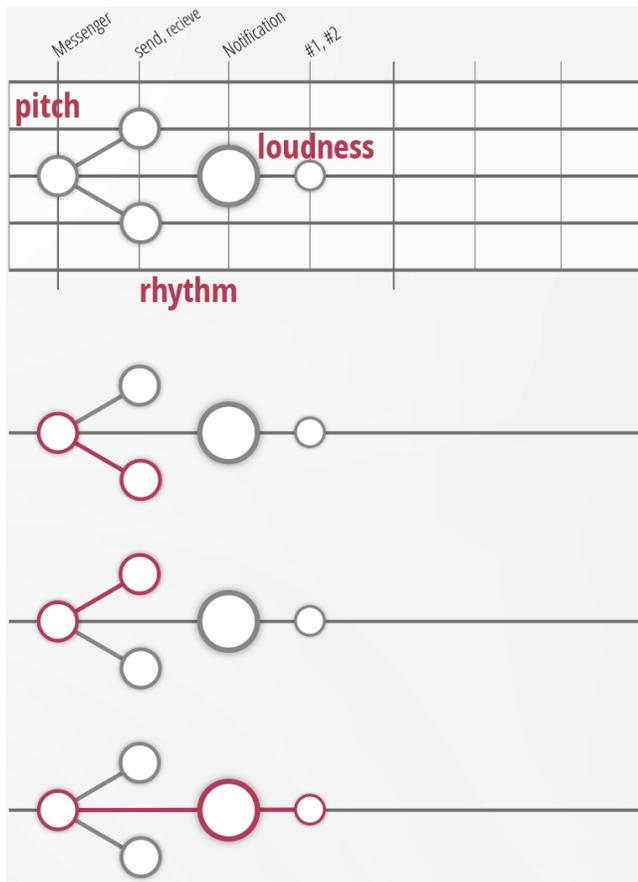
### Earcon Composer

Each sound can be formally described and distinguished from other sounds by looking at different features. The most important features are rhythm, timbre, pitch, intensity and length. The rhythm is the most striking parameter of an earcon. Changing the rhythm (and maintaining other dimensions) is one of the simplest ways to create two significantly distinguishable patterns. The composer visualizes and combines these dimensional features as circles on a time based grid to provide an overview of the total template. At this point the pitch, rhythm and loudness of each earcon (and its components) can be changed (see Figure 3). Hovering above a representation of an earcon in the hierarchy or in the composer the corresponding representations along the inheritance is highlighting the iterative construction (see Figure 1). A change in the composer leads to a regeneration of the earcons in the according subgraph.

## 4 FUTURE WORKS

The root-earcon should be exchangeable by the user, so that here the possibility of an upload will be given. A self-uploaded root-earcon allows personalization and in particular the adaptation to an existing corporate design (sound design). Furthermore, a built-in synthesizer would be able to create suitable root samples.

An expandable system for audio effects shall be added to get more expressiveness of transformed earcons. Thereby, the earcon composer has to be extended to handle various effects like reverb, distortion, panning and filtering. A visual



**Figure 3: Earcon composer with quantized options for pitch and rhythm (musical staff metaphor) for the earcon notes represented as circles. The composer view yields multiple compound earcons (see below the staff), so that the inheritance and relationships are visible at once.**

parameterization of the effects is appreciable, so that the effects can be influenced by the circles shape within the composer itself.

## 5 CONCLUSION

The Earconizer contains all elements that are necessary to sonify a system with earcons. In this way, an opportunity to fundamentally simplify processes in the creation, organization and structuring of earcons is given. The use of the earcon composer and the hierarchy graph, gives the user many possibilities to customize the sound without unintentionally creating misleading earcons. Developers can create earcons without a sound designer's profound knowledge of semantic and syntactic, and keep track of the used earcons. With the Earconizer-App a proof-of-concept is achieved, which should be evaluated in terms of usability and effectiveness.

## REFERENCES

- [1] Meera M Blattner, Denise A Sumikawa, and Robert M Greenberg. 1989. Earcons and icons: Their structure and common design principles. *Human-Computer Interaction* 4, 1 (1989), 11–44.
- [2] Stephen A Brewster. 1997. Using non-speech sound to overcome information overload. *Displays* 17, 3-4 (1997), 179–189.
- [3] Stephan A Brewster, Adrian Capriotti, and CV Hall. 1998. Using compound earcons to represent hierarchies. *HCI Letters* 1, 1 (1998), 6–8.
- [4] Stephen A Brewster, Peter C Wright, and Alastair D N Edwards. 1994. A detailed investigation into the effectiveness of earcons. In *SANTA FE INSTITUTE STUDIES IN THE SCIENCES OF COMPLEXITY-PROCEEDINGS VOLUME-*, Vol. 18. ADDISON-WESLEY PUBLISHING CO, 471.
- [5] Franz-Rudolf Esch, Elisabeth von Einem, and Sabrina Eichenauer. 2016. Visuelle Reize in der Kommunikation. *Handbuch Sozialtechniken der Kommunikation: Grundlagen-Innovative Ansätze-Praktische Umsetzungen* (2016), 1–22.
- [6] William W Gaver. 1986. Auditory icons: Using sound in computer interfaces. *Human-computer interaction* 2, 2 (1986), 167–177.
- [7] Thomas Hermann, Andy Hunt, and John G Neuhoff. 2011. *The sonification handbook*. Logos Verlag Berlin, Germany.
- [8] David K McGoogin and Stephen A Brewster. 2004. Understanding concurrent earcons: Applying auditory scene analysis principles to concurrent earcon recognition. *ACM Transactions on Applied Perception (TAP)* 1, 2 (2004), 130–155.
- [9] Eugene Rochberg-Halton and Kevin McMurtrey. 2007. The foundations of modern semiotic: Charles Peirce and Charles Morris. *The American Journal of Semiotics* 2, 1/2 (2007), 129–156.
- [10] Bruce N Walker, Jeffrey Lindsay, Amanda Nance, Yoko Nakano, Dianne K Palladino, Tilman Dingler, and Myoungsoon Jeon. 2013. Spearcons (speech-based earcons) improve navigation performance in advanced auditory menus. *Human Factors* 55, 1 (2013), 157–182.
- [11] Bruce N Walker, Amanda Nance, and Jeffrey Lindsay. 2006. Spearcons: Speech-based earcons improve navigation performance in auditory menus. Georgia Institute of Technology.