UNIVERSITY OF CRETE
DEPARTMENT OF COMPUTER SCIENCE
FACULTY OF SCIENCES AND ENGINEERING

ACOUSMA: An intelligent mechanism towards providing personalized auditory feedback in Ambient Intelligence Environments

by

Andreas Michelakis

MSc Dissertation
Presented
in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

Heraklion, May 2020

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ
ΤΜΗΜΑ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΚΑΙ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΑΚΟΥΣΜΑ: Ένας ευφυής μηχανισμός παροχής εξατομικευμένης ηχητικής ανάδρασης σε Περιβάλλοντα Διάχυτης Νοημοσύνης

από

Ανδρέας Μιχελάκης

Εκπόνηση

διατριβής

ως μέρος εκπλήρωσης

των απαιτήσεων

για την απόκτηση του

Μεταπτυχιακού Διπλώματος Ειδίκευσης

Ηράκλειο, Μάιος 2020

UNIVERSITY OF CRETE
DEPARTMENT OF COMPUTER SCIENCE

ACOUSMA: An intelligent mechanism towards providing personalised auditory feedback in Ambient Intelligence Environments

MSc Dissertation Presented
by **Andreas Michelakis**
in Partial Fulfillment of the Requirements
for the Master of Science

APPROVED BY:

Author: Andreas Michelakis, University of Crete

CONSTANTINOS STEPHANIDIS
Digitally signed by CONSTANTINOS STEPHANIDIS
Date: 2020.05.27 16:20:34 +03'00'

Supervisor: Constantine Stephanidis, Professor, University of Crete

Margherita Giovanna Antona
Digitally signed by Margherita Giovanna Antona
Date: 2020.05.27 17:31:59 +03'00'

Committee Member: Margherita Antona, Principal Researcher, ICS-FORTH

GEORGIOS PAPAGIANNAKIS
Digitally signed by GEORGIOS PAPAGIANNAKIS
Date: 2020.05.27 18:14:14 +03'00'

Committee Member: George Papagiannakis, Associate Professor, University of Crete

Digitally signed by ANTONIOS ARGYROS
Date: 2020.05.27 18:54:46 +03'00'

Director of Graduate Studies: Antonis Argyros, Professor, University of Crete

Heraklion, May 2020

Acknowledgments

I would like to thank my Thesis supervisor, Professor Constantine Stephanidis for providing the necessary means to inspire, design, develop and complete this dissertation. I would also like to thank all staff and colleagues at the Human Computer Interaction Laboratory for their continuous support and genuine interest throughout this Thesis development.

I thank Emmanouil Zidianakis for his invaluable support, guidance and direction throughout all stages of the current work. For his great support at the first steps of this work, I would like to thank Asterios Leonidis. For their comments in shaping the text of this thesis, I am grateful to Margherita Antona and Emmanouil Zidianakis. For her efforts in making the current usability evaluation possible, I thank Stavroula Ntoa.

Moreover, I would like to thank Professor Maria Papadopouli for her introduction to research methodology & recommendation systems and Professor Nikos Bubaris for his introduction to sound studies, both of which acted as inspiration and initial guide to investigating the use of sound in HCI. Of course, I am grateful to all of my Professors at the department of Cultural Technology and Communication at the University of Aegean for supporting me in practice towards post-graduate studies and providing me with a broad spectrum of knowledge during my undergraduate studies.

Lastly and most importantly, I am eternally grateful to my family and parents. Their care and support has made everything in my life, possible.

Abstract

Ambient Intelligence (AmI) environments are becoming prominent in everyday living as technological advancements are setting the reality of the 'Internet of Things' (IoT). Physical spaces with intelligent and intuitive interfaces embedded in everyday objects respond to the presence of people in a seamless and unobtrusive fashion. Yet, the majority of applications and research for AmI environments has set the focus on communicating information using the visual channel. Screens of various sizes (smartphones, tablets, projectors) are used to display information utilizing the already exhausted visual channel. The focus on the visual channel comes in partial contrast to the inconspicuous nature of AmI environments interfaces. In order to truly offer ubiquitous and multimodal interfaces to provide information, the auditory channel should be considered as well.

Driven by the above considerations, ACOUSMA, a platform based on a micro-services architecture, was developed to allow effortless enhancement of AmI environments with personalized auditory displays. ACOUSMA consists of two system packages designed for interaction with three groups of users: a) auditory display experts, b) AmI application developers, and c) AmI environment end-users. Auditory display experts can generate, upload and share their designed auditory representations using an intuitive Content Management System. AmI application developers with no prior knowledge on the field can integrate and direct personalized, prompt auditory displays to any AmI environment user with minimum effort using a comprehensible application programming interface and an intelligent mechanism module. This intelligent module adopts and adapts to user preferences in real time, enabling conversion of any given semantic information into meaningful user-tailored auditory displays. Finally, AmI environment users can experience auditory displays that occur in the favorable place in a timely manner and can be adjusted according to their preferences.

Usability evaluation from both experts and real users has shown that target user groups can benefit from using ACOUSMA as a platform to store and share auditory representations, as well as to provide personalized, meaningful and according to context auditory displays. Moreover, users could successfully and efficiently communicate their needs and adjust auditory displays using the intelligent mechanism module.

Περίληψη

Τα περιβάλλοντα Διάχυτης Νοημοσύνης (ΔΝ) έχουν καταστεί εμφανή στην καθημερινή ζωή, καθώς οι τεχνολογικές εξελίξεις οδηγούν στην πραγμάτωση του 'Διαδικτύου των Πραγμάτων' (Internet of Things - IoT). Χώροι με έξυπνες και διαισθητικές διεπαφές, ενσωματωμένες σε καθημερινά αντικείμενα, ανταποκρίνονται στην παρουσία των ανθρώπων με απρόσκοπτο και διακριτικό τρόπο. Ωστόσο, η ερευνητικές προσπάθειες και η πλειονότητα των εφαρμογών σχετικά με τα περιβάλλοντα ΔΝ έχει θέσει ως επίκεντρο τη μετάδοση πληροφοριών χρησιμοποιώντας το οπτικό κανάλι επικοινωνίας. Οθόνες διαφόρων μεγεθών (έξυπνα κινητά, ταμπλέτες, βιντεοπροβολείς) χρησιμοποιούνται για την προβολή πληροφοριών μέσω του ήδη υπερφορτωμένου οπτικού καναλιού. Η εστίαση στη χρήση του οπτικού καναλιού δεν είναι συμβατή με την 'αφανή' φύση των διεπαφών σε περιβάλλοντα ΔΝ. Προκειμένου να προσφέρονται πραγματικά αφανείς, πανταχού παρούσες και πολυτροπικές διεπαφές για την μετάδοση πλούσιας πληροφορίας, είναι απαραίτητο να ληφθεί υπόψιν το ακουστικό κανάλι.

Με γνώμονα τα παραπάνω, αναπτύχθηκε το ΑΚΟΥΣΜΑ, μια πλατφόρμα βασισμένη στην αρχιτεκτονική μικρό-υπηρεσιών, για να επιτρέψει την εύκολη ενίσχυση των περιβαλλόντων ΔΝ με εξατομικευμένες ακουστικές προβολές, πλούσιες σε πληροφορία. Το ΑΚΟΥΣΜΑ αποτελείται από δύο πακέτα συστημάτων σχεδιασμένα για χρήση από τρεις ομάδες χρηστών: α) εμπειρογνώμονες ακουστικών προβολών, β) προγραμματιστές εφαρμογών ΔΝ και γ) τελικοί χρήστες περιβαλλόντων ΔΝ. Οι εμπειρογνώμονες ακουστικών προβολών μπορούν να δημιουργούν, να μεταφορτώνουν και να μοιράζονται ακουστικές αναπαραστάσεις χρησιμοποιώντας ένα διαισθητικό σύστημα διαχείρισης περιεχομένου. Οι προγραμματιστές εφαρμογών ΔΝ, χωρίς προηγούμενη γνώση του τομέα των ακουστικών προβολών, μπορούν να ενσωματώσουν και να κατευθύνουν άμεσες και εξατομικευμένες ακουστικές προβολές σε οποιονδήποτε χρήστη του περιβάλλοντος ΔΝ με ελάχιστη προσπάθεια, χρησιμοποιώντας μια ευκόλως κατανοητή διεπαφή προγραμματισμού εφαρμογών και ένα ενσωματωμένο ευφυή μηχανισμό. Η ευφυής αυτή μονάδα υιοθετεί και προσαρμόζεται σύμφωνα με τις προτιμήσεις των χρηστών σε πραγματικό χρόνο, επιτρέποντας τη μετατροπή οποιασδήποτε σημασιολογικής πληροφορίας σε μια ευκόλως κατανοητή ακουστική προβολή προσαρμοσμένη στις προτιμήσεις του χρήστη. Τέλος, οι χρήστες περιβαλλόντων ΔΝ βιώνουν ακουστικές προβολές που είναι ευπροσάρμοστες στις προτιμήσεις τους και τους παρέχονται έγκαιρα και ευνοϊκά.

Η εκπόνηση μιας αξιολόγησης ευχρηστίας τόσο με εμπειρογνώμονες όσο και με πραγματικούς χρήστες κατέδειξε ότι οι ομάδες χρηστών στις οποίες αναφέρεται το ΑΚΟΥΣΜΑ μπορούν να επωφεληθούν από τη χρήση του ως μια πλατφόρμα για αποθήκευση και ανταλλαγή ακουστικών αναπαραστάσεων καθώς και για την παροχή εξατομικευμένων ακουστικών προβολών. Επιπλέον, οι χρήστες μπόρεσαν με επιτυχία και αποτελεσματικά να επικοινωνήσουν τις ανάγκες τους και να προσαρμόσουν τις ακουστικές οθόνες σε πραγματικό χρόνο χρησιμοποιώντας τον παρεχόμενο ευφυή μηχανισμό.

Contents

List of Figures

List of Tables

List of Equations

Chapter 1
1. Introduction

The concepts of Ambient Intelligence [1] [2] (AmI) and Pervasive Computing [3] have revolutionized the traditional interaction paradigm from humans being merely operators of stationary machines to them being surrounded by ubiquitous, technology-enriched artifacts (e.g. smart devices, sensors, actuators) and computational units that intelligently react to user and environment context. In AmI environments, such artifacts and computational units, collaborate and interact in order to achieve the shared goal of improving the quality of life of humans, satisfying their needs and assisting them in daily activities [1] [4].

The vast number of application domains for intelligent environments is only limited by the number of tasks that can be automated to improve inhabitants' quality of life or the necessity for assistance with daily activities and satisfaction of requirements for groups of people such as the elderly or people with disabilities [4].

A common feature of any intelligent environment application is the multimodal method by which users may interact with the environment and application. Users may use touch screens, manipulate artifact properties, use speech or even movement to interact with an intelligent environment. Even though a multimodal technique for interaction is favored, it is easily noticeable that the focus, for the majority of applications, is set on the visual channel to display information. Therefore, more often than never, the auditory channel is neglected or misused.

1.1 Problem Statement

The focus on visual displays results in two significant, correlated issues: (a) the auditory channel is not efficiently utilized by AmI applications and (b) developers, lacking the specific domain knowledge and tools to provide meaningful, personalized auditory displays, misuse it (see 3.1 User Groups & Requirements). The work of [5], emphasizes the requirement to efficiently use the auditory channel to transmit meaningful information for both sighted and visually impaired users in AmI environments. To support that requirement and resolve issues (a) & (b), two complementary challenges must be overcome. The first challenge is for intelligent environment applications to efficiently utilize (along others) the auditory channel and AmI environment infrastructure to display information, and the second is for developers to be able to effortlessly create and provide meaningful, personalized auditory displays.

In order to grasp the emergence of these two challenges, one must first look into the fundamentals of the AmI paradigm. The next section provides definitions and characteristic of the AmI paradigm. Those will be used as guidelines for ACOUSMA's contributions and objectives as well as its design.

1.2 AmI Paradigm

1.2.1 Paradigm Overview

The rapid development of computer science and its continuous growth have made microcomputers capable of holding exceptional processing power while remaining small, able to be integrated into everyday objects [6]. This advancement has led to an introduction of several concepts in computer science, one of them being Ambient Intelligence (AmI). AmI has steered designs to embed computing power in daily used objects like home appliances (e.g. kitchenware [7]) and enrich everyday actions such is knocking on a door [8] with rich informatory feedback. This has come to compliment the shift in the ratio of computing power per person as seen in [3] and Figure 1. In the early years of computers, a single unit would usually be utilized by many users as it was expensive to manufacture and occupied a lot of space (computers were the size of a room). As technology advanced,

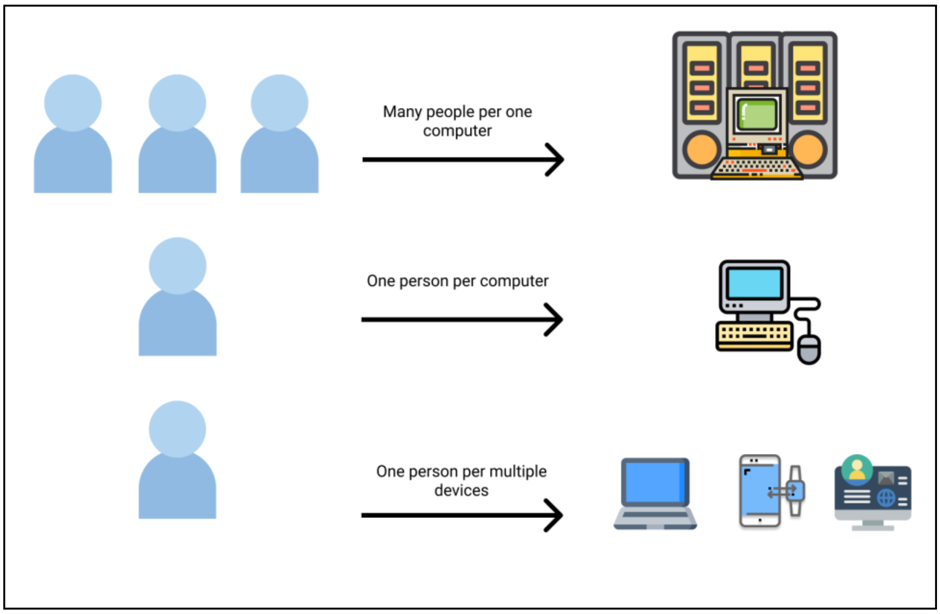


Figure 1 - Computing power per user over time (as inspired by [3])

computers decreased in size and manufacturing and market costs dropped allowing people to own a personal computer. Nowadays, a typical user owns multiple devices, all capable of complicated computing tasks [3].

The combination of microprocessors decreasing in size, the high computing power to user ratio and the successful user experience with mobile devices and technological artifacts has set the foundation to realize AmI [3]. AmI refers to electronic environments that sense and respond to the presence of humans, combining multidisciplinary fields (see Figure 2) and setting new

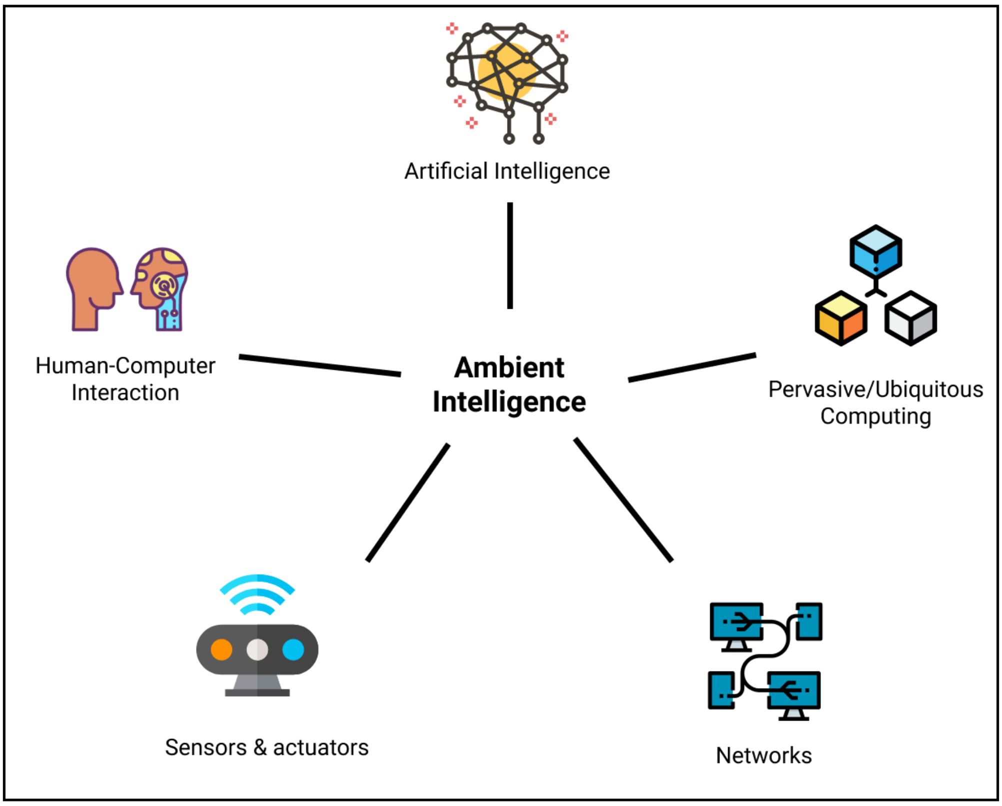


Figure 2 - Multidisciplinary nature of AmI (as inspired by [5])

concepts to achieve that [9]. AmI environments are implemented with technology that operates collectively, using contextual information and intelligent services that are hidden in connected devices (Ubiquitous Computing).

This along with natural and intuitive user interfaces outline the AmI user experience [3]. Common artifacts are used to integrate technology in an unobtrusive manner that makes use of distributed information and intelligence emerging from interconnected services and systems. Services and sensors can record and provide valuable information for an AmI environment, such as temperature and lighting conditions or a resident's vital signs (e.g. heart rate) in real time. Management of that information using specialized algorithms so that the environment can profile users, act and react to their needs is established as the environment's intelligence.

1.2.2 Definition and characteristics

The term 'Ambient Intelligence' exists in literature since 2001 when it was used by European Commission [10] to refer to the future of digital systems in the period 2010 – 2020 with users and devices interacting through interfaces in real-time and pro-actively [11]. Despite the widespread use of the term, a formal definition of AmI does not exist. Although, many researchers have attempted to define AmI:

- Cook et al. [3] report it as an emerging discipline that brings intelligence to our everyday environments and makes them sensitive to people.
- Aarts et al. [9] describe it as "A developing technology that will increasingly make our everyday environment sensitive and responsive to our presence.".
- The European Commission's IST Advisory Group in 2001 [10] describes AmI as: "A potential future in which we will be surrounded by intelligent objects and in which the environment will recognize the presence of persons and will respond to it in an undetectable manner".
- Dohsaka et al. [12] in the January 2006 issue of NTT technical review stated that Ambient Intelligence implies intelligence that is all around us.
- Ramos et al. [13] state that AmI deals with ubiquitous computing devices, allowing users to interact with their environment in an intelligent and unobtrusive way.
- Quoted from [14] : "Ambient Intelligence (AmI) is a new research area for distributed, non-intrusive, and intelligent software systems both from the direction of how to build these systems as well as how to design the collaboration between ambient systems".

- Vasilakos and Pedrycz in [15], mention that "In an AmI environment people are surrounded with networks of embedded intelligent devices that can sense their state, anticipate, and perhaps adapt to their needs".

Additional definitions exist in literature, but, regardless of apparent differences, an equal idea is shared amongst researchers. An AmI environment is characterized by five primary features [9] [16]:

1. **Embeddable**
   It is built upon technological advancements of computer networks, sensors, actuators and distributed computing.
2. **Context-aware**
   Utilizes contextual information regarding users, creating profiles, recognizing user activities, situation and location.
3. **Personalizable**
   It is customizable according to user preferences.
4. **Adaptive**
   It adapts dynamically to change in user preferences and context.
5. **Anticipatory**

   Uses artificial intelligence to predict user intentions.

Those characteristics are used as aid in setting design and development standpoints for AmI environment applications. ACOUSMA, being an application for AmI environments, embodies all of those features while shifting the focus towards the use of auditory feedback.

1.3 Contributions

Two challenges were indicated in the first section of this chapter. That (a) AmI applications need to efficiently utilize the auditory channel as well as the AmI environment infrastructure to display information and that (b) AmI application developers must be able to effortlessly create and provide meaningful, personalized auditory displays. Those challenges align with the core characteristics of an AmI environment to be embeddable, context-aware, personalizable, adaptive and anticipatory. As a solution to both challenges, this Thesis introduces ACOUSMA, a complete framework and suite of tools designed to aid developers and auditory display experts in enhancing AmI environment applications with meaningful auditory displays that are personalized to inhabitant's preferences and adapt according to environment and inhabitant context.

Specifically, the system aims to contribute by providing:

1. The necessary functions and interfaces to allow easy storage and management of auditory displays.
2. An intelligent mechanism to evaluate environment and user context and prioritize auditory display playback accordingly.
3. A method to control and deploy auditory displays anywhere within any AmI environment, taking advantage its available infrastructure and resources.
4. Personalization of auditory displays. Displays that are tailored to user profile, can be adjusted to user preferences and needs at any time.

In addition to the above, ACOUSMA will provide interfaces to allow any AmI environment application, existing or new, to:

5. Efficiently request a meaningful auditory display towards any user or location within an AmI environment
6. Integrate methods that will allow users to comment on displays via application-defined modalities. Therefore, allowing users to adjust provided auditory displays to their preferences.

Those contributions can be mapped to specific objectives to be met by ACOUSMA. Forthcoming chapters formulate and specify those objectives. The next chapter regarding background and related work examines the state-of-the-art in auditory display literature and provides a useful perspective on how other applications provide auditory feedback in the AmI context. Based on that perspective, specific objectives are framed. Chapter 3 analyzes how user requirements were elicited and how ACOUSMA was designed upon them. Chapter 4 regards the implementation of ACOUSMA and chapter 5 the evaluation of that implementation. Evaluation results have positively indicated that all user requirements and objectives were met.

Chapter 2

2. Background & Related Work

In this chapter, background and the current state-of-the-art in auditory representations and auditory displays will be presented along with applications in various domains including intelligent environments. In addition to that, related work for ACOUSMA is going to be examined.

2.1 Focusing on auditory feedback

Numerous benefits can derive from providing auditory feedback in an AmI environment. The basis of these benefits lies on the physiology [17] [18] [19] and psychology [20] [21] of hearing and the human perception of sound [22] [23]. In the following paragraphs, research results supporting those benefits will be presented.

Research has indicated that the human brain deeply tethers the relationship between an action and the resulting sound. Neural and behavioral aspects of this tethering should be focused upon, when considering the benefits of providing auditory feedback in an AmI environment. The human ear can discriminate fine temporal events as are amplitude and frequency modulations (chapter 5 of [20]). Furthermore, there is strong evidence that processing time of auditory information can be shorter than visual [17]. The human sensory system can process auditory information without interfering with visual feedback processes [24], making multi-sensory interaction with the environment possible. The latter suggests that using the auditory channel in an AmI application (or any other) for providing rich information can be done regardless of what is provided on the visual channel. While the visual channel allows for creating previsions of approaching events and detection of changes in any dynamic environment, the use of the auditory channel and its combination with the visual can enhance perception and learning [17] [18].

The ability of the human brain to receive, separately or in combination, information from the auditory and visual channels, derives from a class of visual/auditory - motor neurons called mirror neurons, discovered in the mid 90's. The neurons only fire when there is an object present or an action being performed (visually or auditory) [25]. The same neurons code the meaning of actions whether the said action is heard, performed or seen [19]. The mechanism of the mirror system allows for the internal representation of an auditory object when hearing the sound of an action linking it to source and context [22] [23]. In a well cited study of ecological psychology, Gaver [21] introduced the term of everyday listening, which focuses on how humans experience everyday hearing events in their surrounding environments. Gaver describes sound producing events as connected to a specific source (e.g. the sound of a car's engine approaching) and analyses how sound is perceived (Figure 3) by the human auditory system having distinct properties as is the material that the sound source is made of (e.g. metallic). Humans are able to finely distinguish different everyday sounds and connect them to meaningful information such as a person running upstairs or downstairs.

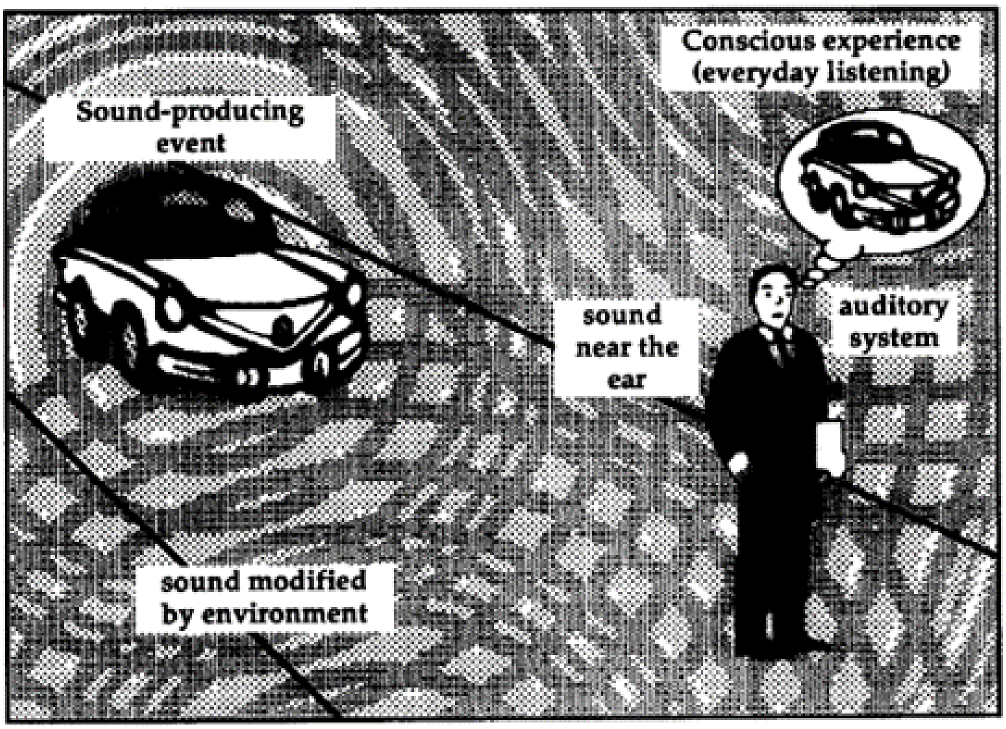


Figure 3 - Perception of sound (from [21])

In recent decades, along with great advances in computing environments and human-machine interfaces, a considerable amount of research projects and applications have set the focus on the use of audio to represent events emerging from these environments and interfaces. The field that addresses these issues is the field of Auditory Display, which is a part of Human Computer Interaction (HCI). According to Kramer et al. [26], the field of Auditory display is examined in accordance to achievements, research questions and potentiality of using audio to transmit information. According to that study, there exist several applications and tasks where displaying information through an auditory display would be particularly advantageous:

- In monitor tasks, usually eyes are busy and an eye-free interface is useful to have; e.g. driving vehicle, cockpit operations, network monitoring, factory floors.
- Rapid detection of event in high stress environments, where immediate reaction is essential.

- Viewing and analyzing large data sets. The auditory system has the ability of backgrounding. A person can listen to some sounds with a low attentional priority while giving enough awareness to those with higher priority.
- Comparing multiple data sets and monitoring multiple tasks is possible because of the capability of parallel listening.
- Exploring wide ranged time-sequenced data using auditory displays is preferable as the human hearing has a temporal resolution between milliseconds to several thousand milliseconds.
- Pattern recognition in data could be assisted since humans can remember highly salient sonic patterns.

In addition to these, Perrot et.al in [27] support that:

- The auditory channel can be used for orienting tasks where ears aid in directing the eyes towards an important object. This type of application is very useful when sound is used to indicate the importance of a variable with the visual channel used for extracting details.

Finally, in a well-cited study [28], it is established that:

- Our auditory sense is sensitive to temporal changes and this is very useful in analysis of periodic/aperiodic events and temporal processes.

Therefore, it is evident that the abundance of interaction tasks and applications in AmI environments can benefit by providing meaningful auditory feedback via auditory displays while also supporting the AmI paradigm's multimodal nature. The next section investigates related techniques and methods, as well as applications of auditory displays in several domains.

2.2 Auditory Representations

An auditory representation is defined as a technique or design approach that aims to, as the term may signify, the creation of a sound that can be used to communicate and represent information or a meaning [29] [30].

The growing importance of audio in computing environments has revamped the interest of using basic auditory representations. The need of meeting requirements of new interfaces has come to found new design approaches based on the basic auditory representations. Basic auditory representations can be defined as unaltered or slightly altered versions of human speech, music and environmental (natural or human made) sounds [29]. Examples of basic auditory representations include volume or length adjusted clips of instrumental music, recording of everyday and environmental sounds, human speech recordings and electronically created warning signals. New auditory representation designs occur by editing, compressing or joining elementary sounds. The timeline presented in Figure 4, shows the increasing interest in creation of new auditory representations in the past few decades.

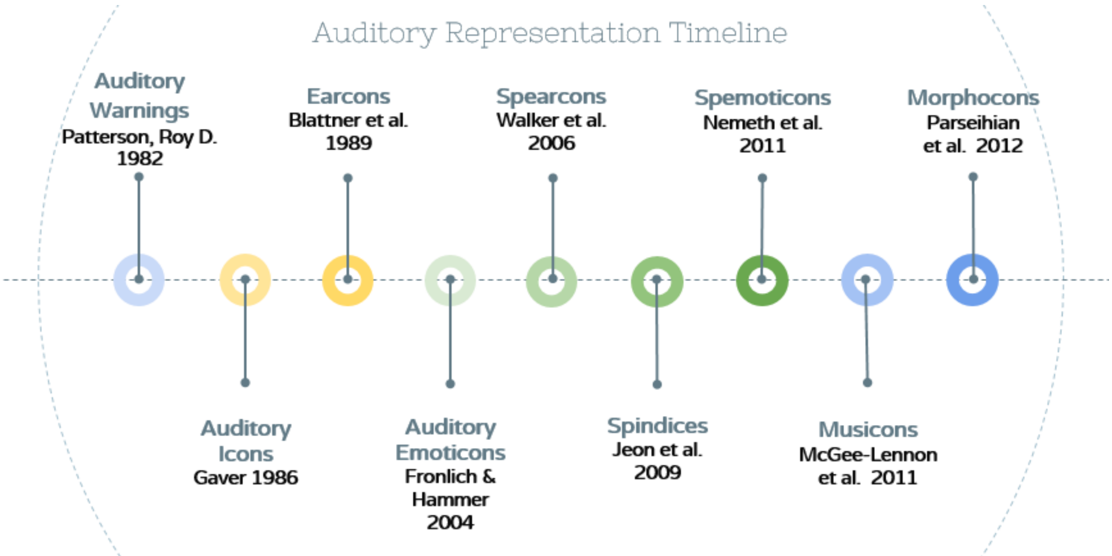


Figure 4 - Auditory Representation timeline

The following subsection will discuss basic and compound auditory representations according to their use and purpose in computing environments.

2.2.1 Basic auditory representations

The basic auditory representations are 'Auditory Icons' and 'Earcons' [29]. Those, are often viewed as two extremes of the auditory representation applicability spectrum. As seen on Figure 5, each edge of the spectrum represents a level of abstraction that regards to the ability of representing information. Effectively, that would mean that one Earcon could be used to represent a wider range of information from one Auditory Icon as the latter tend to be more descriptive ('iconic'). Of course, that does not limit the applicability of either [29].

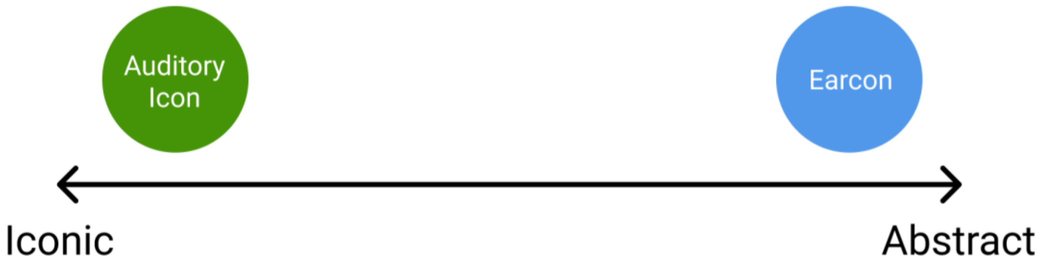


Figure 5 - Abstraction of basic auditory representations (from [29])

2.2.1.1 Auditory Icons

'Auditory Icon' is a term introduced in the literature by Gaver in 1986 [31]. Auditory icons are characterized as short, icon-like sound events that hold a semantic connection to the physical events they represent. The greatest advantage of auditory icons is that they are easy to learn and interpret. Users easily map the auditory representation to visual events after being exposed for the first time. Gaver created a conceptual way to organize auditory icon "families", by connecting the natural physical properties of objects along with the events and processes they generate. The latter concept was successfully tested in [32] [33], yielding positive results (also see 2.3 Auditory Displays).

2.2.1.2 Earcons

The concept of 'Earcons' was introduced by Blattner et al. in 1989 [34] with the description, "non-verbal audio messages in the user-computer interface". Contrasting auditory icons, earcons are message-like sounds consisting (in the simplest form) of a note or a series of notes that gain meaning through abstract relations between what is signified and the signifier. Users are explicitly required to learn the associations between a system's event and the earcon used as the auditory representation of the event. This happens as earcons do not create connections based on users' environmental experience; therefore, this induces a learning curve that must be surmounted by the user. Brewster in 1994 [35], provided a set of design principles for earcons along with a hierarchical structure of earcon "families". It was later demonstrated by Leplatre and Brewster [36] that users could recall up to twenty-five distinct earcons given that earcons were structured in a small number of conceptually and structurally distinct earcon families.

2.2.2 Speech based and emotional auditory representations

In recent decades, new designs for auditory representation have emerged that attempt using speech and human emotion as the mean for communication. Alterations of basic auditory representations, can be viewed as filling the gap between the two spectrum edges (see Figure 5).

2.2.2.1 Spearcons

'Spearcons' are referred to as acoustic representations of spoken words. The term was introduced by Walker et al. [37]. Spearcons are usually unrecognizable as speech since they are obtained by speeding up speech sounds. Through multiple studies [37] [38] [39] [40] [26, 27, 28, 29], spearcons have been proved to be suited for audio-based navigation of GUI menus. In [39] it was demonstrated that learning rates were shorter than for auditory icons, and in [40] that it was shorter than for earcons. In [40] it was also found that the learnability of spearcons in other scenarios than GUI menus was equally superior to auditory icons and earcons, hinting towards the importance of using spearcons when designing auditory interfaces, especially for navigation purposes. In regards to the generation of spearcons, it has been found that spectral analysis can be used to improve quality measures and compression ratios while the accessibility can be improved by changing various parameters as loudness and speech [41]. In the same article language independence was also investigated using spearcons in different languages and accents for the same events.

2.2.2.2 Spindices

The spindex concept was introduced by [42] as a set of enchantments for spearcons. Spindex stands for speech index and the central idea is to use accelerated versions of original spearcon sounds used for menu navigation, maintaining only the initial letter for the word or phrase (e.g. Aaaaa, Bbbbb, Ccccc). The sounds were found to be especially beneficial for users when navigating alphabetically ordered lists, especially in longer lists ([42] tested a positive effect on lists with 150 items). In [43], an adequate amount of variations for spindices were introduced with the most important being decreasing loudness of a spindex after its first occurrence while the user traverses a list; since the latter showed that users were able to use with minimal effort. Other spindex types also performed well under user tests with the exception of the minimal spindex in which only the first occurrence of a spindex can be heard (e.g. A…, B…).

2.2.2.3 Auditory Emoticons

Auditory emoticons were created as a vocal analogy to graphical emoticons using auditory derivatives of human emotional expression such as crying, laughter etc. Since graphical emoticons are used to represent emotions in a simple and limited graphical form, auditory emoticons have a similar function, using short sounds (similarly to environmental sounds of auditory icons). Auditory emoticons, in general, aim to reflect the emotional status of the speaker, are nonverbal and language independent, and can be enhanced with other sounds outside the human speech-based emotional expression scope to enable a deeper understanding of the intended meaning. Froehlich and Hammer introduced auditory emoticons in 2004 [44], as part of their automated e-mail reading application. Tests regarding the usefulness of auditory emoticons were lacking since only half of their test subjects preferred the latter over abstract musical signals. Later studies by

Wersenyi [45] [46] offered a more general set of experiments both for sighted and blind users and test results showed improved reception and suggested that auditory emoticons can aid the user increase their understanding of the emotional aspect of a conversation. The study also provided a dataset of all sounds used (auditory emoticons, icons and earcons) and the accepted (according to the study) mapping between visual and auditory representations of emoticons.

2.2.2.4 Spemoticons

Nemeth et al. [47] in 2011 first defined spemoticons as text-to-speech (TTS) based auditory representations of emotional states. In comparison to spearcons that are created based on existing words, spemoticons are synthesized using meaningless vocalized expressions that don't occur in real life. Using TTS synthesized phrases, spemoticons can be produced by modifying the intensity, pitch and temporal structure. Regarding the practicality of spemoticons, authors designed and implemented a test asking subjects to categorize forty-four distinct sounds into seven categories that reflected a message and an associated emotion (e.g. Congratulations, this is a success!" – positive evaluation, commendation; or "I am sad! I am not in good mood" – bad mood and its consequence). The sound samples that were attributed to only a few of the categories were viewed as valid emotional categories and the associated sound samples as good examples of spemoticons. Spemoticons advantages are real-time generation and straightforwardness of emotion-sound mapping, while the main disadvantage is the cultural dependency in interpretation.

2.2.3 Sonification

Kramer [48] defined sonification as "the use of non-speech audio to convey information or perceptualize data". In a broader sense, sonification describes transforming any data type into sound. Most of the aforementioned auditory representations have been used in a sonification process. Numerous studies and applications have been conducted that inspect the use of different auditory representations in diverse contexts.

Such applications lie in the field of auditory displays. Therefore, sonification applications will be extensively discussed in section 2.3 Auditory Displays. Next sections, briefly examine sonification-based auditory representations that focus on providing warning, alert and navigation information.

2.2.3.1 Morphocons

Morphocons (morphological earcons) have been proposed by [49] as a solution to the problem of generating earcons and earcon families that are in accordance to users' preferences. Centering on user satisfaction, morphocons use dynamical properties of sound (envelope, rhythm & harmonic properties, etc.) to produce wide range sound pallets. Retaining the general properties of sound, specifics can be determined based on the user's taste; for example, if a user would find the sound of a river flowing preferable, this sound (synthesized or recorded) could be mapped to the morphocon or the morphocons family general dynamic properties. Morphocons have been positively used in an audio-based navigation system for the visually impaired aiding in recognition of obstacles, landmarks and points of interest.

2.2.3.2 Musicons

Musicons range from brief to extremely brief samples of well-known music and can be used to provide reminders and other information in a plethora of scenarios both in the private (e.g. home) and the public (e.g. work) space [50] [51]. Musicons in a familiarity-learnability scale should be imagined as being between earcons and auditory icons. They are far more familiar than earcons, but less than auditory icons. They are more private than auditory icons, but less than earcons (earcons can be understood only by those who have learned the designed concept-to-sound mapping while auditory icons can be understood by any person). Regarding the generation of musicons, it is done by sampling a song or music piece and then drawing out short sections, usually between two hundred to two thousand milliseconds (ms) long.

Effectiveness of musicons can be improved by sampling user-selected parts of music [52], suggesting the use of personal musical databases. In the same study, users were asked to select a labeled “favorite” and “most representative” five second part of their favorite songs. Sampling selections into two and five hundred ms long bits, recognition rates were measured to be 69 – 78% for the 200ms and 84 – 94% for the 500ms. Evaluating melodic and rhythmical patterns as well as structure and timbre, it was established that best recognized parts of a piece are usually the first part of the chorus and the main riff or solo.

In another study by [53], an experiment was conducted to evaluate the performance of musicons and auditory icons in terms of users identifying a point of interest (POI) in a selected route using a mobile guide (android device). Participants of the experiment were asked to walk along the same route and identify POIs based on the cues they received. Cues were visual, musicons, auditory icons and mixed. In the process of designing musicons for the experiment, focus was placed in mapping melody, tempo, rhythm and lyrics of indie (non-popular) music to semantic information of a POI. As for auditory icons, those were designed with equal attention from recordings of objects or places (sound of stir-frying, chopping food etc.). The results of the experiment showed that in the context of serendipitous discovery, musicons offer a more pleasant experience and better identification accuracy than auditory icons and that auditory icons support a greater feeling of autonomy rather than guidance. Moreover, it was highlighted that auditory icons and musicons suffered from greater identification errors than visual cues.

2.2.3.3 Alerts & warnings

Computerized systems, ever since their naissance, have been using both visual and auditory warning signals to alert operators about different levels of urgency. Although mapping different levels of urgency onto sound was used early on, the importance of auditory warning signals and the need of a structured design was only apprehended in the 1980s. First guidelines were initially set by Patterson in the context of aviation [54] and later in broader contexts such as the work environment [55]. Guidelines consist of the optimal loudness relative to noise level of different frequency domains, the optimal spectral distribution and temporal characteristics such as pulse-repetition rates and rhythms. It was illustrated that good auditory warning signals consist of a series of bursts, each comprised of a number of repetitive pulses and each with a variant starting point and intensity. Through definite analysis in posterior studies [56] [57], it was also established that parameters such as harmonic composition, envelope shape, fundamental frequency and delayed harmonics have significant effects on the perceived urgency. Other parameters as the speed and repletion of pulses within bursts were found to contribute weightier than others in the perceived urgency [58].

2.2.4 Summary

Summarizing the current state-of-the-art in auditory representations, three categories of representations can be distinguished, namely basic, speech-based and sonification-based [29]. Basic auditory representations include auditory icons and earcons. Speech based, includes Spemoticons, Auditory Emoticons, Spearcons and Spindices. Sonification based, includes Auditory Warnings, Morphocons and Musicons. The figure below aims to visualize this summary as an auditory representation ontology.

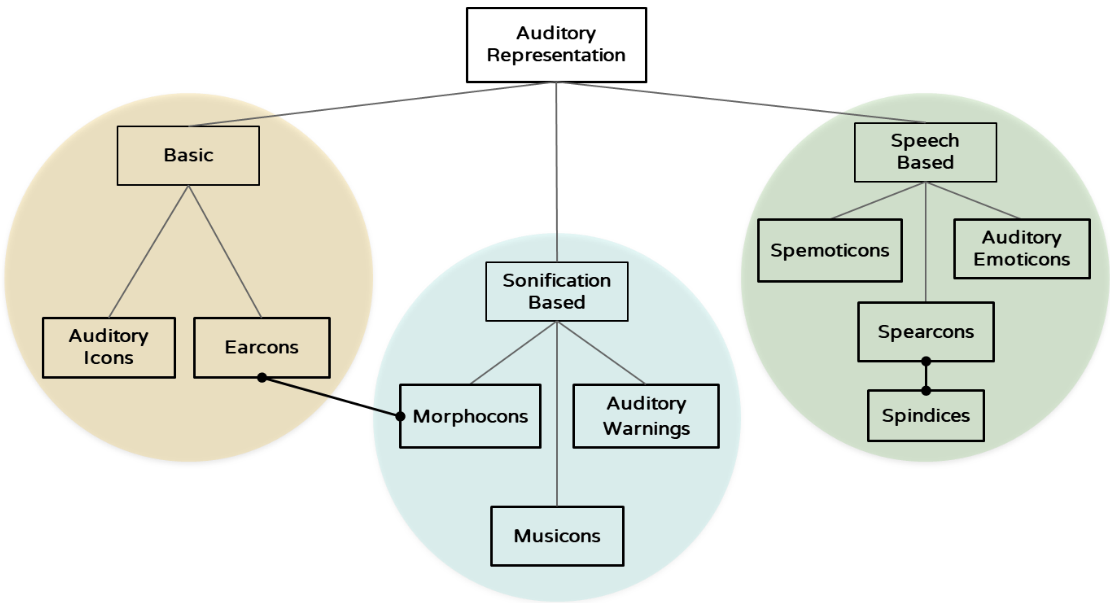


Figure 6 - Auditory Representation "Ontology"

All auditory representations are tools that a display designer can use to create an auditory display. An extensive depiction of the field of Auditory Displays is provided in the chapter that follows.

2.3 Auditory Displays

As stated in [59], the field of Auditory Displays examines the use of sound to depict information in computing environments. Applications range from informing car drivers about their driving habits and performance to promoting discovery in medical and astronomical data. For any application, modes of interaction with the display and technical solutions for collecting, processing and computing data to alter the display must be addressed. Researchers in the field usually come from multidisciplinary backgrounds such as acoustics, physics, computer science, sound engineering, social sciences and psychology, naming a few. Multidisciplinary knowledge can be a determining factor in a successful Auditory display application, especially in the case of sonification applications. As discussed in the previous chapter, sonifications are a major subset of Auditory Display and often the two terms are confused. Many but not all auditory displays are a result of sonification. Currently used techniques for providing auditory displays as well as modes of interactions for such displays will be discussed in the following subsections.

2.3.1 Modes of Interaction

Interaction with auditory displays can be divided into two cornerstones, one involving no interaction with the display and the other involving full user-controlled interaction and alteration of the display. The first has been called concert mode [60] or tour based [61] and refers to an auditory display being initiated and played while the user can only listen. The second has been called conversation mode [60] or query based [62], and describes the ability of the user to actively control the display. Control of the display can vary from choosing and changing presentation parameters [63], such as playback speed, frequency or pitch, to completely driving a display to retrieve information [64]. Therefore, applications using auditory displays can range from simply presenting the display to relying on user interaction for the presentation. With Concert mode and Conversation mode acting as the two cornerstones, auditory displays can encompass interaction that lie between the two. Establishing the two interaction modes first is necessary for understanding the different approaches that are followed when developing an auditory display. Those will be examined next.

2.3.2 Categorization & approaches

Walker et al. in [60] provide a taxonomy of auditory displays. Another way to categorize auditory displays is proposed by de Campo in [65]. Including both works, two ways to categorize auditory displays exist. These are by function of the display or by the approach/technique used for sonification. The function of the display refers to its purpose. Auditory displays are created to serve the following generic functions:

- Alarms, alerts & warnings
- Process status, monitoring, messages
- Data exploration
- Art, entertainment, sports & exercise.

Those general functions can be used to categorize an auditory display.

In this work, the categorization by approach as introduced by de Campo [56] will be focused upon. As depicted in Figure 7, four techniques/approaches can be followed when developing an auditory display:

- Audification
- Event Based – Parameter Mapping
- Continuous
- Model Based

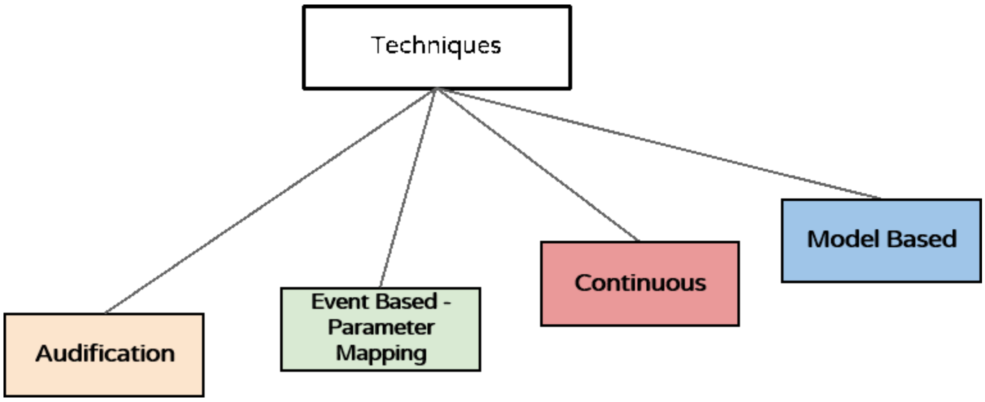


Figure 7 - Sonification approaches

For each technique an extensive description will be given and an example-application will be examined.

2.3.2.1 Audification

Audification is the most straightforward approach for sonification and refers to the technique of direct sonification, where waveforms of periodic data are directly translated into sound [48]. Seismic data have been transformed into sound using Audification to simplify categorization of seismic events [66] [67], with accuracy scores of over 90%. To make an original waveform audible, some of its properties are transformed during the Audification process. Frequency of the periodic waveform may be shifted as well as its time to make it audible to human ears.

2.3.2.2 Event based – Parameter mapping

Parameter Mapping or event based sonification is the connection of data dimensions with auditory parameters for displaying data and therefore useful information [68]. The technique is well suited for displaying multivariate data and can be used to depict data variation. The effectiveness of this technique lies in the appropriate association of a data dimension to a sound parameter, either physical (e.g. frequency) or psychophysical (e.g. pitch). The association can be either one-to-one or one-to-many. In the one-to-one association, one data feature is connected to a single data parameter, and changes in the feature affect only that parameter (e.g. temperature to frequency – 100c == 100hz). The one-to-many connection, first introduced by Kramer [48], requires that a data feature be connected to more than one sound parameters. A way to achieve that is to connect ranges of a feature to ranges of a sound parameter value. A simple example is given below where the load percentage of a power grid is connected to different sound parameters.

Table 1 - One-to-many connection of data to sound parameters

Data feature	Sound Parameter
Load(0,50)	Gain(-15dB,0dB)
Load(40,70)	Frequency(75hz,135hz)
Load(60,100)	Rhythm (long intervals, short intervals)

Both ways of data feature to sound parameter association allow for a low amount of data dimensions. This constrain of event-based sonification is required to achieve a highly perceivable display. Mapping must be qualitative or discrete to data change, and therefore series of discrete data points are needed. Usually, auditory displays created with this technique make use of the concert mode of interaction. Of course, in some cases, where the function served is data exploration, benefits can be derived from offering active listener interaction.

2.3.2.3 Continuous

Continuous sonification requires data to be logged as a time series with sufficient sampling rate so that interpolation between data points is meaningful [65], thus creating a quasi-analog signal that can be directly translated into sound. Both Audification and Parameter mapping can be encapsulated in continuous sonification providing continuous sounds. The main advantages of this approach are:

- The resulting subjective perceptual smoothness (from using interpolation, vanishing the perception of sampling interval)
- The ability to represent continuous shapes such as curves
- The ability to meaningfully represent structures based on time.

The major drawback of this approach is that it is tied to a linear, singular axis movement. Moreover, often data may signify events that are difficult to represent using this technique.

2.3.2.4 Model-based

The model based approach is the most complex technique for sonification, since there is no direct connection of data to sound. Instead, the approach introduces the idea of employing a virtual model built by the auditory display designer that a listener can interact with [65]. The various properties of the virtual model are driven by data. User input vitalizes the sonification by interacting with a setup of dynamical elements. Doing so, they are able to understand the underlying data structure. A high amount of data dimensions is supported by this approach, and different datasets can be employed by the same virtual model. Consequently, any domain knowledge can be presented via the model with the drawback of a possible bias which may lead to more limited domain understanding. The latter, in addition to "the sense of disconnection between sound and data results" [65], signifies the need for a careful and extensive design process. The main functions that are served by this approach are data exploration, learning, art, entertainment and sports. An example of this approach is the tangible data scanning sonification model of [69], where users were able to explore

high dimensional distributions of data using a physical object. User's activity, position and orientation would cause different feedback sounds emerging from excitation of the model's properties. Another interesting example is the Racing Auditory Display [70], where a virtual model was created with the goal of making racing games accessible to blind people. Blind users were able to accomplish lap times close to sighted users without diverging from the track race line.
An interesting subclass of the Model-Based approach is a yet not well-established approach called Blended Sonification introduced by researchers at Ambient Intelligence Group of Bielefeld University [8]. The approach introduces a framework under which sonifications can be designed to either be used as ambient communication channels or to display information. Guidelines require that sonifications are made with calmness, coherency, expectability and familiarity as well as physical origin. The central idea is to provide interactive ambient access to information with auditory displays remaining unobtrusive and calm. Data or/and audio signal is captured from physical and digital environment as well as users and is transformed into an auditory display that blends with a user's environment. Everyday objects are augmented with sensors that act as interfaces for user input and auditory results are simulated to match and blend with those objects. A limited amount of applications has used this approach and are described in section 2.4.3 AmI Auditory Displays.

2.3.3 Summary

Four approaches for developing an auditory display are established in literature [29] [30] [60], Audification, Event-Based – Parameter mapping, Continuous and Model Based, while two modes of interaction exist, Conversation and Concert mode. Figure 8 depicts the aforementioned.

Each approach has specific traits and a variety of functions that can be achieved, thus allowing applications in a variety of domains. Figure 9 provides an overview of each approach, while Table 2 shows application domains and special focus of auditory display in each domain.

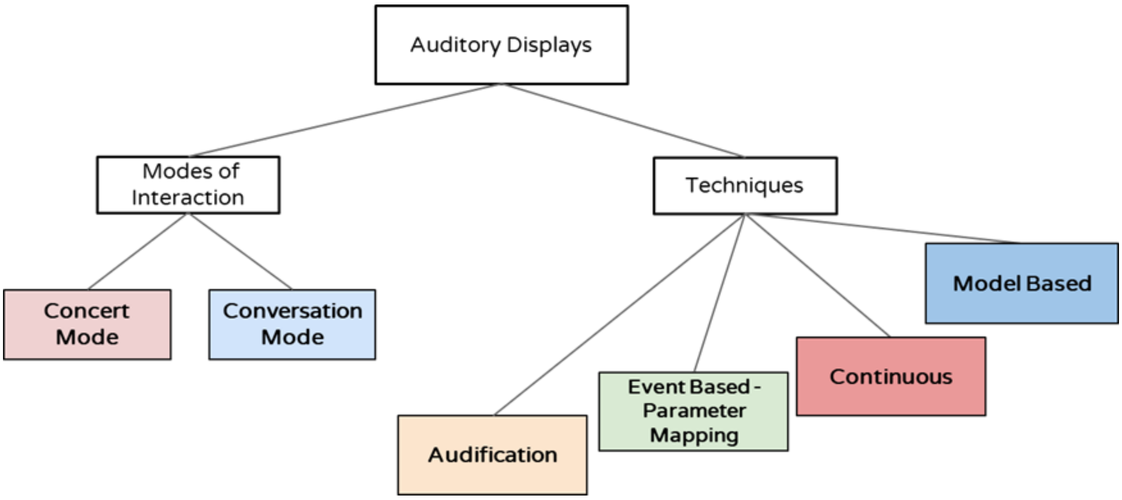


Figure 8 - Auditory Display interaction modes & techniques

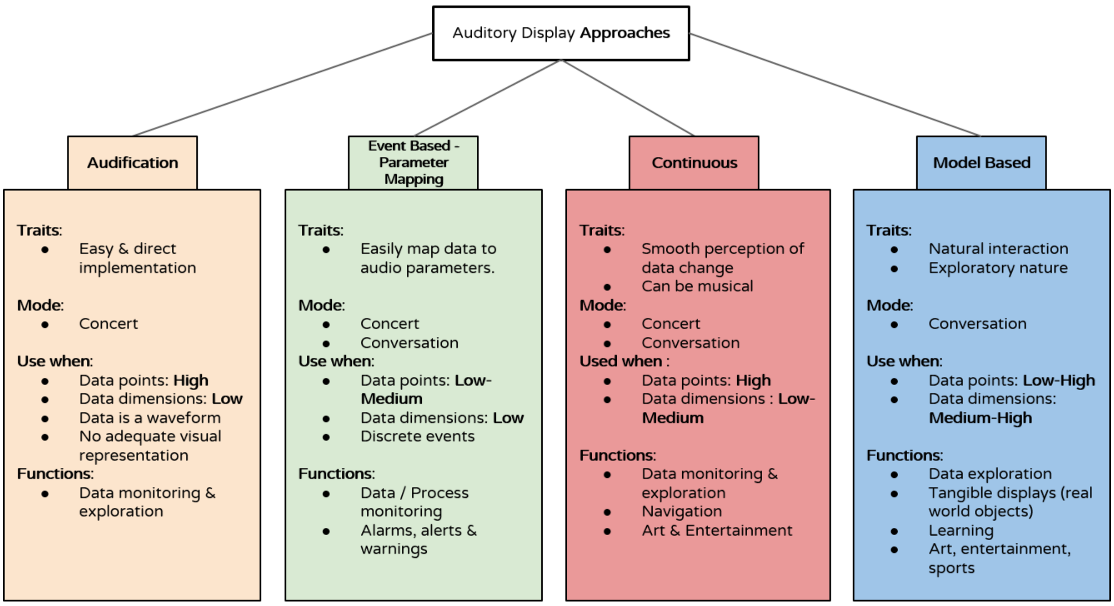


Figure 9 - Technique Analysis

Table 2 - Auditory Display application domains

Domain	Focus
Health	Surgical aid, diagnosis, rehabilitation
Workplace & Living	Notifications, alerts, presence, consumption awareness
Vehicles	Alerts, situation awareness, eco-friendly driving
Accessibility	Blind users: web browsing, games, navigation outdoor/indoor
Culture	Museum exhibits, art installations
Data	Exploration via sonification, representation

Table 2 data were extracted from published papers in the period 2005 to 2019 (available at [71]), the majority of which were presented at the International Conference on Auditory Displays (ICAD). In the next chapter, as part of ACOUSMA's related work, several applications from different domains are overviewed.

2.4 Related Work

Only a limited amount of research work exists in the field of Auditory Display that regards to Ambient Intelligence environments. To the best of our knowledge, we did not find any work to involve a platform or framework for management and distribution of personalized, context sensitive auditory displays in AmI environments. This came as an empowering motivation to work on ACOUSMA. The majority of work in Auditory Display does not consider environment and user context during deployment of auditory displays. For refining system requirements, the available research work regarding auditory displays based on AmI environment data will be presented, in addition to work involving generation and deployment of auditory displays.

2.4.1 Data sonification tools

The following sonification systems are used for generating auditory analogies to datasets and are mainly used for data analysis and exploration. Interaction with those systems is done via a GUI and there is no requirement for programming skills. For inputting data, a tabular document is needed (text, Excel/CSV), as the tools offer no database support.

2.4.1.1 Sonification Sandbox

Sonification Sandbox was created and is maintained by the Sonification Lab at Georgia Tech [72] [73]. The tool is provided for various operating systems and can be used to generate auditory graphs using the parameter mapping sonification approach and MIDI for sound output. Sonification Sandbox has been used for data exploration and analysis, education, auditory display for blind, and musical interpretation of data such as the Kepler Space Telescope data [74].

2.4.1.2 xSonify by NASA

xSonify [75] is an extension to the space physics data capabilities of the NASA Space Physics Data Facility, and offers sonification for one-dimensional space physics data such as the Cassini spacecraft crossing the bow shock of Saturn or detection of micrometeoroids impacting Voyager 2. For the latter, impacts were not visible in the plotted data but emerged when displayed as hailstorm sounds. xSonify uses the Java sound API and MIDI output for sound. User groups for xSonify involve visually impaired scientists and students.

2.4.1.3 Sonification Workstation

Sonification Workstation [76] is a very recent published work in sonification tools. It is an open-source, multi-platform application designed for general sonification tasks. Functions covered by the tool are data exploration and art (see 2.3.2 Categorization & approaches), while approaches available are mainly parameter mapping and Audification. Since the tool is in a prototype phase, a created sonification can be only exported as a JSON file that holds all needed information to be imported by a user of the same tool. The audible output of the tool can only be heard with the Sonification Workstation running.

2.4.1.4 Sonifyer

Sonifyer was developed to be an easy-to-use sonification program accessible to amateurs [77] [78]. The tool mainly uses the Audification approach and a limited version of parameter-mapping (frequency modulation synthesis). Created sonifications can be recorded and exported as an audio file (e.g. .wav). The system introduced a companion website alongside the tool with the aim of sharing audio samples and community knowledge. Unfortunately, despite its aim for wide-adoption, the tool is available only for macOS and a license is required to use the tool, obtainable only by sending an email to the authors. This is easily noticeable by visiting the Sonifyer website, with no new content posted after 2011.

2.4.2 Sonification programming environments

Several programming frameworks have been developed to allow sonification of data. Differently from sonification tools, using such frameworks requires an adequate knowledge of sonification techniques and programming as well as provided framework functions and editors. Interaction with these frameworks happens in a text-based environment, a source code editor.

2.4.2.1 SoniPy

The SoniPy framework [79] is based on Python programming language and is publicly available. The tool uses components of Python for data acquisition, storage and analysis and adds modules for perceptual mappings and sound synthesis. Through the provided application programming interface (API) [80] users can program and use functions of the aforementioned components.

2.4.2.2 Csound

Csound is a sound and music computing system that was originally released in 1985 by Barry Vercoe at MIT Media Lab and since the 90's has been under continuous development and available publicly [81]. Using the tool requires installation of an API that provides users with access to Csound's software synthesis engine. Programming languages supported include C, C++, Java, Python and others. Csound has been used for sonification in various applications. An interesting example is the use of Csound in [82] where Gerd Shmitz et al. developed a mobile sonification system to aid with motor rehabilitation after stroke, yielding positive results.

2.4.2.3 SuperCollider

SuperCollider is an open source platform for audio synthesis and algorithmic composition, used by musicians, artists, and researchers working with sound [83]. The platform is available for Windows, macOS, and Linux and features a synthesis engine 'scsynth', a programming language 'sclang', and is distributed with a dedicated IDE 'scide'. The platform has been used for sonification in numerous applications amongst them, [84] where the parameter mapping technique was used on heart rate data to determine underlying health and pathology, and [85] where the platform was used to explore sonification of the Riemann Zeta Function, a function of outmost importance in number theory for investigating properties of prime numbers.

2.4.3 AmI Auditory Displays

The following systems have been designed for AmI environments and draw real-time data produced by the environment and users to create auditory displays. Systems use a variety of hardware and software tools (most of which presented in the previous chapter) in order to provide an auditory display.

2.4.3.1 UPSTAIRS

UPSTAIRS [86] is a prototype system that uses auditory displays to connect two non-collocated people by simulating them as mutual upstairs neighbors by using contact microphones, the SuperCollider engine and speakers directed towards the ceiling. The system follows the Blended Sonification approach (described in 2.3.2.4 Model-based). Goal of the system was to create a sense of co-presence without users having to actively communicate or get distracted from other tasks. Floor vibrations from movement around the room were along with speech were transformed into auditory displays of different kinds of noises (e.g. laptop fan) and sounds (e.g. footsteps, chair movement, unintelligible speech). Though limited in size, evaluation of the system showed positive results with users enjoying the effect created by the system and reporting high levels of copresence, social presence and telepresence.

2.4.3.2 Knock 'Knock

In [8], aside from the approach of Blended Sonification, an example application was presented. With equipment and software same to the UPSTAIRS system, a door was augmented with an auditory display to inform people knocking on the door regarding the absence or presence of people on the other side. A reverb effect was added to the user's knocking sound to depict this, with a higher reverb time meaning that the person inside was missing from the room for a longer period of time. Unfortunately, no evaluation was done for the Knock 'Knock system other than the rational of creating such a system. Authors support that such an application would be calmer and unobtrusive comparing it to a much more cumbersome visual display.

2.4.3.3 Powerchord

A real-time, appliance-level sonification method for monitoring electricity consumption is described in [87]. A specially designed device could be placed between a plug and an appliance (e.g. a microwave oven) and would produce a sonification every six seconds describing current energy consumption cost.

An audio track would loop until the power consumption range changed. If no energy consumption was detected, no sound would play. The entirety of computation was done locally by the on-board microcontroller so no information could be monitored via the internet. For personalization of displayed audio, a micro-SD card with user selected tracks could be plugged in to the device and it would use those tracks for sonification.

2.4.3.4 Notifall

Notifall [88] is an experimental prototype ambient auditory notification system for non-urgent matters designed for work and home environments. The system, based on AmI environment data, would generate a variety of water sounds by dripping water on drums located in a water tank. The system was oriented towards the calm technology idea. A case study for the smart home was conducted, that would notify residents regarding energy consumption. Ten users participated in the study and expressed their liking for the system. Although seven out of ten, expressed concerns regarding the system being water-based, worried that water would be splashing.

2.4.3.5 Augmentation of an institute's kitchen

Another sonification application for enhancing energy consumption awareness is presented in [7]. The system was set up in an institute's kitchen and used two loud speakers for playback and a laptop to receive data emerging from sensors with SuperCollider for audio processing. Measurements were recorded in real-time by five wall-plug sensors that would send data when an appliance was turned on. Audio reverberation levels were mapped to depict energy consumption by appliances while the auditory display was excited by user interaction with the kitchen (e.g. when placing a cup on the countertop) detected by a microphone. A two-part qualitative evaluation was executed to assess perception of the display with findings supporting increased perceptibility with increased energy consumption (increased reverberation). Only a few participants realized they could interact with the display by speaking or making certain noises. Another finding, was that different users had dissimilar affective responses towards the system. Two users felt that the system was distracting them and switched it off by the offered emergency switch placed on one of the speakers while two others complimented on its 'subtle feedback'.

2.4.3.6 Infodrops – Sonic shower

In the work [89], two exemplary systems are showcased that utilize auditory displays in combination with visual and tangible control objects to tackle challenges for AmI environments in bathroom contexts. Infodrops, the first exemplary system, is designed for users to receive feedback regarding their energy and water consumption while taking a shower. A flow and temperature sensor is augmented in the shower nozzle that feeds data to the sonification and droplets are captured by a contact microphone (that can only capture vibrations) and are directly sonified. The auditory outcome is a coherent modified blend of the sound of water droplets that informs shower users about their consumption as captured by the flow & temperature sensor. The second exemplary system's objective is "to create immersive audiovisual scenarios by combining ambient soundscapes and mood lighting to create emotive and enjoyable atmospheres". Ambiences are triggered by tangible interaction, moving objects placed in the bathroom. Objects used, blend in as decorations to the environment. Four ambiences were created and connected to four objects. Each ambience embodied a collection of sound samples categorized in longer ambient sounds (>10s) and single object sounds (<10s). Longer sound samples would act as the sonic foundation of an ambience and single object sounds would be randomly triggered with different effects (e.g. reverb) creating a unique experience every time an ambience theme is triggered. Different lighting color themes were applied to each ambience theme (e.g. warm orange color for the meditative theme).

2.4.3.7 AudioResponse, EntranceSounds & RainForecasts

Bakker et al. [90] developed and evaluated three interactive auditory display systems. Goal of the systems was to convey information using auditory displays in the periphery that users could attend to or ignore.

2.4.3.7.1 AudioResponse

AudioResponse used a microphone to record loudness (in decibel) in an office, data that was then mapped to amplitude of an ongoing soundscape of piano tones of semi-randomized pitch. Evaluation participants reported that the display amplified everyday sounds (e.g. the office door closing) raising awareness of these sounds. Some participants found the system to be informative of their own loudness, causing them to work quitter. Amplifying already loud sounds, caused a feeling of annoyance for some participants. All participant agreed that conveyed information was not relevant.

2.4.3.7.2 EntranceSounds

EntranceSounds used a motion sensor placed above the entrance of an open office to create and play a short piano chord whenever a person was detected. The pitch of the chord root was mapped to the number of people detected in the last hour with the purpose of informing users regarding office busyness. Although, the direction of people (entering or leaving) was not detected. Participants in evaluation noted that it was easy to ignore the system while at the same time could provide useful information regarding office busy hours. None of the participants found the system as annoying and most of them realized pitch changes only when multiple people passed the door in a short time period.

2.4.3.7.3 RainForecasts

The RainForecasts system provided an auditory display every half an hour carrying information regarding the local area weather forecast, thirty minutes in the future. Data were extracted from a real-time online weather forecast and mapped to eight levels of precipitation corresponding to eight different auditory icons. For example, a forecasted 50mm precipitation per hour would trigger the auditory "mild thunder sound" to play. Evaluation that run over a period of three weeks, showed that relevance of the display varied depending on users' needs and activities. A participant who travelled by bicycle, based his departure time in accordance to the system's weather forecast while other participants were not interested in a weather forecast. A few participants reported that the system made them more aware of time passing and its sound would become more noticeable towards noon when usually office workers take their lunch break.

2.5 Discussion

Researching applications in the field of Auditory Display indicated the need for a system that supports a standardized way to store, share and deploy personalized, meaningful and context-aware auditory displays in AmI environments. Specific issues were detected in the analyzed application categories in related work:

From the analysis of data sonification tools and programming environments, it emerges that currently:

1. There is no standardized way of producing, storing, sharing and deploying created sonifications

From the analysis of AmI Auditory Displays, it becomes clear that:

2. There is no standardized hardware or software framework to deploy and control auditory displays
3. User context and preferences are not considered when using an auditory display
4. Environment context is not dynamically evaluated before the use of auditory displays.
5. They do not adjust presented auditory displays to user preferences
6. They do not adapt to changes in environment and application context
7. They do not provide flexibility in usage of output devices (meaning that applications are bound to the hardware used).

In AmI environments, where an abundance of information arises from sensors and services (see section 1.2 AmI Paradigm), it is crucial that standardized methods exist for developers to harness information, as well as to use software and hardware infrastructure to deploy their applications and create memorable user experiences. The ACOUSMA platform, that is targeted to be used for management of auditory displays in AmI environments, should allow for such standardization and address issues 1 and 2 seen above. As the primary characteristics of AmI environments require that interfaces are embeddable, context-aware, personalizable, adaptive and anticipatory (see section 1.2.2 Definition and characteristics), ACOUSMA must align with these characteristics and deploy context-aware personalizable and adaptive auditory displays, thus addressing issues 3, 4, 5, and 6. Moreover, ACOUSMA must be embeddable and use available network resources (devices and speakers) to direct and control auditory displays anywhere in an AmI environment, solving issue 7.

The identified issues have motivated the idea behind the design of ACOUSMA. This was also the spark for collaboration with AmI application developers and researchers to discover their specific needs when using auditory feedback in their applications (see section 3.1.2 User Requirements).

2.6 Objectives

Challenges set by related work and questions posted by AmI application developers and researchers constitute the foundation to create ACOUSMA. The system aims to contribute by providing:

1. A standardized way of producing, storing and sharing created auditory representations.
2. A standardized way to deploy and control auditory display playback.
3. Flexibility in use and monitoring of hardware (output devices).
4. Auditory displays that are in harmony with user and environment context (occurring at the right place, at the right time).
5. Auditory displays that can be adjusted in real time to user preferences.
6. Dynamic evaluation of environment context before and during playback of an auditory display.

7. Accurate -according to literature-, personalized and meaningful auditory displays for any given AmI environment data or event.

The above objectives are addressed by ACOUSMA, which provides a state-of-the-art platform for Auditory display experts and AmI application developers to effortlessly and effectively deliver personalized and meaningful auditory displays to AmI environment inhabitants. The next three chapters will present the design approach, implementation and usability evaluation of the system.

Chapter 3
3. ACOUSMA: Design

In this chapter, user groups and requirements will be analyzed as well as the design approach followed. Furthermore, scenarios of use for ACOUSMA will be presented. By the end of this chapter, a concrete understanding of the platform's goals and design rationale will have been established.

3.1 User Groups & Requirements

Chapter 2.5 Discussion had provided a first glance of ACOUSMA's aims and user groups. Aims described in that chapter were the most significant parameter in shaping requirement for the system. Here, each user group is going to be described along with their desired functionality for the system. Functional and non-functional requirements are going to be presented. Requirements were also affected by several brainstorming sessions that were carried out with UX/UI designers and HCI experts, the majority of which worked in AmI application research and development. All people involved in the sessions were presented with state-of-the-art in auditory representations, auditory displays and sonification. Many people had integrated auditory displays in their applications before and described issues in the workflow during sessions.

3.1.1 User Groups

This section describes the stakeholders addressed by the present research work, analyzes basic fields of interest and makes an analysis of each user group.

3.1.1.1 UG1 Auditory Display Experts

This user group consists of experts in the field of Auditory Display (see section 2.3 Auditory Displays). Users of this group are able to use sophisticated, domain specific tools to design, develop and deploy auditory displays (see sections 2.4.1 Data sonification tools and 2.4.2 Sonification programming environments). Usually, Auditory Display experts come from multi-disciplinary backgrounds combining advanced knowledge in diverse fields such as physics, acoustics, computer science or psychology.

In order to effectively deploy an auditory display, experts must first:

- Choose the most efficient sonification approach for a provided dataset and task. (see 2.3.2 Categorization & approaches)
- Have a concrete understanding of the dataset, its dimensions and points.
- Select and program appropriate hardware, output devices and input controllers. (see 2.4.2 Sonification programming environments and 2.4.3 AmI Auditory Displays)
- Pay special attention to interaction techniques, user needs and preferences.
- Design auditory representations to be used in the auditory display
- Use data sonification tools to achieve various sonification effects. (see 2.4.1 Data sonification tools)

Therefore, users of this group have to attend to a variety of diverse tasks in the process of developing an Auditory Display application. Many of this tasks must be repeated for every new auditory display application. For example, a different dataset may require a different sonification approach for its information to be presented effectively while hardware must be set up and tested accordingly for a new auditory display. Standardizing and creating software to lift the burden for these repeated tasks can help accelerate development of auditory displays.

3.1.1.2 UG2 AmI application developers

AmI application developers (as well as researchers) are involved in the creation of applications for multiple domains [4], ranging from restaurants to houses, museums and greenhouses. Users of this group program multimodal interfaces to be experienced by AmI environments inhabitants.

Applications developed by users of this group are built upon the AmI paradigm (see section 1.2 AmI Paradigm). Therefore, AmI application developers are required to have in-depth knowledge of areas such as artificial intelligence, networks and human-computer interaction. In addition to these, they need a concrete understanding of architecture and concepts of ubiquitous computing, whilst they are also capable of setting up and using hardware such as specific environment sensors and actuators. When an application is being developed for an AmI environment, developers have to accomplish specific tasks following certain general steps:

- Obtain concrete knowledge of the AmI environment (e.g. factors of developing an application for a greenhouse are different than those of a museum)
- Use specific application programming interfaces and libraries. Often, different hardware (e.g. sensors) require the use of specific libraries that AmI application developers have to explore and master in a short period of time.
- Program and test multimodal interfaces. Developers set up and adjust hardware to meet project needs and overcome environment restrictions.
- Work with data that emerge from different web services and sensors. That means that AmI application developers integrate functions that make requests to web services and manipulate useful response data for their application requirements.

3.1.1.3 UG3 AmI environment inhabitants

Users of this group are people that live in and experience AmI environments. Residents of a smart home or visitors of a museum, these users have extremely diverse needs and preferences. AmI environment inhabitants need to use available multimodal interfaces in a natural way as those adapt to their profile. This means that users of this group should be able to use various technology-enhanced (everyday) objects to control and interact with the environment. Users become accustomed to the seamless blend of technological intelligence and expect the environment to adapt to their actions and preferences. For example, a smart home user may expect the lighting to adapt and dim when she watches a movie in the living room [91] and a visitor of an AmI museum may expect to be able to interact with exhibits through simple hand gestures or her position in relation to the exhibit [92]. Moreover, users of this group are not bound to any age group. Indeed, AmI environment applications are created for various goals, from difficulty-adapting games that monitor and follow the skill development progress of preschoolers [93] to round-the-clock stress management for smart home residents [94].

3.1.2 User Requirements

There exist several techniques for user requirement elicitation. In their work [95], Antona et al. provide an overview of those techniques and methods and discuss appropriate selection under the perspective of universal access [96]. In [97] the same techniques are analyzed regarding their advantages and disadvantages. Methods described in both works include:

1. Brainstorming
2. Direct observation
3. Activity diaries and cultural probes
4. Survey and questionnaires
5. Interviews
6. Group discussions
7. Empathic modeling
8. User trials
9. Scenarios and personas

10. Prototyping
11. Cooperative and participatory design
12. Art-based approaches

Regarding user requirements elicitation for ACOUSMA, techniques 1, 2, 6 and 9 were used. The following paragraphs discuss the reasons for selecting those techniques and analyze how those were executed.

3.1.2.1 Discovering needs

Throughout a continuous discussion with AmI application developers and researchers, and after a series of brainstorming sessions, several issues were identified that AmI application developers face in the process of providing auditory feedback in their applications.

Developers shared four major issues/questions:

1. What is the right sound for a certain event or data?
2. How is an auditory display personalized to a user's needs and context?
3. When is the right time to play a sound?
4. Where is the right place in an AmI environment to deploy the sound?

Those questions come as a natural evolution from focusing on visual displays, and complement issues found in related work. Developers agreed that when the need to use sound as a medium for information arises, they are required to use advanced knowledge in an area different from their expertise (that of Auditory Display).

Usually, teams follow one of two paths. Either an expert in auditory displays is consulted or sounds are chosen according to preferences of the individual developer or team working on the component. The latter, particularly when followed for an AmI application, can lead to auditory displays of minimal information (1.) that do not appropriately take context into account (2. & 3.), and do not utilize optimal output sources (4.). Moreover, conflicts in playback and sensory overload for inhabitants can easily occur when more than one applications initiate multiple auditory displays.

3.1.2.2 Brainstorming

Brainstorming is an informal discussion where all participants freely express themselves regarding a new kind of system to be developed. The method was selected for eliciting requirements for UG2 & UG3. The reason for selecting this method was that it is comprehensible and easy to implement and it allows each participant to speak and equally share ideas. Two brainstorming sessions were organized. Both sessions were recorded in order to later create a summary of user requirements.

In the first brainstorming session, seven experienced researchers in the field of Human Computer Interaction and Ambient Intelligence participated. Participants selection was made based on their long experience in target user groups UG2 – as they have designed and developed numerous AmI environment applications – and UG3 – as they work on and experience AmI environments almost daily. The sessions lasted for approximately one hour. During the first session, background and related work on Auditory Display and Sonification were briefly presented as well as motivating scenarios of use in the context of AmI environments. Afterwards, an open discussion was initiated where participants expressed ideas for the use of auditory displays in AmI environments as well as expressed several issues they personally come across when integrating an auditory display. Moreover, participants indicated usual tasks, needs and requirements of UG3 users and the focus of a major part of the conversation addressed UG3 users that may have visual impairments or are blind.

For the second brainstorming session, four expert UX/UI (user experience/user interface) designers were invited and participated. These participants were selected as they had worked on interfaces for AmI applications and could offer great insights regarding user experience requirements for UG3. Similarly, with the first session, a brief presentation regarding auditory displays was made as well as motivating scenarios. In addition to that, an initial idea for the ACOUSMA system was described via a scenario of use. Throughout the session, multiple interaction ideas for auditory displays were discussed, focusing on how those could be adapted to a system as ACOUSMA. Participants of the second session also emphasized requirements of UG3 users with disabilities, while they also shared several non-functional requirements that are usually prominent in AmI environment applications. The session lasted for about an hour.

3.1.2.3 Direct Observation

Direct observation is the process of observing a target user's environment without interfering with their work in order capture usual tasks and events (as well as completion times) during the observed process. The method was selected for eliciting requirements for UG2. Reasons for selecting this method include that it can give a concrete and authentic idea of how users may interact with the system, that it can be helpful to validate several already collected requirements and that it is fairly inexpensive. Two of the main disadvantages of this method is that it is time consuming, as multiple session are needed and that interpersonal skills are needed to ensure cooperation of participants. Fortunately, both disadvantages were counterbalanced by the fact that the participants were willing to be observed and there was a sense of ease and cooperation.

Selected participants were experienced AmI application developers that were working on an AmI application. Two thirty-minute sessions were organized to observe work on specific parts of their project. Sessions took place the week after brainstorming and were aimed to record how AmI application developers work with API calls and how they integrate auditory feedback into their applications. Participants were prompted to initiate a session when they were about to work on any of those tasks. Each participant worked on an AmI application that used several API functions to request data from a web-service and then played a sound when a specific data-bound rule was satisfied. Results from both sessions showed that in average, ten minutes were needed for the developer to read documentation and integrate several requests to a web-service while less than two minutes were needed when the developer has worked with the API before. During the first session, the participant searched for and used an earcon from an online repository to indicate the satisfaction of the data-bound rule. Searching for the sound and integrating into their application took almost ten minutes. When asked regarding their selection criteria after the session, the developer responded that he wanted a "tone-like sound to indicate success" and that he was not sure if he selected the right sound. Participant in this phase had not participated in the brainstorming sessions and had no experience in auditory displays. Observing the participants during their work helped determining several functional requirements for UG2 and also validated some requirements of the first brainstorming session. It also sparked the idea for conducting a Group Discussion session with AmI application developers.

3.1.2.4 Group Discussion

The general idea of Group Discussion (or focus groups) is to gather a sample of a target user group and allow participants to discuss and share their understanding with the purpose of establishing a common, collective view. This method was selected as it can provide quality views on selected group's requirements. Seven AmI application developers (UG2) were invited to the group discussion. Discussion lasted for an hour and a half. First, the general concepts of auditory displays were presented, and then specific scenarios describing use of and interaction with a system such as ACOUSMA were shared with the group. The group of developers described their workflow when integrating an auditory display in an AmI application and agreed upon several issues faced. Issues were analogous to those recorded from researchers of the first brainstorming session. Furthermore, participants stated their expectations and needs from a system like ACOUSMA as well as their thought of the system architecture. All of these aided significantly in the elicitation of requirements for UG2.

3.1.2.5 Scenarios

Scenarios are narrative portrayals of a future system's interactive processes. Well-written scenarios can give realistic examples of how users may interact with the future system in a specified context. This method was selected, as scenarios can make the concept of a system easier to understand even by people with no technical knowledge. Created scenarios were used during the second brainstorming session and at the beginning of the group discussion in order to give an idea of how ACOUSMA could be used by real users. Scenarios were also the major source for eliciting requirements for UG1. Scenarios of use were refined over time and are presented in chapter 3.3 Scenarios of Use.

3.1.3 Functional Requirements

Functional requirements describe the specific behaviors of a system that are essential to support user tasks. Functional requirements are meant to signify what a system is supposed to do and usually hold the form "The user should be able to <functional requirement>". Requirements for each user group were elicited using methods described in the previous section. An aggregate summary was made from the outcome of all methods executed. In the following sections, functional requirements are numbered for each user group.

3.1.3.1 UG1-FR Auditory Display Experts

Auditory Display experts use sophisticated techniques and tools in order to create auditory displays based data. However, as it was made clear in chapter 2.4 Related Work, there exist no online platform where experts can upload, share and reuse their work. Moreover, there is no standardized way of deploying and monitoring output devices in an AmI environment, making it time-consuming to setup application-specific hardware every time an application is made. Based on related work and created scenarios, UG1 has the following functional requirements:

- UG1-FR1: The user should be able to upload and store any audio file containing an auditory representation or sonification output.
- UG1-FR2: The user should be able to access a database of auditory representations and explore items via an interface.
- UG1-FR3: The user should be able to share stored items with their team so that reusability is improved.
- UG1-FR4: The user should be able to generate auditory representations using a provided interface.
- UG1-FR5: The user should be able to utilize audio output hardware that is easily attached to the system and monitored at real-time, allowing flexibility of use.
- UG1-FR6: The user should be able to rapidly test auditory displays in an AmI environment's specific location and speaker.

3.1.3.2 UG2-FR AmI application developers

AmI application developers often need to incorporate auditory displays in their applications; to complement a visual display, to increase accessibility for their applications or for occasions where a user is unable to use a visual display (e.g. in a bathroom context). Often, AmI application developers lack in-depth knowledge of auditory displays, sonification techniques and tools, which makes difficult for them to create and integrate in AmI applications meaningful auditory displays. In addition to that, they hold great concerns regarding deployment of the right auditory display, at the right AmI environment location (and speaker), at the right time and in accordance to user and location context. Established from brainstorming sessions, group discussion, scenarios and direct observation with UG2, are the following functional requirements:

- UG2-FR1: The user should be able to direct a meaningful auditory display towards an AmI environment inhabitant.
- UG2-FR2: The user should be able to deploy auditory displays that can express a variety of data.

- UG2-FR3: The user should not have to worry about selecting an appropriate auditory display. The system should lift the burden of selecting an appropriate auditory display from developers.
- UG2-FR4: The user should not have to worry about managing context for deploying an auditory display. The system should select the right place and time to deploy an auditory display.
- UG2-FR5: The user should have playback control of deployed auditory displays; developers should be able to play, stop or pause a display.
- UG2-FR6: The user should be provided with an easy integration method for the auditory display adjustment mechanism.

3.1.3.3 UG3-FR AmI environment inhabitants

AmI environment inhabitants are diverse individuals with different needs and preferences. Depending on the AmI environment, users of UG3 may take part in a variety of activities, e.g. in the context a smart home activities may range from “cooking” to “taking a shower” or “sleeping”. Those activities affect the ability of users to receive information from visual and auditory displays. Also, the location of users in an AmI environment is not static. Users move and can be anywhere inside the environment at any time. Consequently, auditory displays need to be played and adjusted to user preferences, activities, location, as well as environment context in real-time. Brainstorming sessions as well as scenarios indicated the aforementioned constructing the following functional requirements:

- UG3-FR1: The user should be presented with auditory displays that respect their context and preferences.
- UG3-FR2: The user should be able to adjust, impromptu, a presented auditory display.
- UG3-FR3: The user’s activities as well as urgency and environment context should be evaluated and prioritized accordingly by the system before a playback action.

UG3-FR4: The user should be able to freely move around the AmI environment leaving the system to locate and take advantage of all available infrastructure (output devices) to play an auditory display near them.

3.1.4 Non-Functional Requirements

In contrast to functional requirements, non-functional requirements refer to specific criteria that will be used to dictate the operation of the system. Non-functional requirements are those quality attributes of a system that will ensure user satisfaction and increase efficiency of user goals completion. Usually, they hold the form of "The system shall be <non-functional requirement" and are applicable to all user groups. Since ACOUSMA is a platform that will be used by AmI applications to provide auditory displays in an AmI environment, non-functional requirements are adjacent to the Ambient Intelligence paradigm (see 1.2.2 Definition and characteristics) as well as its domains & applications [98] [99] [100] [101]. In this context, non-functional requirements are:

- NF-R1: **Learnability** - It is essential for every AmI environment application to have a minimum learning curve for users. To this end, the provision of meaningful and intuitive auditory displays is a fundamental requirement.
- NF-R2: **Memorability** - Interfaces must hold certain user configuration or preferences. Users must be able to efficiently start working with an interface after some time has passed since last use.
- NF-R3: **Satisfaction** - Providing an interactive interface that is satisfactory to use is considered to be a principal requirement. To achieve this, the system must offer support for recovering from errors as well as an overall feeling of smooth operation. Complex operations should be executed through intuitive and usable interfaces.
- NF-R4: **Robustness** - The system shall be extremely robust against all kinds of misuse and errors. Wrong inputs must not lead to a system malfunction or crash.

- NF-R5: **Disaster Recoverability** - As the system may be used to support vital technology infrastructure, it shall be able to recover from natural or human-induced disasters.
- NF-R6: **Availability** - The system shall do its job even in the presence of hardware component crashes, shortage of hardware resources such as storage or communication bandwidth, and other exceptional conditions.
- NF-R7: **Extensibility** - The system shall support extension by new hardware or software components at runtime.
- NF-R8: **Portability** - As the system is aimed to be integrated into any AmI application and therefore into any AmI environment, a significant requirement is for the system to be portable. The same system should be easily deployed and be usable in any AmI environment.
- NF-R9: **Security** - An AmI environment system, continuously monitors its inhabitants in order to adapt to user context and preferences. A well-defined degree of privacy must be guaranteed for inhabitants. Privacy rules must be precisely formulated and verified.
- NF-R10: **Timeliness** - Most services in an ambient environment system, have to be carried out in real time, such as the emergency treatment. Long propagation delays after detection of an emergency are not tolerable.
- NF-R11: **Resource Efficiency** - Available resources, i.e., processing power, memory, communication bandwidth, audio output devices, have to be utilized as efficiently as possible to minimize system cost and increase hardware endpoints efficiency.
- NF-R12: **Natural, Anticipatory Human-Computer Interaction** - The system shall provide human interfaces for current user groups. Each group has different requirements for interacting with the system. Responsive interfaces that adapt to user devices and multimodal interaction are a powerful approach to enhance usability.

Anticipatory interfaces, which proactively contact users in certain situations, are considered mandatory.

- NF-R13: **Adaptability** - The system shall be able to adapt itself to users and the AmI environment at runtime. Auditory displays and interfaces that adapt to context and preferences are a crucial requirement.
- For the system to satisfy NF-R13, it must also satisfy the following non-functional requirements:
- NF-R13a: **Self-optimization** - The system shall be able to adapt its algorithmic behavior to the changing needs of the AmI application that integrates it, the AmI environment as a whole and its users. An example of self-optimization is the dynamic increase of the volume of loudspeakers in the case of existing environmental noise.
- NF-R13b: **Self-configuration** - The system shall have the ability to dynamically integrate new software or hardware components and remove existing ones when not needed. Self-configuration is a form of self-adaptation at the architectural level of a system.

3.2 Design

According to the requirements presented in the previous section, ACOUSMA was designed as a modular framework to support:

1. Storing, sharing and exploring auditory representations.
2. Integrating, monitoring and testing any device with audio output capabilities.
3. Enhancement of any AmI application with meaningful personalized auditory displays that are effortlessly selected and deployed by the system.
4. Auditory displays that occur at the right place and at the right time.
5. Real-time adjustment of auditory displays to AmI environment inhabitants' preferences.

3.2.1 Software architecture

Two interconnected software packages, namely the Speaker Director and the Auditory Display Director (see Figure 10) have been designed that constitute the entirety of the platform. Each software package contains modules with specific functionality and interfaces that satisfy the set user goals and requirements.

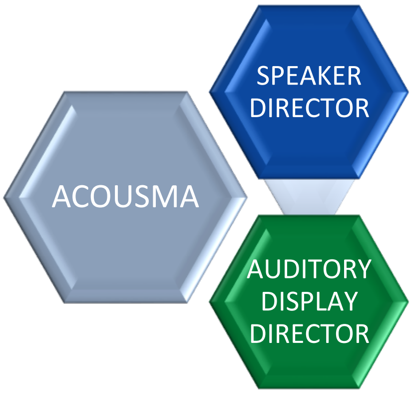


Figure 10 - ACOUSMA software packages

The platform follows a modular architecture, therefore supporting NF-R7: Extensibility and NF-R8: Portability. New modules can be added and old ones can be removed or modified over time without compromising the integrity of the platform. In addition to that, modules run independently and are able to be deployed anywhere.

In the next sections, both the designed packages and their underlying modules will be examined upon how they satisfy user requirements.

3.2.2 Speaker Director

The Speaker Director package was designed to satisfy several functional and non-functional requirements. The main role of the package is to act as the brain of the hardware infrastructure, which in an AmI environment may change dynamically. The Speaker Director has two modules, the Speakers Server and the AmI Audio Client, as can be seen in Figure 11.

Figure 11 - Speaker Director Modules

The Speaker Director was designed to meet the non-functional requirements of NF-R4: Robustness, NF-R5: Disaster Recoverability, NF-R6: Availability, NF-R7: Extensibility, NF-R8: Portability, NF-R9: Security, NF-R10: Timeliness, NF-R11: Resource Efficiency and NF-R13b: Self-configuration. Each of its modules serve specific functional requirements, outlined in the next two sections.

3.2.2.1 AmI Audio Client

The idea of designing this software module occurred as a solution to functional requirements UG1-FR5 and UG1-FR6, and in order to create the basis for fulfilling requirements UG2-FR4, UG2-FR5, UG3-FR4. Essentially, the AmI Audio Client module can be used to transform any device in an AmI environment, with audio output and network capabilities, into an over-the-network controllable audio endpoint. Installing the software module should be made easy for any device and operating system, and requires minimum effort from any user group. Devices enhanced with such module are called in the subsequent sections AmI Audio Clients. A low fidelity mockup of the interface is provided in Figure 12.

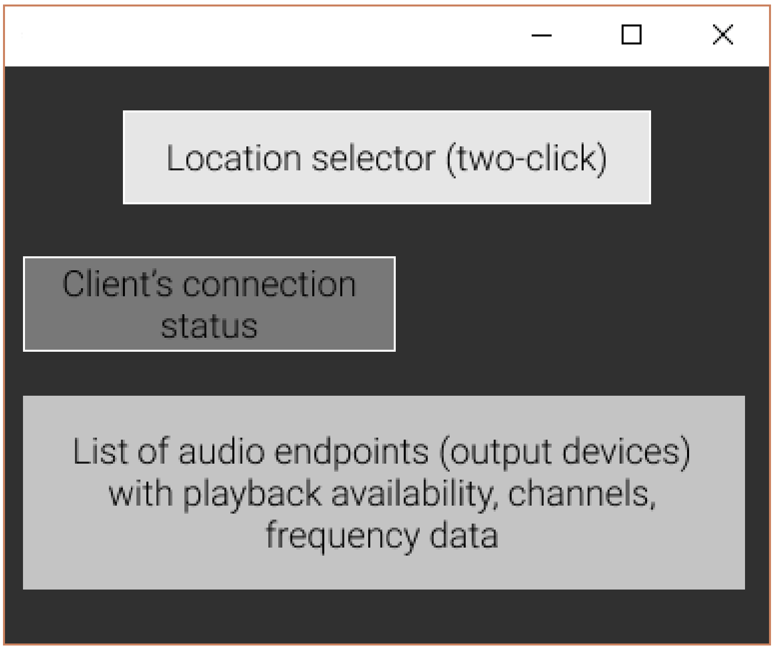


Figure 12 - AmI Audio Client interface - Low fidelity mockup

An AmI Audio Client transmits information regarding its current location as well as playback availability for its audio output(s). For example, if the module would be installed on a smart-home resident's office personal computer with a pair of loudspeakers as well as headphones connected to it, the module would transmit the location of the device, i.e. "office", and the availability of audio outputs, e.g. "Loud Speakers – Ready" and "Headphones – Busy". The availability of audio outputs should be "Ready" if the output channels are free of any sound and "Busy" if not. Moreover, AmI Audio Clients should allow simple playback controls (play, pause, stop, volume-up etc.) of audio streams. Since AmI Audio Clients will work over network, the preferred method of playing audio is that of audio streams, as they are quicker and efficient, requiring less bandwidth and no in-device storage, thus complying with non-functional requirement NF-R11.

3.2.2.2 Speakers Server

As already mentioned, AmI Audio Clients transmit certain information in addition to their ability to be controlled over the network. The Speakers Server module is the receiver of that information and can control AmI Audio Clients. The Speakers Server was designed to fulfill requirements UG1-FR5, UG1-FR6, UG2-FR5 and part of UG2-FR4 and UG3-FR4. The module works independently as a hub for AmI Audio Clients to connect to. Figure 13 illustrates that.

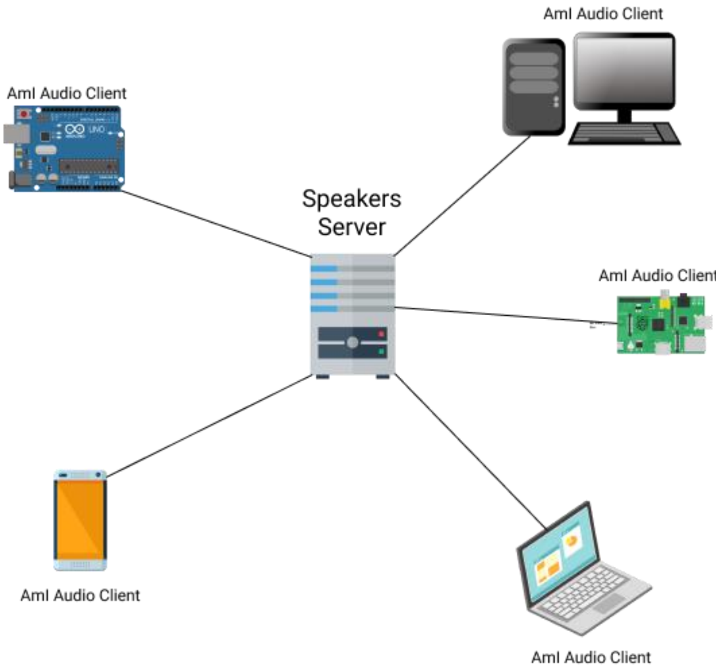


Figure 13 - Speakers Server & AmI Audio Clients

In addition to that, the Speakers Server is able to execute playback actions (e.g. play, stop, pause) received over the network. Playback actions will be either of type “action at location” or “action at user location”. For example, if the module receives a request to start playing an auditory display at a specific location (type of “action at location”), it will evaluate and select the most appropriate device for output at the selected location and then execute the action. In a case where an “action at user location” command is received, the module will first evaluate where the user is located before continuing with the aforementioned workflow of the first playback action type.

Regarding the star network topology, it is preferable as malfunctioning nodes do not affect proper operation of others, devices can be added or removed without any network disturbance, while the topology works well under heavy load with many connected clients [102]; all the above contribute to NF-R5, NF-R6, NF-R7 and NF-R8. The major disadvantage of star networks is the existence of a single point of failure, its center, here the Speakers Server. To counteract that, a mechanism has been implemented to (a) initiate backup machines in case the original fails and (b) redirect clients to the backup machine. Of course, no user group of ACOUSMA would perceive any of these, as recovery from this type of failures occurs in milliseconds.

3.2.2.2.1 AmI Speakers Explorer

Part of the UG1-FR5 and UG1-FR6 requirements is the ability for Auditory Display experts to monitor connected AmI Audio Clients in real-time and test sounds upon them. A simple, responsive graphical user interface was designed to accomplish that. The simplicity of the interface can be seen in the low-fidelity mockup of Figure 14.

Figure 14: AmI Speakers Explore - Low fidelity mockup

Through such user interface, users can see at a glance the status of AmI Audio Clients and output device availability, as well as quickly test audio at specific devices. Interface's elements also respond in real-time to status changes. For example, if an output device disconnects from an AmI Audio Client, the interface refreshes, showing that change.

3.2.3 Auditory Display Director

The Auditory Display Director package comprehends the main group of modules that user groups will interact with. Therefore, the majority of user functional requirements were used as a basis to design its modules. Each of the Auditory Display Director modules has different roles, but the package as a whole provides the infrastructure and interfaces for Auditory Display Experts to store, share and explore auditory representations, AmI applications developers to deploy meaningful auditory displays to AmI environment inhabitants, and inhabitants to adjust those displays to their preferences. Five modules were designed to achieve that, namely the AmI Audio Library, the Auditory Display Server, the API client library, the Queue Manager and the Auditory Display Recommender, as depicted in Figure 15.

Figure 15 - Auditory Display Director Modules

In the following sections, the design of each module will be presented, along with a mapping of the addressed requirements.

3.2.3.1 AmI Audio Library

Storing and sharing auditory representations as required by UG1-FR1, UG1-FR2 and UG1-FR3 involved designing a database structure to hold information regarding those representations as well as the corresponding audio file. Reviewing a case were an Auditory Display Expert would like to share created sonifications with their team, it was clear that this should be done over the network, but also meant that information regarding intended use should be retained. A cloud-based, extendable database structure was designed to support the storage of auditory representations along with metadata. Users should be able to upload representations from any device with a network connection at any time.

Metadata include information regarding the representation type (e.g. Spearcon), the sound category (e.g. Voices) and information type (e.g. Answer), as well as a list of descriptors/tags related to the semantic content of the representation. As an example, an auditory icon of the sound of heavy rain may hold the descriptors “heavy”, “rain”, “loud” and be of sound category “Weather” and information type “Predictive”.

3.2.3.1.1 Information Types

Buck, J.R. in [103] provides a classification of types of information that are transmitted via a display:

- **Instructions** - This is information that can guide user behavior in a particular way. It supports task completion and prompts on what and when to do. An example of this information type would be messages shown on an ATM machines i.e. “Please enter PIN”.
- **Command** - This information type holds usually a candid statement on what someone must or must not do. An example would be a sign posted on a door writing 'Do not enter'.
- **Advisory** - Advisory information usually takes the form of recommendation messages, similar to commands only made to direct a user away from an unpleasant situation or to assist user with useful information i.e. “We are experiencing a technical error; the show will be back on soon”.
- **Answer** - This type of information is provided when a response in needed to a specific query. Usually, it is information satisfying a question i.e. “The time is two o’clock”.

- **Historical -** Displays with this type of information usually show the state of a variable over a specific measurement (e.g. period of time). An example of this type of information would be recorded temperature readings at Heraklion (a city in Greece) in the past year.
- **Predictive -** In contrast to historical displays, predictive information enables a variable to be plotted into the future. Predictive information displays signify a prediction of how a variable will change in the future. An example would be a GPS application reading an estimated time of arrival to a destination.

In the developed system, the information type describes the intended use for the auditory display. For example, "Predictive" may suggest using the item to transmit predictive information, i.e. "There is a high chance of heavy rain in the afternoon" or "Advisory" may suggest using the item to display an advice, i.e. "You should take an umbrella with you".

3.2.3.1.2 Sound Categories

Regarding sound categories, Bones et.al [104] provide an extended taxonomy for sounds based on research of soundscape studies, nature sounds, manmade sounds, animal sounds and engine sounds. This taxonomy, although mainly used in the context of psychology, can be used to describe a variety of auditory representations. For example, one may place an Auditory Icon of a car accelerating in the sound category "Transport". alteration modified version of this taxonomy has been created to support the description of auditory representations and includes Alarms, Animals, Doors, Human Sounds, Voices, Impacts, Office, Sports, Weather, Transportation, and Household sounds. Such categorization is preliminary and new categories can be added if necessary.

3.2.3.1.3 Descriptors / Tags

In order to semantically describe the content of an uploaded item, a set of descriptors must be bound to the item. A set of one or more individual words are used to describe the content of a representation, i.e. an auditory icon of a dog barking may hold the set "dog", "barking" or an earcon meant to be played when an email has been received may hold the set “email”, “received”. Adding sets of descriptors will aid in the efficient exploration of stored items, as well as the elaboration of recommendations (see 3.2.4 Auditory Display Recommender).

3.2.3.2 AmI Audio Library Explorer

A graphical user interface was designed to support interaction with the AmI Audio Library database. A low fidelity mockup of the interface's home page, from early stages of design, can be seen in Figure 16. According to functional requirements UG1-FR1, UG1-FR2, UG1-FR3 and UG1-FR4, the interface allows users to explore, listen to, upload and generate auditory representations.

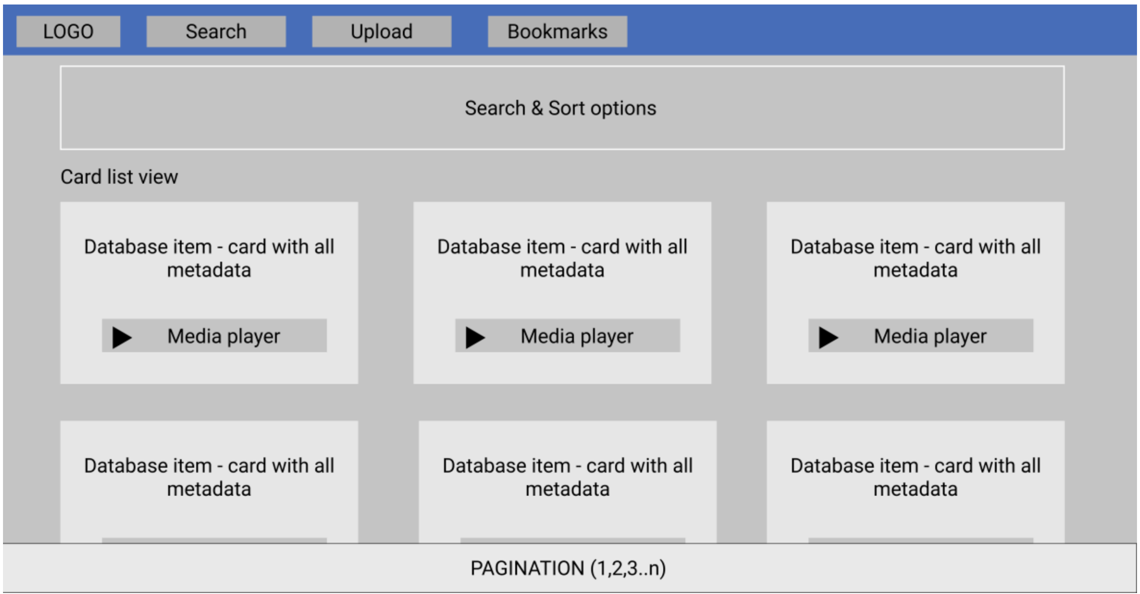


Figure 16 - AmI Audio Library Explorer - Low fidelity mockup

The interface provides the following functionality:

- searching and filtering functions to allow for efficient exploration.
- bookmarking of discovered items for later use.
- listen to uploaded items using an audio player.
- uploading and transforming any audio file into an AmI Audio Library audio stream by adding metadata through an intuitive user interface.

- generating auditory representations with the ability to directly upload them to the database or download locally for further editing.

In addition to the above, the interface is web-based responsive, allowing users to access it from any place through their preferred device, be it a personal computer, laptop, tablet or smartphone.

3.2.3.3 Auditory Display Server

This core module of the Auditory Display Director package was designed to encapsulate the functionality required by all UG2 and UG3 functional requirements. Using functions and interfaces of the Auditory Display Server, AmI application developers (UG2) will be able to easily integrate meaningful and personalized auditory displays; on the other hand, AmI environment inhabitants (UG3) will be able to receive them timely and according to context, as well as further adjust them to their preferences if they wish to. In order to achieve that, four interconnected distinct modules have been designed:

- The Auditory Display Server API
- The API client library
- The Auditory Display Recommender
- The Queue Manager

Although interconnected, those modules include distinct functions and work individually. In the following sections, their designed workflows are examined.

3.2.3.3.1 Auditory Display Server API

The role of the Auditory Display Server API is to provide functions and execute two types of requests for AmI application developers. Since an AmI environment usually consists of many locations (rooms) and is inhabited by many users, an auditory display may be directed to either a location (where one or more users may be present) or to a specific user. Therefore, functions for two types of requests must be implemented by the API, initiate playback of an auditory display (a) at an AmI location or (b) to an AmI inhabitant.

For (a), the context of the location as well as the context and preferences of the users currently present must be evaluated before deploying the auditory display. For (b), the auditory display should occur nearby that user, after their preferences and context are evaluated.

The content of these requests will have to include the desired location or desired user to deploy the auditory display to, a semantic description of what should be displayed, as well as how urgent this display is. Urgency is required to determine the playback order for multiple playback requests, and is examined in section 3.2.3.3.4 Queue Manager. For example, imagine the case where an AmI application informs Paul, a smart home resident, that the taxi he has called has arrived and is waiting for him outside; the only thing the AmI application developer would have to determine in a request to the Auditory Display Server would be the user id, here let us assume "Paul", a description of the display "your taxi has arrived and is waiting outside" as well as the request's urgency "medium urgency". The Auditory Display Server would then:

1. Use the Auditory Display Recommender module to get the most appropriate auditory display for the user "Paul".
2. Send the display to the Queue Manager with "medium" urgency.

The Queue Manager would then evaluate the user and environment context and forward the playback request to the Speaker Director to be played to Paul.

The designed interfaces to create the aforementioned requests in addition to the workflow of the Queue Manager and Auditory Display Recommender are examined in subsequent sections.

3.2.3.3.2 An interface for developers

AmI application developers write their applications in a variety of programming languages. A common way to communicate with AmI environment services is by using the provided web APIs to send requests over the network. Usually, services use web APIs to expose certain functions that can be used to retrieve useful information. Developers use the provided web API documentation to send appropriate requests. This process although time consuming at first, gets faster as developers memorize or reuse request parameters.

For communication with the Auditory Display Server API, an interface has been designed that requires minimal effort from developers to use. The interface, named API client library, provides easy to use methods that developers can use in any programming language. Developers simply have to download and import the module for their programming language. Using language rules, developers may have to create an object instance or directly use the module and its methods. Methods will have easy to understand names and rich documentation that a user's integrated development environment (IDE) can utilize to show hints during method input. An example would be:

```
PlayAtUser("Paul", "your taxi has arrived", "medium")
//or
PlayAtLocation ("kitchen", "the cake is baked", "low").
```

The IDE would then give various hints while developers complete the input. For example, when inputting the first parameter of PlayAtUser, the IDE indicates "Please provide a user ID. The user to receive the auditory display." or if a location that does not exist is inputted to PlayAtLocation the IDE should indicate "There is no such location. Please select one of 'kitchen', 'bathroom', …". With the same rationale, methods will be implemented for all provided functions of the Auditory Display Server API.

3.2.3.3.3 An intelligent mechanism module

An intelligent mechanism module was designed to recommend an appropriate auditory display for a given display description and AmI environment user profile. Section 3.2.4 Auditory Display Recommender provides an in-detail description of techniques for such systems, as well as how the system is designed to provide recommendations for auditory displays. Here, the focus is set on examining the role of the module in regards to its communication with the Auditory Display Server as well as interaction with AmI environment inhabitants.

Essentially, the intelligent mechanism module is designed to return a list of top n similar auditory displays (items stored in the AmI Audio Library) given a user profile and a description of the display. Higher similarity means that a database item matches the given description and is appropriate for the user profile.

In regards to its interaction workflow, the module receives requests from the Auditory Display Server API that provide descriptive context for the auditory display as well as a user id. The module then creates a profile for the user (if it is a new user) and begins learning their preferences. Initially, before the module learns about user preferences, the most-similar auditory display is evaluated based only on the given descriptive context and domain knowledge. Over time, the module learns about user preferences and adjusts the results accordingly.

To allow the module to learn about user preferences, a critiquing mechanism has been implemented. AmI applications, using that mechanism, can allow their users to adjust and personalize auditory displays. After the initial playback of a recommended display, users may feel that they did not understand its meaning or that it is not to their liking. Using the critiquing mechanism, they are able to state their opinion about the display with modalities provided by the AmI application (e.g. natural language). Then the application forwards the extracted information regarding preferences to the intelligent mechanism module, so it can adjust the display accordingly. The adjusted display will then be played back to the user for further adjustment. Through this process, the module learns about user preferences and use such knowledge to provide personalized auditory displays for AmI applications. Note that the module is independent and accumulates knowledge from different AmI environment applications that use its critiquing mechanism.

An overview of the interaction workflow is depicted in Figure 17 - Recommender: Interaction overview. Notice that, at step 1 of the figure, the use of an Auditory Display Server API method is assumed, and at step 5 it is executed after evaluation from the Queue manager and Speaker Director.

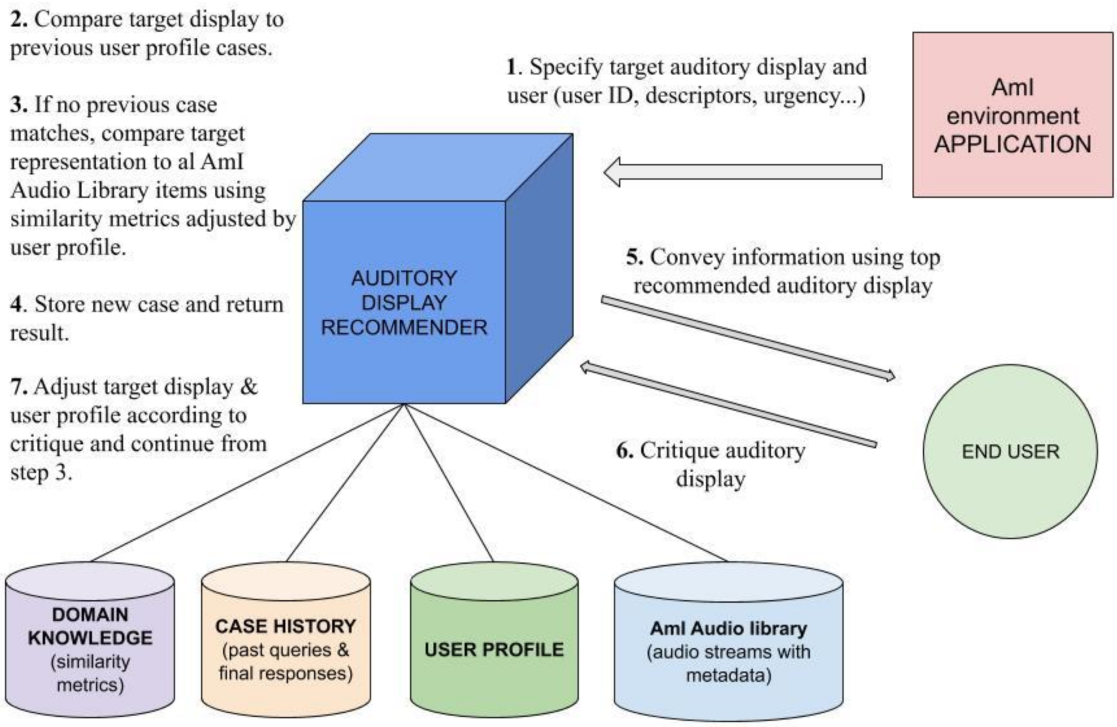


Figure 17 - Recommender: Interaction overview

3.2.3.3.4 Queue Manager

The Queue Manager module, as its name may suggest, has been designed to control the flow of the auditory display playback requests for an AmI environment. The module might receive multiple playback requests from multiple AmI applications (that use the Auditory Display Server API) in a short period of time. Satisfying requirements UG2-FR4 and UG3-FR3, the module evaluates and prioritizes playback of auditory displays according to urgency, inhabitant activities and environment context. Information for each of these parameters is obtained from different sources.

Urgency is a necessary parameter for evaluating the execution order for auditory display requests. AmI application developers have to provide this parameter's value in their requests. Ranging from low to emergency, developers have to select how urgent they consider their requested auditory display to be. To validate the defined values for the urgency parameter, a well-known time management method was used. Such method is called the Eisenhower's Matrix and is used in various domains for task prioritization [105] [106] [107]. Eisenhower, a former USA president, created this method to distribute affairs according to urgency and importance [105]. The technique suggests that each task is positioned inside a two by two matrix that helps define the order of execution. The following table presents the Eisenhower's Matrix prioritization technique.

Table 3 – Eisenhower's Matrix [105]

	Urgent	**Not Urgent**
Important	**A PRIORITY TASKS** Important and urgent tasks. Must be executed immediately. Serious consequences may occur if not. Usually emergencies.	**B PRIORITY TASKS** Important tasks but not urgent. Execution can wait until A priority tasks are complete.
Not Important	**C PRIORITY TASKS** Urgent but not important tasks. Execution for such task should be delegated or postponed until B priority tasks are complete.	**D PRIORITY TASKS** Not urgent and not important tasks. Such tasks can be avoided.

Based on Eisenhower's Matrix, four parameter values are designed to aid with prioritizing playback order of auditory displays:

1. **Emergency** (A priority tasks) – Auditory displays that should be played immediately, stopping all other playback. This urgency value should be used for emergency auditory displays, e.g. a house fire alarm.
2. **High** (B priority tasks) – Auditory displays that should be deployed as soon as possible. These displays usually relate to important matters that need immediate attention but with no serious consequences. An example would be an auditory display about an appointment with the car mechanic this evening.
3. **Medium** (C priority tasks) – Auditory displays that should be postponed until displays with high urgency are complete.
4. **Low** (D priority tasks) – Auditory displays that are not important or urgent to be played. Those should be deployed when no other displays are in queue and usually have a generic context i.e. news about the latest transfer of a basketball team.

Although the urgency parameter is a helpful indicator to define the order of execution for auditory displays, it only partly answers when a display should be deployed (for emergency it means immediately) and does surely does not answer where.

Inhabitants activities might or might not allow successful playback of an auditory display. For example, in the case a smart home resident is cooking using the oven and a frying pan, the loud noise from frying or the oven's vent may overlap with the display. In such a case two options are available:

1. Increase the volume of playback to surpass other noise.
2. Schedule the playback of the display for a later time.

The Queue manager will evaluate user activities and the environment context before initiating a display. Specifically, only in the case of emergency the module will select option 1. In all other cases, it will reschedule the deployment until the context allows it. In the previous example, the display could be presented a short time after the resident has finished cooking, just before they serve food.

Regarding the place of deployment, that depends on whether the display is requested to play in an AmI environment location or to an inhabitant. In the first case, it will be deployed at the selected location according to urgency and after all present's user activities are evaluated. In the second case, it will be deployed at the closest available speaker to the user after their activities and environment context are evaluated.

3.2.4 Auditory Display Recommender

One of the most important modules of ACOUSMA is the Auditory Display Recommender, which provides personalized, meaningful auditory displays and an easy to implement critiquing mechanism for AmI applications that users may interact with to further adjust displays to their preferences. The complete designed workflow of the module and its interaction with other modules of ACOUSMA is examined in section 3.2.3.3.3 An intelligent mechanism module. In this section, the rational for selecting a recommendation approach is discussed.

Since the dawn of the Internet, information overload has been a main issue for users and a great challenge for system developers to provide information from alternative sources, that is filtered and tailored to user preferences [108] [109]. With the aim to overcome that challenge, recommendation systems have been developed that automatically recommend items relevant to user preferences in domains such are e-commerce, e-learning and leisure [110]. Widely used real-world examples of recommendation systems include recommendations for books provided by Amazon and movies by Netflix. In a well-cited study, Ricci et.al [111] define recommender (or recommendation) systems as techniques and software tools that provide suggestions for items that are most likely of interest to a particular user. Through the large amount of research that has been made in the field of recommender systems, a precise classification of those systems has been established [109] [110] [111] according to the technique used in recommendation. A literature review of recommendation systems and current techniques is available at section Appendix B - Recommendation systems). Reading the aforementioned chapter is useful for understanding the theoretical background that leads to specific design decisions for the Auditory Display Recommender. In the next section the rationale of selecting a recommendation approach is discussed.

3.2.4.1 Selecting a recommendation approach

During the design phase of the Auditory Display Recommender, the selection of a recommendation approach was made according to the functional and non-functional requirements as well as the specific context of AmI environments where ACOUSMA will be used.

According to [112] and [113], before choosing a recommendation approach, the specific conceptual goal and knowledge source / input required by each approach must be examined. The following table presents those attributes for each of the common techniques (CF, CB, KB) used by recommenders.

Table 4 - Conceptual goals and knowledge sources of recommendation approaches [114, 118]

Approach	Conceptual Goal	Knowledge source
Collaborative-filtering	Give me recommendations based on ratings and action of myself & my peers.	• User ratings • Community ratings
Content-based	Give me recommendations based on content I have favored in my past ratings and actions	• User ratings • Item attributes
Knowledge-based	Give me recommendations based on explicit specification of the kind of content (attributes) I want.	• User specification • Item attributes • Domain knowledge

The fact that the Auditory Display Recommender will recommend auditory displays for AmI environment applications in addition to the fact that items stored in the AmI Audio Library hold specific attributes (sound category, information type, descriptors); effectively narrowed the selection to either a Content-based or Knowledge-based approach. AmI environments may be inhabited by a small (e.g. only a few residents may live in a smart home) or a big (e.g. hundreds of visitors may be at museum's hall) number of people. In order to provide auditory display recommendations for AmI environments of any population and to support non-functional requirements NF-R2, NF-R3, NF-R12 and NF-R13, it is crucial to identify potential drawbacks and advantages of each approach.

3.2.4.1.1 The Content-based approach

As seen in Appendix B - Recommendation systems, Content-based recommenders suffer from the cold-start problem, meaning that users must first rate several items before a profile is created and meaningful recommendations can be produced. If a Content-based technique were to be chosen, it would mean that AmI environment inhabitants will have to specify their preferences by rating a significant number of auditory displays before ACOUSMA could provide personalized and meaningful auditory displays. According to the functional requirements concerning AmI environment inhabitants, UG3-FR1 and UG3-FR2, it is crucial that ACOUSMA presents personalized and context-adapted auditory displays even at first use, while the presented displays must be adjustable at any time (a requirement of UG2, see UG2-FR6). A Content-based approach cannot efficiently support those requirements, as users would have to state their preferences through a constant rating of auditory displays. For example, if a user would decide that they no longer like a specific type of auditory display, it would take a significant amount of time before the system adapted to that change (not satisfying NF-R10 and NF-13a). Finally, Content-based recommenders do not rely on domain-knowledge, meaning that the recommendation process will not be able to be dynamically configured to existing and new scientific evidence and knowledge in auditory displays and representations (not satisfying NF-R7 and NF-R13b).

3.2.4.1.2 The Knowledge-based approach

On the other hand, Knowledge-based recommenders do not have the cold-start problem as recommendations do not rely on user ratings. Domain-knowledge, alone, can be leveraged to generate meaningful auditory display recommendations. Knowledge-based recommenders offer interaction approaches that allow for immediate adaptation to user preferences (satisfying NF-R10 and NF-13) and can create and retain profiles for users (satisfying NF-R2) to personalize future recommendations. Moreover, users can conveniently request recommendations with specific attributes that they desire. Therefore, users of UG2 could easily request auditory displays with specific data-driven attributes that the system could then personalize for a selected user. The major drawback of Knowledge-based recommendation systems is their reliance on domain-knowledge, which must be mapped to item attributes as well as filtering/similarity functions. This drawback would not apply in the case of the Auditory Display Recommender, as domain-knowledge is already mapped to item attributes during the design of other modules (see section 3.2.3 Auditory Display Director) and well-defined taxonomies exist in literature (see sections 2.2 Auditory Representations & 2.3 Auditory Displays).

Therefore, the focus in designing the Auditory Display Recommender is stirred towards the use of a Knowledge-based approach.

3.2.4.2 A Case-based recommender

The designed auditory display recommendation system addresses two target user groups:

- AmI application developers who need to provide input and request a recommended auditory display for a selected user or location (group of users)
- AmI environment inhabitants who need to critique a recommended auditory display and adjust it to their preferences.

As examined in Appendix B - Knowledge-based (KB), two types of Knowledge-based recommenders exist in literature, the Constraint-based and the Case-based [113]. Both those approaches offer similar interactive processes for recommendation (see Figure 57 for an overview). However, the Case-based approach does not require users to have knowledge of the product domain when querying for a recommendation and therefore will be followed here satisfying requirements UG2-FR1, UG2-FR2 and UG2-FR3 as well as UG3-FR2. Moreover, the conversational style of critiquing in the Case-based approach coincides with the non-functional requirement NF-R12 for the provision of natural human interfaces in AmI environments.

To effectively design a case-based recommender, two aspects of the system must be well defined [113]:

- **Similarity metrics**: Carefully determining the importance of the various attributes within similarity functions will lead to retrieval of appropriate results that are relevant to a given target.
- **Critiquing methods**: Providing well-designed interactive critiquing methods can aid users with better determining their preferences and exploring recommendations.

3.2.4.2.1 Designing similarity metrics

Similarity metrics are usually defined by the attributes of the item space, thus in the case of ACOUSMA from the item attributes of the AmI Audio library database. Four similarity functions have been defined to compare a provided target query with items of the AmI Audio Library database:

1. **Representation type** - Matching of this attribute will be made according to taxonomy of auditory representations determined in literature (see 2.2.4 Summary). Representation types within the same category will receive a higher similarity score.
2. **Information type** - Similarity for this attribute will be calculated based on an absolute matching of the candidate item with the target. Since information type is an attribute that aids to signify the

proposed use of the representation as an auditory display, it is considered a less important similarity metric.

3. **Sound category** - This similarity metric is based on the taxonomy connections defined in Bones et.al [104], that were modified and extended in the current work (see 3.2.3.1.2 Sound Categories). Higher similarity values occur according to connections and node distances.
4. **Descriptors** - Given a description for an auditory display, the system matches it to sets of descriptors found in items metadata of the AmI Audio Library. The semantic similarity is calculated. A higher similarity score is given when a set of descriptors approximates semantically the description given in the target-query.

For determining the representation type in the target-query, urgency information is also considered. In their requests, AmI application developers determine an estimated urgency (see 3.2.3.3.2 An interface for developers). That urgency information is then used by the Queue Manager module to evaluate and control the auditory display playback (see 3.2.3.3.4 Queue Manager). The same can be also used to select a target representation type. Literature has shown that some representation types transmit the feeling of urgency better than others (see 2.2.3 Sonification).

Regarding the importance of each of the similarity functions, this will be fine-tuned according to evaluation results.

3.2.4.2.2 Critiquing methods

Two critiquing methods for Case-based recommenders are examined in Appendix B - Knowledge-based (KB), directional and replacement critiquing. Both methods are used in the Auditory Display Recommender critiquing mechanism. Essentially, two interactive steps must be executed from an AmI application for the mechanism to function:

1. First, the application initiates a critiquing session and allows the user to comment on the display using modalities provided by the AmI application (e.g. a tablet screen or by conversation with an agent).

Note that it is up to the application developers (and their team of interaction experts) to decide how, when and where the critiquing session will be initiated.

2. Second, the application, querying the recommender, presents a list of alternative recommendations based on user critique. The user is then able to select either a preferred auditory display or further comment on the results, repeating the first step.

The two steps might be repeated several times until the user is satisfied with an auditory display. It is therefore easily noticed that only one function must be implemented and exposed by the recommender. The function, namely CritiqueDisplay, accepts user critique (direct or replacement) as input, and returns a list of recommendations. Input is a simple string of the captured user comments. The recommender then disambiguates the critique to provide appropriate recommendations. The disambiguation method maps a given critique to target attributes creating a new target-query. A naive example of such disambiguation is:

- **User critique**: "I didn't like the display as I am afraid of animals"
- **Disambiguation**: Replacement critique - "sounds other than animals"
- **Recommendations**: "auditory displays from sound categories other than 'Animals'.

3.3 Scenarios of Use

In this section, narrative descriptions of ACOUSMA's interactive processes are presented, including user and system actions and dialogues. Specifically, a set of realistic examples is provided regarding tasks that users can carry out in the specified context of ACOUSMA. Each scenario addresses one of the three target user groups.

3.3.1 A smart home emergency service

A home building company is developing a smart home emergency service. The company has decided that an emergency service should be provided with every house they build, so they assigned a team of developers and AmI environment experts to design and develop the service. The team was given a template to work on; a prebuilt house that the company offers. The experts decided that, amongst other things, auditory feedback should be provided in the case of a household emergency.

3.3.1.1 Auditory Display expert

Two members of the team who are experienced in auditory displays are assigned to design and upload to the ACOUSMA platform auditory representations for a variety of emergencies, such as “fire in the kitchen”, “gas leak” and “burst showerhead”. They, carefully design auditory representations for the list of household emergencies and upload them to ACOUSMA, using the AmI Audio Library Explorer interface. One of the two members decides to test several designed auditory representations, so she grabs her mobile phone and moves towards the template prebuild house. Going inside one of the bedrooms, she unlocks her phone and pushes on the AmI Speakers interface icon. The interface quickly loads, and she can see the available connected speakers in the bedroom. One is inside the smart closet and the other is placed on the bedside. Opening the closet, she pushes a button on the interface and listens to one her designs. Then she pushes another button and turns her head as an auditory display emerges from the bedside. She thinks that her designs are of high quality. She then tests some of the auditory displays her colleague has shared through ACOUSMA.

3.3.1.2 AmI application developer

Other members of the team start with designing visual displays, while a single developer is assigned to code events under which auditory display playback will be initiated. The developer, Martha, was recently hired by the team and did not have the chance to work with ACOUSMA or with auditory displays. Feeling excited and determined to start developing something new, she opens her computer and her favorite programming environment and loads the relevant project she is required to work on. She then downloads the API client library module for the programming language she is using and starts developing the various cases. She feels surprised that she is not required to write any complex requests or read any documentation for deploying auditory displays. The module shows helpful instructions to help Martha define input for the playback functions. A snippet of her code looks like this:

```
if ( FireDetectedAtKitchen )
        PlayAtAllUsers( "kitchen fire", emergency )
```

For selecting the PlayAtAllUsers method, Martha followed a hint given by the module “This function is suitable for initiating an auditory display to any location a user might be”. A short time after she started programming, Martha has finished. She thinks to herself “I thought that would take longer to program”.

3.3.1.3 AmI Environment Inhabitant

A young couple, Susan and Michael, decide to buy the smart home solution the house building company offers. They feel excited to install applications for their new smart home such as the stress and sleep management application. After almost a year of living in their new home, a household emergency breaks out. The couple has been using an old kitchen oven they brought from their previous home and unfortunately, it malfunctioned starting a kitchen fire. The smart home infrastructure had detected the failure promptly triggering the household emergency service with the event. Over time, ACOUSMA has learned Susan’s preferences and knows she dislikes Alarms and Auditory warning representations. Therefore, while she is taking a shower in the bathroom she is informed using a speech representation (found to be as effective as an Auditory Warning in cases of emergency) announcing “There is a fire in the kitchen, please evacuate immediately”. Michael, who is relaxing in the living room, is indifferent regarding Alarms and Auditory warnings so he is informed by a recommended Auditory Warning representation, a regular fire alarm. They both react quickly and evacuate the house.

Chapter 4
4. ACOUSMA: Implementation

In this chapter, the implementation of ACOUSMA is described. A high-level architecture portrayal of the system is presented first, followed by an analysis of each of ACOUSMA's modules and interfaces.

4.1 System Overview

As designed, the complete system consists of two distinct packages, the Speaker Director and the Auditory Display Director. Both packages, in combination, aim to enhance Ambient Intelligence environments with personalized auditory displays. Auditory displays are deployed based on user, environment and application context, as well as user specific preferences regarding auditory representations and sound categories. A system overview for ACOUSMA is shown in Figure 18. All modules and interfaces will be analyzed in the sections that follow, although a short descriptive summary is provided here.

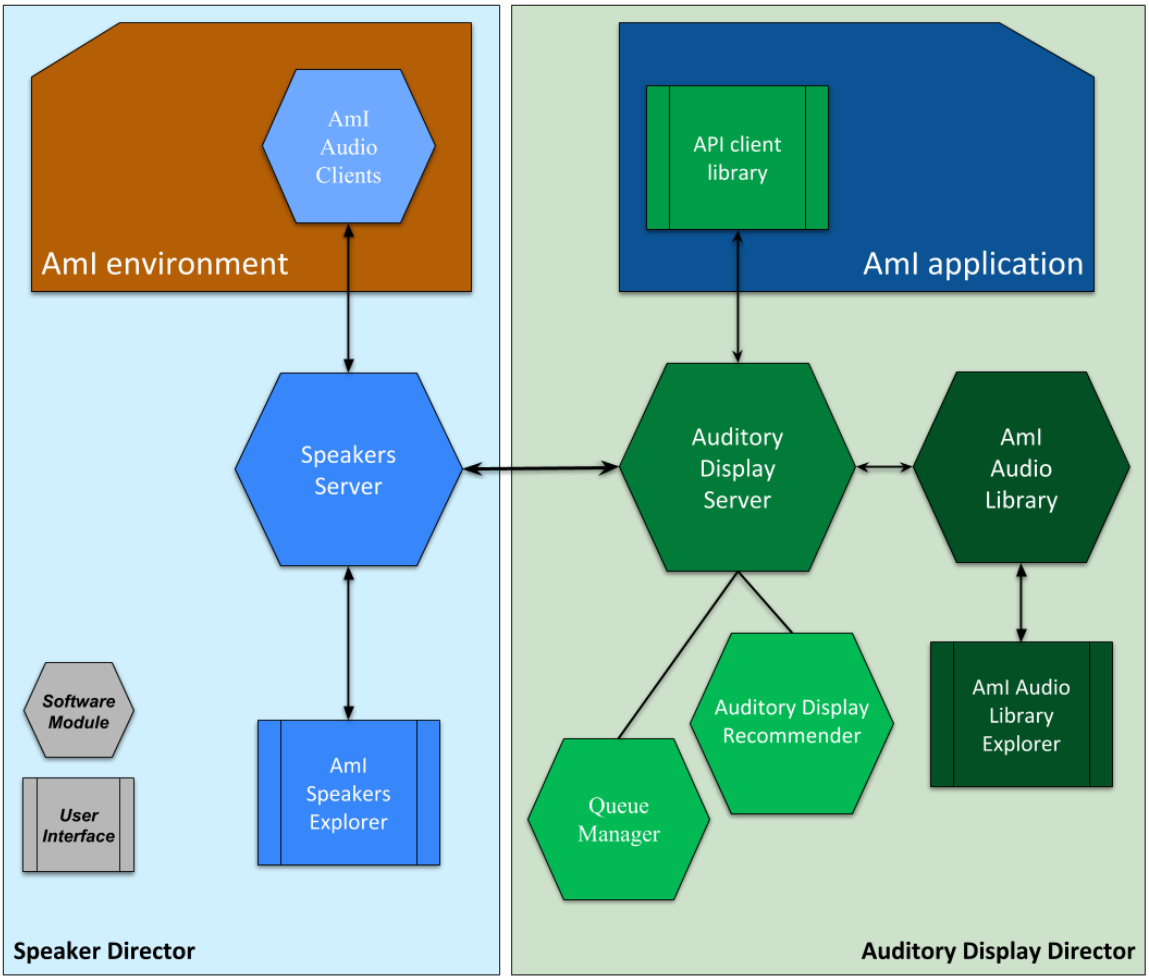


Figure 18 - ACOUSMA system overview

The Speaker Director package's role is to act as the backbone for deploying and monitoring auditory displays anywhere in an AmI environment. The package consists of two software modules and one user interface:

- **AmI Audio Client**(s) – a software module that can be set up at any device within an AmI environment transforming it into a network-controllable audio output.
- **Speakers Server** – a software module to act as a connection hub for AmI Audio Clients, controlling auditory display playback and monitoring status of connected clients.
- **AmI Speakers Explorer** – a graphical user interface to run tests and monitor status of AmI Audio Clients.

The Auditory Display Director package has three roles, to enable storage of auditory representations, deployment of auditory displays from AmI applications and management of playback according to context. The package has four software modules and two user interfaces:

- **AmI Audio Library** – a software module, a database for storage of auditory representation audio streams and metadata.
- **AmI Audio Library Explorer** – a graphical user interface that is used as a CMS for the AmI Audio Library. Users can explore uploaded items while they can generate, upload and store their designed auditory representations.
- **Auditory Display Server** – a software module that manages playback of personalized auditory displays according to environment and user context. The module contains two other modules:
    - **Queue Manager** - a software module that evaluates the order of playback for auditory display requests according to urgency, environment and user context.
    - **Auditory Display Recommender** - a software module that keeps an adjustable preferences profile for each AmI environment inhabitant and provides personalized auditory display according to those preferences.
- **API Client library** – an application programming interface module that can be imported into an AmI application project of any programming language. Users can request and control auditory display playback using functions of this module. (requests are received by the Auditory Display Server).

4.2 Speaker Director Package

In order to present an auditory display, the most basic requirement is an output device, a speaker. In an AmI environment, intelligent and intuitive interfaces are embedded in various everyday objects, and the recognition and response to the presence of people is possible in a seamless, unobtrusive and usually invisible manner [4]. Therefore, output devices in such an environment are required to be ubiquitous, enabling the distribution of auditory displays anywhere and in every condition. The Speakers Director package was developed as a solution to support this requirement. The package consists of two modules, the Speaker Server and the AmI Audio Client, as well as a simplistic graphical user interface called AmI Speakers Explorer.

4.2.1 AmI Audio Client

An AmI Audio Client can be described as a physical device (e.g. a computer, smartphone, TV) with one or more audio output devices, running software enabling playback control of these devices over network. A cross-platform software was developed that enables easy integration and over network playback control for any device usually found in AmI environments. Currently, devices with the most popular operating systems are supported (Linux, Windows, Android, macOS, IOS). Consequently, any device with audio output capabilities can be enhanced with the software and transformed into an AmI Audio client.

In essence, an AmI Audio client enables the seamless distribution of auditory displays in AmI environments through a network of AmI Audio Clients. The Speaker Server controls the communication and distribution of auditory displays to the AmI Audio Clients. Every AmI Audio Client establishes a connection to the Speaker Server and sends real-time updates regarding output devices playback status, as well as the client's location (e.g. "kitchen" in a Smart Home). The relation between the AmI Audio Client and Speaker Server is many to one, while the network topology is a wireless star topology with the Speaker Server acting as the central hub.

4.2.2 Speakers Server

The Speaker server is a RESTful API, developed for easy direction of a playback request to an AmI Audio Client. As mentioned above, the Speaker Server keeps track of each connected AmI Audio Client's output devices status, as well as the of the client's location, allowing an auditory display request to be directed to any of the available output devices.

The exposed functions of the RESTful API are the following:

1. Playback at location – an auditory display playback is requested for a specific location.
2. Playback at user – an auditory display playback is requested for a specific user of the AmI environment.
3. Get connected clients – returns the list of connected AmI Audio Clients containing information regarding output devices.

In the case of the first function, the playback is directed to a specific location, while in the case of the second an AmI Audio Client near the targeted user is selected. Both exposed functions are used by the Auditory Display Director package, so there is no immediate user interaction. The third function returns the connected clients' status. This is used by the AmI Speakers Explorer, described next.

4.2.3 Setup & AmI Speakers Explorer Interface

A user interface testing and monitoring status of AmI Audio Clients has been developed in the form of a web-application called AmI Speakers Explorer. Its simple interface was developed for Auditory Display experts and AmI application developers willing to quickly view available locations and output devices and deploy tests while developing an application for an AmI environment. Users can quickly view all registered clients (offline and online) as well as their devices playback status, in the form of a card list. The information is made available in real-time by the Speakers Server.

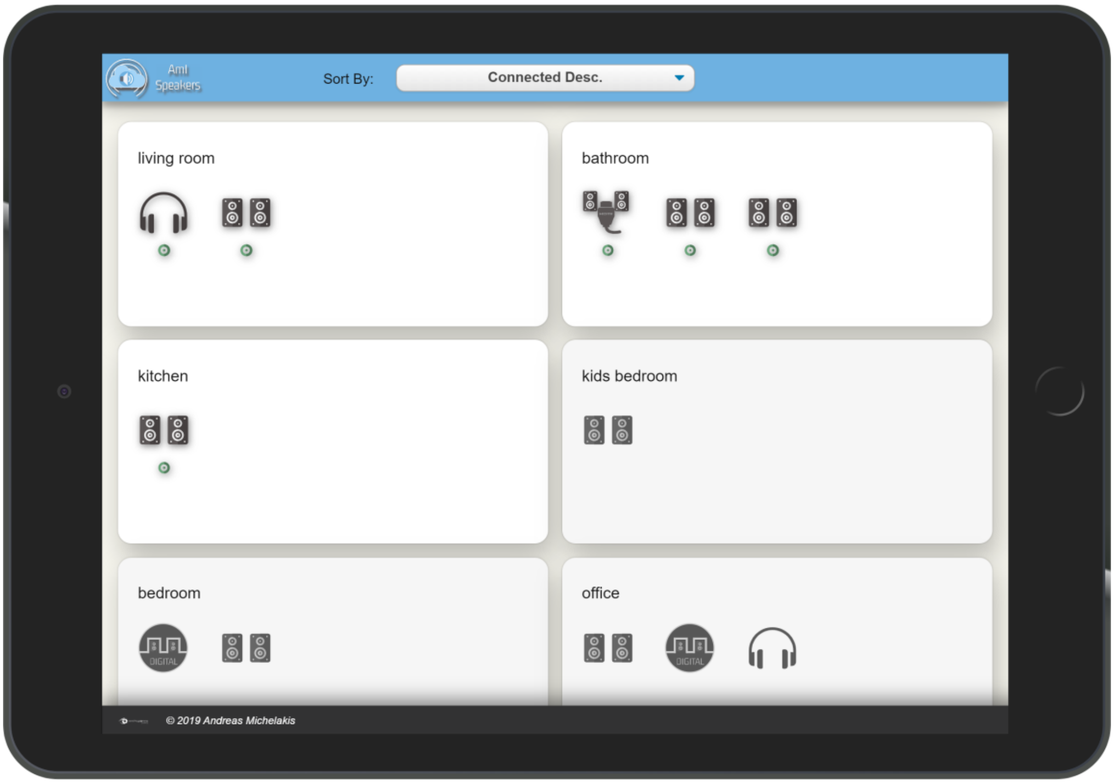


Figure 19 -AmI Speakers Explorer: Monitoring device status

The application is responsive and can be loaded from any device with an internet browser, meaning that a developer can use it on the spot through their smartphone, tablet or PC.

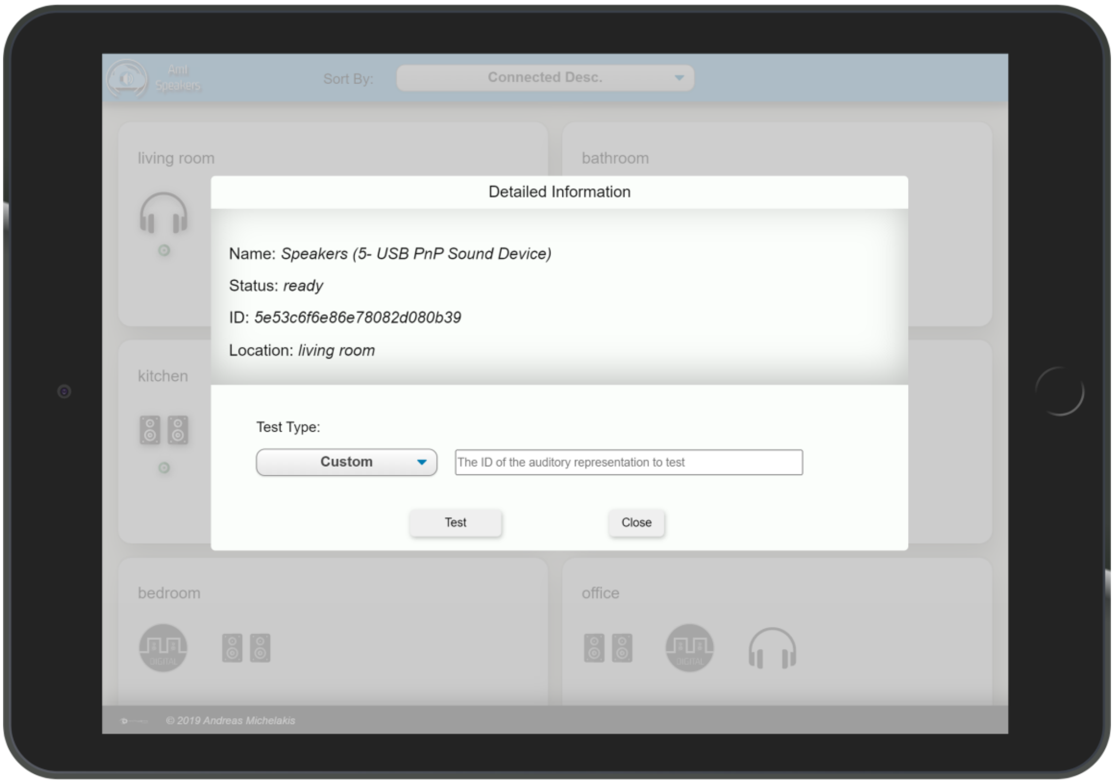


Figure 20 - AmI Speakers Explorer: Executing tests

In the case a developer or a technician would like to add a new AmI Audio Client, they only need to run the provided software at a device with an available audio output and select a location for it. Figure 21 shows the simple interface for the Windows operating systems.

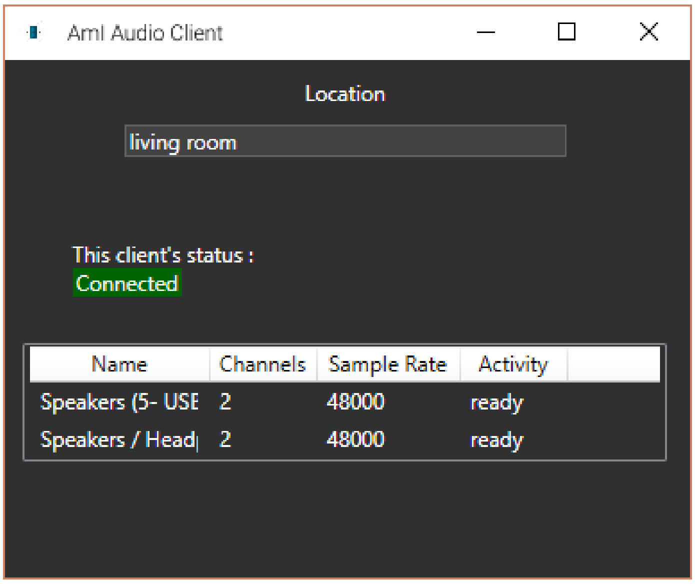


Figure 21 - AmI Audio Client: Software interface

The change in the AmI environment audio output infrastructure would be immediately visible at the AmI Speakers Explorer interface as the AmI Audio Client software automatically updates the Speakers Server regarding its status.

4.3 Auditory Display Director Package

The Auditory Display Director package's components provide an extensive suite of interfaces for AmI application developers and Auditory Display experts that they can use to deploy meaningful and personalized auditory displays to AmI environment inhabitants. The Auditory Display Director package consists of five components, the AmI Audio Library, the Auditory Display Server, the API client library, the Queue Manager and the Auditory Display Recommender. The last three can be viewed as an extension of the Auditory Display Server.

4.3.1 AmI Audio Library

As designed, the AmI Audio Library is a cloud-based database structure to store the complete variety of auditory representations found in auditory display design. The structure in which database items are stored enables retrieval in the form of an audio stream, including information regarding the representation, sound category and information type, as well as a list of descriptors/tags describing the semantic content of the representation. Items stored in the database are retrieved and used from various other modules of ACOUSMA. The AmI Audio Library Explorer uses those items to populate the interface; the Auditory Display Recommender module uses them to produce recommendations and the Auditory Display server forwards items to the Speaker Director to be played.

4.3.2 AmI Audio Library Explorer

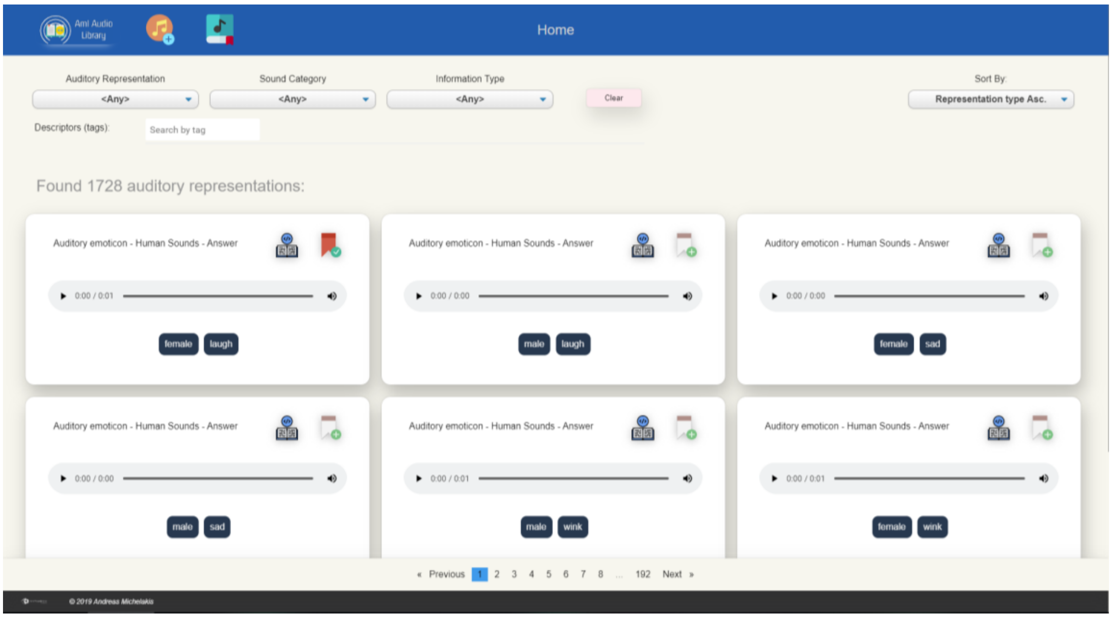


Figure 22 - AmI Audio Library Explorer: Searching for a representation

A content management system (CMS) was developed for the AmI Audio Library to enable searching, listening, generating and uploading auditory representations. The CMS was developed as a responsive web application, easy to use in all devices with a web browser. The system is intended to be used by auditory display experts. The application offers three main functions:

1. Searching
2. Adding a new representation
3. Bookmarking for future use.

4.3.2.1 Searching

The user of the application can easily search for a specific auditory representation by filtering the database items by a specific representation type, sound category, information type or tag. Results appear in the form of cards containing information relevant to database items meeting the search query. The user can listen to an uploaded representation and add it to her bookmarks by clicking on relevant buttons.

4.3.2.2 Adding a new representation

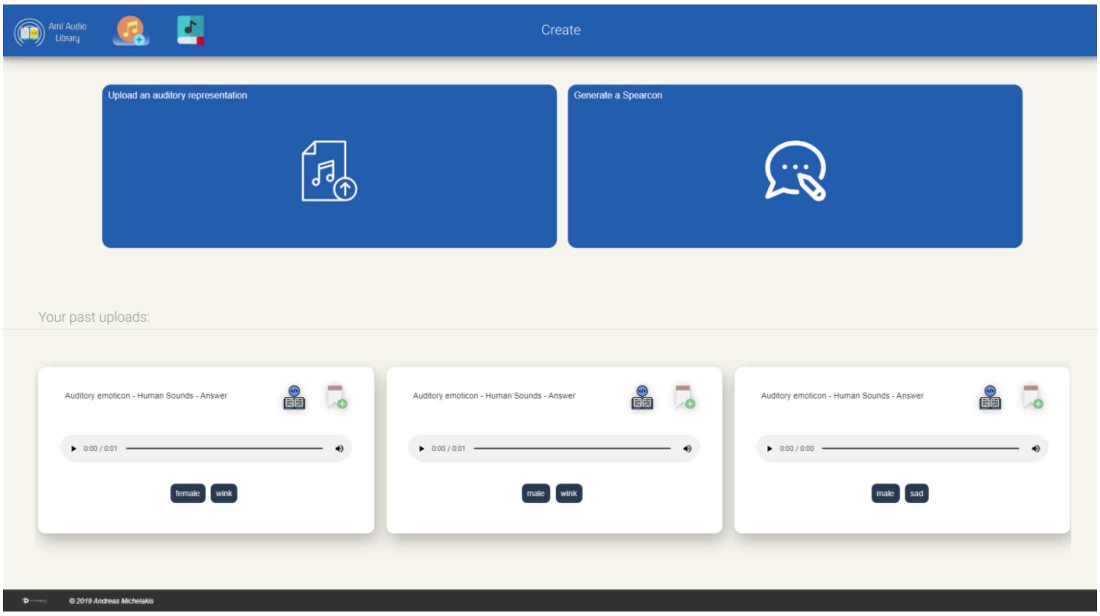


Figure 23 - AmI Audio Library Explorer: Adding a new representation

The user can either choose to upload a created representation to the database or generate a new auditory representation. Currently, only the generation of Spearcons is supported. Moreover, in this page, the user can view their already uploaded items.

4.3.2.3 Uploading a representation

Filling a short form, the auditory display expert can upload a new auditory representation to the database. For convenience, tags are automatically extracted by the filename. The user can then add additional tags or remove the extracted ones before uploading.

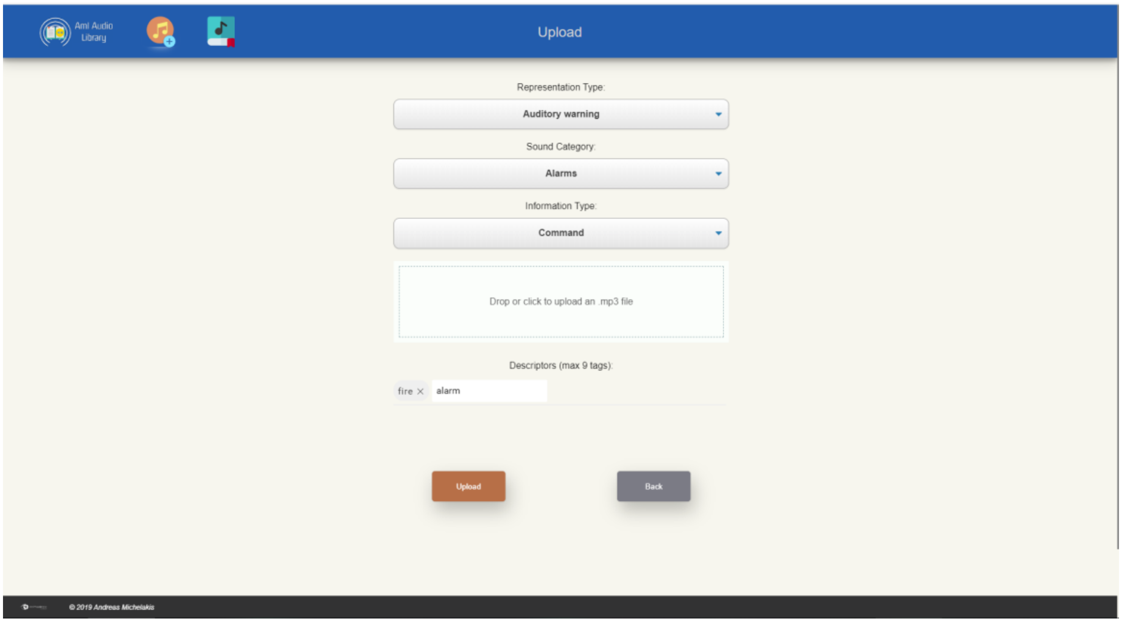


Figure 24 - AmI Audio Library Explorer: Uploading a representation

Clicking the upload button, the representation is uploaded to the AmI Audio Library. After an item is uploaded, the user has the chance to review the item on the interface. If there is something wrong or missing, the user can choose to remove the item in order to apply the desired changes.

4.3.2.4 Generating a TTS-based representation

The user can choose to generate Spearcons in one of five languages (English UK/US, French, German, Greek, Spanish). Currently, all used voices are female. The user can generate multiple Spearcon variations and listen to the generated results. If one or more results are to the user's liking, the user may choose to download them for further editing or directly upload the generated item to the AmI Audio Library.

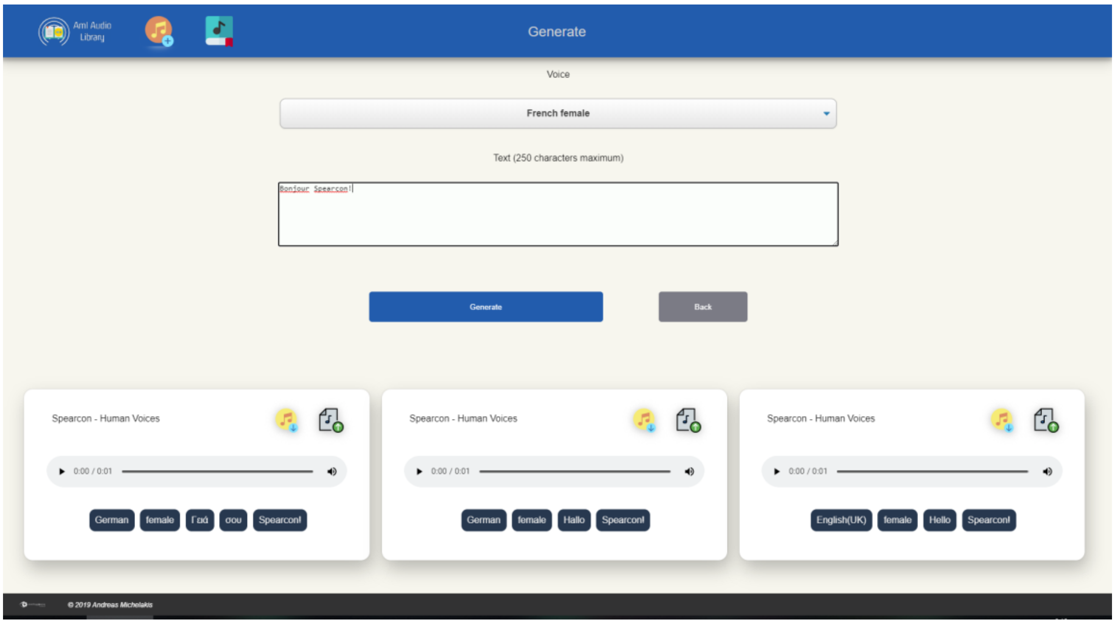


Figure 25 - AmI Audio Library Explorer: Generating Spearcons

4.3.2.5 Bookmarking for future use

After searching and listening to representations, a user may choose to bookmark several auditory representations. Bookmarked items can be viewed in the form of cards similarly to the home page, containing easy-to-copy code snippets of the API client library (examined in 4.3.3 Auditory Display Server). Moreover, the user can listen to bookmarked auditory representations. The user can remove an item from their bookmarks by clicking on the relevant icon.

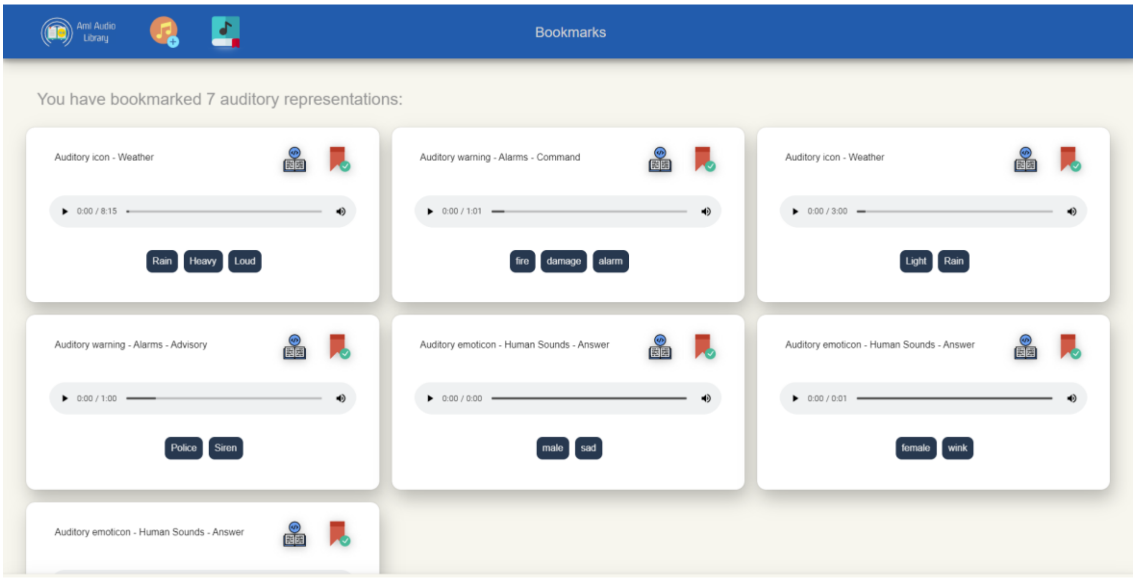


Figure 26 - AmI Audio Library Explorer: Bookmarks

4.3.2.6 Additional features

As a personalization feature of the application, an automatic detection of user's preferred colored scheme is implemented. Additionally, content viewing across different devices is supported. Figure 27 shows two mobile devices of users with different color scheme preference.

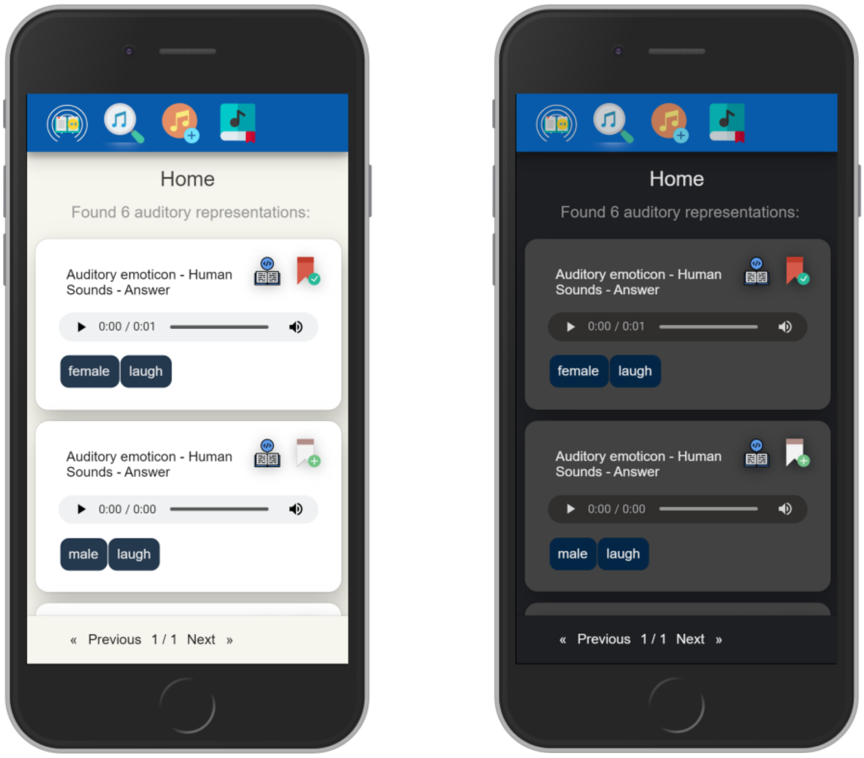


Figure 27 - AmI Audio Library Explorer: Adaptive interface

4.3.3 Auditory Display Server

Four components constitute the Auditory Display Server; a RESTful API (referred to as the "Auditory Display Server API"), an API client library, a Queue Manager and an Auditory Display Recommender. Each of these components play a significant role in providing AmI environment users with personalized auditory displays and empowering developers with tools to easily incorporate personalized and data-driven auditory displays into existing and new applications.

4.3.3.1 Auditory Display Server API

This RESTful API exposes functions that application developers can query to:

A. Start and control a specific auditory display playback (at a location of an AmI environment or at the location of a specific user of the environment).

B. Request and control a recommended auditory display playback (for a specific user).

After each query, a specific workflow is followed.

In the case of A., a specific representation is given with a playback action, location or user and urgency level. All of those parameters are passed to the queue manager in order to be then sent to the Speaker Director for playback.

In the case of B., specific input for the recommender is given, as well as the desired playback action and urgency level. The recommender then produces a recommended representation which is passed to the queue manager in order to be then sent to the Speaker Director for playback.

In order to better understand the two workflow directions, the individual components that take part in the process will be analyzed.

4.3.3.2 API Client Library

In order for application developers to easily query and use the exposed functions of the RESTful API, an API Client Library was established. Developers with applications in different programming languages can add language specific modules that ease communication with the RESTful API. Currently, modules for the C# and Typescript programming languages have been developed. The modules are well documented, designed for developers ranging from zero to expert knowledge in auditory displays. Practically, experts in auditory displays may want to have better control using the specific auditory display playback functions, while developers with no knowledge of the field can use the recommender for producing auditory displays.

Some of the module functions and in-IDE features are shown in the Figure 28 & Figure 29.

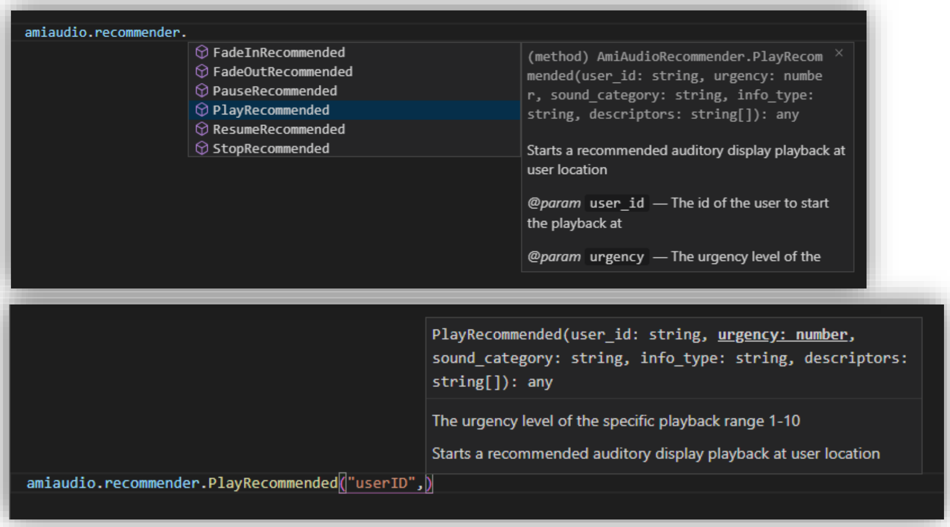


Figure 28 - API Client Library: IDE recommender functions

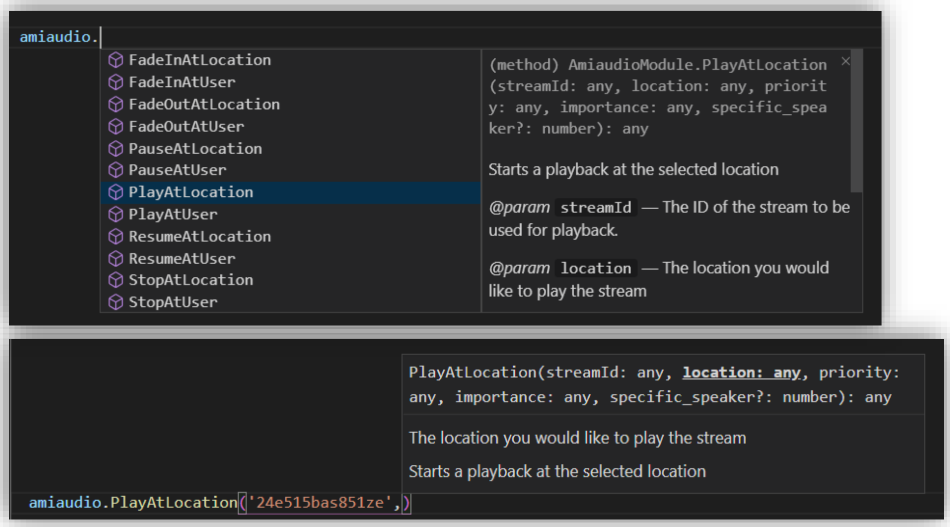


Figure 29 - API Client Library: IDE - expert functions

4.3.3.3 Recommender

A knowledge-based recommender was designed using current state-of-the-art approaches in auditory representations and displays, as well as the specific type of recommender and technique. As decided during the design phase (see section 3.2.4 Auditory Display Recommender), the recommender uses the case-based technique to provide recommendations. The aim of the recommender is to return a single recommended auditory display for a given query-target while keeping the recommended item personalized to user preferences. A user-profile database is held along with an auditory representation database and case history for each user.

An example of a query-target is depicted in Figure 30.

```
{
    "user_id": "Susan"
    "urgency": "medium",
    "info_type": "Predictive"
    "sound_category": "Weather",
    "descriptors": [
        "heavy",
        "rain",
        "road"
    ]
}
```

Figure 30 - Recommender: Target-query example

Case-based recommenders use similarity metrics to retrieve meaningful results in response to a specific target query. Here, the recommender runs a similarity function for each feature found in the expanded target (T),

$$f(\overline{T}, \overline{X}) = \frac{\sum_{i \in S} w_i \cdot Sim(t_i, x_i)}{\sum_{i \in S} w_i}$$

Equation 1 - Recommender: Similarity Metrics

comparing it to each feature found in every item in the auditory representation database (X). The score of each similarity function is regulated by a specific weight configured by auditory display experts. Each function has a feature specific asymmetric reward (a) configured by user's preferences.

Four similarity metrics are calculated:

1. Representation type
2. Information type
3. Sound Category
4. Descriptors similarity

The final resulting similarity is the weighted average of the above four.

Before any similarity metrics are run for a received target, the recommender first checks for similar cases in the case history. User profile features along with the received target are compared with past cases data. In the event a similar case is found, the result is returned. Otherwise, the query-target is expanded. The case history is a way for the recommender to learn about each user preferences and adapt its results. The format of a case in the case history is of query - user profile - result. It should be mentioned here that the result is dependent on the query and user profile combination, meaning that given the same query & user profile, the same result will be produced.

The target-query expansion happens in order to obtain the same features found in items of the auditory representation database. The representation type feature is elicited by using domain knowledge connections between representation types and urgency & between sound categories and representation types. If no sound category is provided, then the above connections are also used. In the case an information type is not provided, the feature is not checked for similarity.

4.3.3.3.1 Representation Type

The representation type feature is checked for all items in the auditory representation database. Distance between target & item nodes in the auditory representation taxonomy is checked. Efficient algorithms to find the distance between two nodes of a taxonomy have already been established in literature and answer to the Least Common Ancestor (LCA) problem [114] [115]. An LCA algorithm was implemented here to calculate

$$Sim(t_i, x_i) = 1 - |t_i - x_i| + \alpha_i \cdot I(x == t_i) \cdot |t_i - x_i|$$

Equation 2 -. Recommender: Similarity calculation & asymmetric reward

the distance between two item nodes. The closer two nodes are, the higher is the similarity score. The asymmetric reward is defined by the user's representation type likes & dislikes, so that a = 1.0 if she likes a representation type and 0 if she does not.

4.3.3.3.2 Information Type

The information type feature is checked for all items in the auditory representation database. The absolute similarity is checked here, so if the items are the same the score is 1.0, if not, 0; User preference is not in play here, as there is no evidence for correlation between information categories and user preferences. The information category merely aims to provide a classification of the use of the auditory representation in an auditory display. (also see 3.2.3.1.1 Information Types)

4.3.3.3.3 Sound Category

The sound category feature is checked in a manner similar to the representation type feature for all items in the auditory representation database. Distance between target & item nodes in the sound category - auditory representation taxonomy is checked. The closer two nodes are, the higher is the similarity score. The asymmetric reward is defined by the user's sound category likes & dislikes, so that a = 1 if she likes a sound category and a = 0 if she does not.

4.3.3.3.4 Descriptors

The descriptors feature is checked for all items in the auditory representation database. A two-layer neural network is used to evoke semantic similarities between the two descriptor arrays. Specifically, the well-cited word2vec model and application by Mikolov et al. [116] is used to find the cosine similarity between descriptors found in the database item and target-query. The neural network uses pre-trained vectors trained on part of the Google News dataset (about 100 billion words). The model contains 300-dimensional vectors for 3 million words and phrases. The phrases were obtained using a simple data-driven approach described in [116].

Finally, the cosine similarity between word-vectors of the two arrays is found. The similarity function then applies with no asymmetric reward.

4.3.3.3.5 Recommender results

Results are ranked in descending order according to total similarity. The most similar item is forwarded to the Queue Manager and finally to the AmI Speaker director to be displayed to the user. The recommendation process takes place at run time when a recommended auditory display is requested


```json
{
    "descriptors": ["rain","on","car","heavy"],
    "representation_type": "Auditory icon",
    "stream_url": "https://www.amiaudiolibrary.com/play?id=5d9c9e2642810fc3b4f9d007",
    "sound_category": "Weather",
    "info_type": "Predictive",
    "representation_similarity": 1,
    "sound_category_similarity": 1,
    "info_type_similarity" : 0.5,
    "descriptors_similarity" : 0.5430003680640484,
    "total_similarity" : 0.7607500920160121
}
```


Figure 31 - Recommender: Result example

for a user by an AmI application. Note that a few milliseconds are needed for the calculation of recommendations and that a history of results is kept in memory to accelerate performance. Results have a format similar to Figure 31. The Queue Manager evaluates the right time and place the display for the auditory display playback.

4.3.3.3.6 Result critiquing

The refinement of the case-based recommender results works by using the technique of critiquing (discussed at 3.2.4.2.1 Designing similarity metrics). Using the different modalities provided by each AmI application, the receiver of the auditory display (AmI environment user) can critique the recommended auditory display. The critique is then used by the recommender to adjust the user's profile and therefore the produced auditory display. Usually, a list of recommended auditory display is presented to the user for further critiquing. If a user likes a recommended auditory display, then the respective case in the case history is adjusted.

4.3.3.4 Queue Manager

The queue manager component evaluates queues and directs playback requests to the Speaker Director. For every location in an AmI environment, a queue to hold incoming requests is initiated. Queues are evaluated according to requests' urgency. Effectively, that means that displays with higher urgency will precede those of lower urgency in the order of play. In the case of the emergency – urgency level, all current active displays will be superseded.

Since multiple AmI applications may request different auditory displays for different users and locations at a single point in time, the Queue Manager ensures that those are handled appropriately and sent to the Speaker Director in an urgency-specific order. This, in combination with Speaker Director's evaluation of contextual information, prevents the scenario of a user getting overwhelmed and confused by auditory displays. Therefore, auditory displays are presented according to applications demands and according to users' activities and busyness of the auditory channel.

Chapter 5
5. ACOUSMA: Evaluation

Both a heuristic and a user-based usability evaluation were planned for all modules of ACOUSMA. The heuristic evaluation was designed with the purpose of acting as a first indication of ACOUSMA's usability in addition to detecting major usability issues before running the user-based evaluation Most of the usability problems detected by the heuristic evaluation were resolved before the user-based evaluation. A case study was shaped to base the user-based evaluation. A weather application for the smart home acted as the central axle under which different use cases and tasks were considered for each user. In the following section, the qualitative, heuristic usability evaluation will be reported along with its results. Then, the user based will be reported and results will be analyzed.

5.1 Heuristic Evaluation

Heuristic evaluation is an informal usability inspection method that aims in discovering usability problems [117]. The process involves having a small number of experts judge whether interaction elements of an interface follow established usability principles (heuristics). The final output is a list of discovered usability problems with references to the heuristics that were violated. For this heuristic evaluation, the following ten well established heuristics by Jacob Nielsen [118] were used:

1. Visibility of system status
2. Match between system and the real world
3. User control and freedom
4. Consistency and standards
5. Error prevention
6. Recognition rather than recall
7. Flexibility and efficiency of use

8. Aesthetic and minimalist design
9. Help users recognize, diagnose, and recover from errors
10. Help and documentation

The following 0 to 4 rating scale was used as suggested in [119] to rate the severity of usability problems:

0 = I don't agree that this is a usability problem at all

1 = **Cosmetic problem only**: need not be fixed unless extra time is available on project

2 = **Minor usability problem**: fixing this should be given low priority

3 = **Major usability problem**: important to fix, so should be given high priority

4 = **Usability catastrophe**: imperative to fix this before product can be released

Two interfaces were examined for usability issues, the AmI Audio Library Explorer and AmI Speakers Explorer. In order to detect the highest possible number of usability problems, six evaluators were called to participate, all experts in the field of HCI.

5.1.1 Method

The evaluation was executed in two phases. During the first phase, a short presentation of ACOUSMA was given to participants to acquaint them with the terminology and purpose of the graphical user interfaces they were called to evaluate. This was followed by one-to-one meetings with the evaluators where usability problems were detected and recorded. The setup used by evaluators was a desktop computer and a mobile phone. Evaluators were given generic tasks that aimed to give a sense of the features offered by each interface. They could try and complete them in any order. Specific tasks were not given in order not to limit exploration and possible discovery of usability issues. For the second phase, indicated problems were aggregated. Then, evaluators rated the severity of each usability problem as well as marked the heuristic(s) that were being violated.

5.1.2 Results

A total of 62 usability issues were detected for the AmI Audio Library Explorer and 12 for the AmI Speakers Explorer interface. The next sections will analyze received severity ratings, heuristics violated and solution of major issues for both interfaces.

5.1.2.1 Severity ratings

Regarding the average severity ratings received for the AmI Audio Library explorer:

- 16 were marked as 3
- 36 were marked as 2
- 10 were marked as 1

The percentage per severity rating received is depicted in Figure 32.

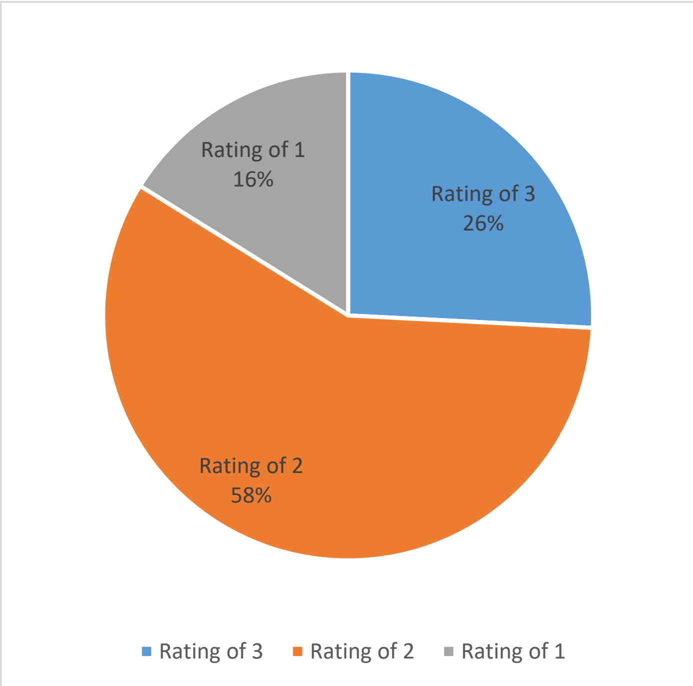


Figure 32 - AmI Audio Library Explorer: Severity ratings percentages

Regarding the average severity ratings received for the AmI Speakers explorer:

- 1 was marked as 3
- 7 were marked as 2
- 4 were marked as 1

Figure 33 shows the percentiles of received severity ratings for that interface.

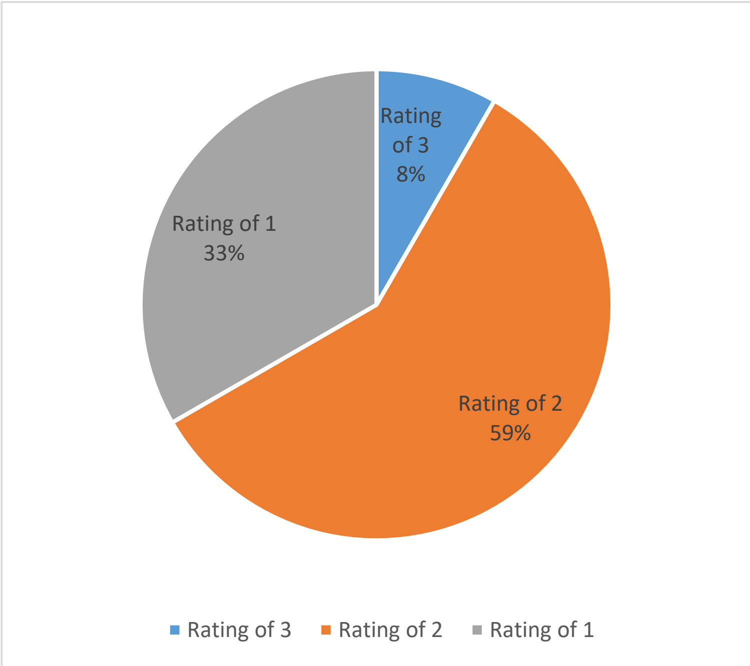


Figure 33 - AmI Speakers Explorer: Severity ratings percentages

Evident by Figure 32 and Figure 33, the majority of issues found referred to either a minor usability problem or a cosmetic problem. For the AmI Audio Library Explorer, only 26% of the detected usability problems were marked as a major usability problem while for the AmI Speakers Explorer only the 8%. Those issues were given the highest priority to be resolved.

5.1.2.2 Heuristics violated

For the AmI Audio Library, 62 usability issues were discovered. Figure 34 summarizes the amount of issues found per heuristic rule for the AmI Audio Library Explorer.

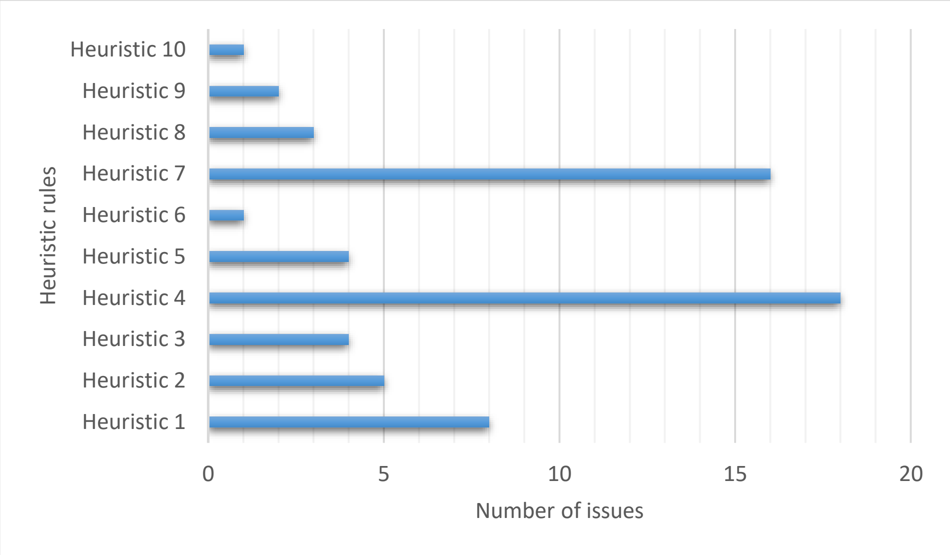


Figure 34 - AmI Audio Library Explorer: Number of issues per heuristic rule

Most of issues were violating heuristic "4. Consistency and standards", followed by "7. Flexibility and efficiency of use" and "1. Visibility of system status". Looking at major usability problems, a similar trend is followed as evident in Figure 35, with the addition of "5. Error prevention".

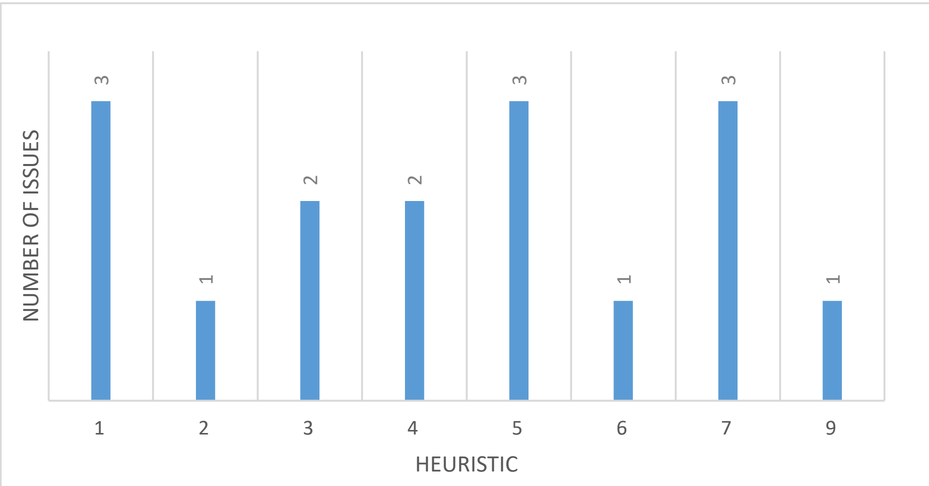


Figure 35 - AmI Audio Library Explorer: Major usability issues per heuristic

Regarding the AmI Speakers Explorer, 12 usability issues were discovered. Figure 36 summarizes the amount of issues that were found per usability principle.

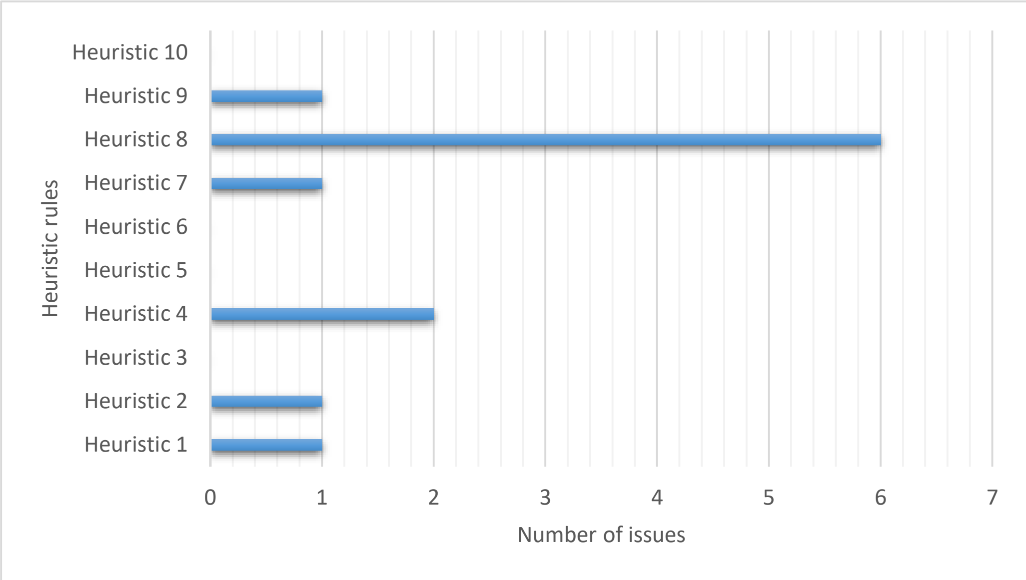


Figure 36 - AmI Speakers Explorer: Number of issues per heuristic rule

Here the most issues were violating the heuristic "8. Aesthetic and minimalist design" followed by "4. Consistency and standards". A single major usability issue was detected for this interface and violated the heuristic "7. Flexibility and efficiency of use".

The two sections that follow will look at which major usability were addressed and how. It must be noted that all usability issues, both major and minor, were addressed before the user-based evaluation took place in order to maximize usability of provided interfaces. The two sections that follow examine which major usability issues (severity rating of 3) were addressed and how for each interface.

5.1.2.3 Major usability issues: Audio Library Explorer

Major usability problems were detected for four pages of the Audio Library Explorer web application:

i. Home Page
ii. Create Page
iii. Upload Page

iv. TTS generator page

For each page, a table the with detected issues is presented along with two figures to visually map issues and solutions to specific components. A discussion of how such issues were resolved will follow.

5.1.2.3.1 Home page

Six major usability issues were detected for the home page. Table 5 numbers these issues as they were identified by evaluators. The table also shows which heuristic principle (rule) is violated and recommendations received by evaluators. Figure 37 shows UI elements corresponding to usability issues, while Figure 38 depicts how those issues were solved.

Issue 1 was solved according to recommendation for the mobile version of the application. For the desktop version, it was solved by removing the button and making the search element always visible on screen (also a solution to a minor usability issue). Issue 2 was solved by adding an element to select a preferred sorting option. Selecting the same sorting option twice switches between ascending and descending ordering. The default ordering is descending. A title was added to show the current page as seen in Figure 35 solving issue 3. Vertical scroll was changed to horizontal when multiple tags are present in the mobile view, solving issue no 4. As a solution to issue 5, uploaded representation appear on top when a user enters the system while using sorting options, a user can change that. Finally, multiple files are no longer allowed to play simultaneously in the home page. Selecting to start playback of an auditory display automatically stops the last one playing, solving issue 6.

Table 5 - Major usability issues: Home page

i. Home page				
No	**Component**	**Issue**	**Violates Principle**	**Recommendation**
1	HEADER→ SEARCH BUTTON	Links to home page	4	*Change button link*
2	HOME LANDING PAGE	There are no ordering, sorting options for items	7	*Add sorting & ordering options*
3	PAGE	No written indication of current page	1	*Adding a title of current page*
4	CARD→ TAGS→ SCROLL	Vertical scroll is confusing (multiple tags on mobile)	3	*Change to vertical scroll, Add a "+3" button*
5	HOME LANDING PAGE	My uploaded representations not visible	1	*User uploaded representations should be visible at home page*
6	CARD→ AUDIO PLAYER	Allowing multiple files to play simultaneously	5	*Stop previous playback on play button click*

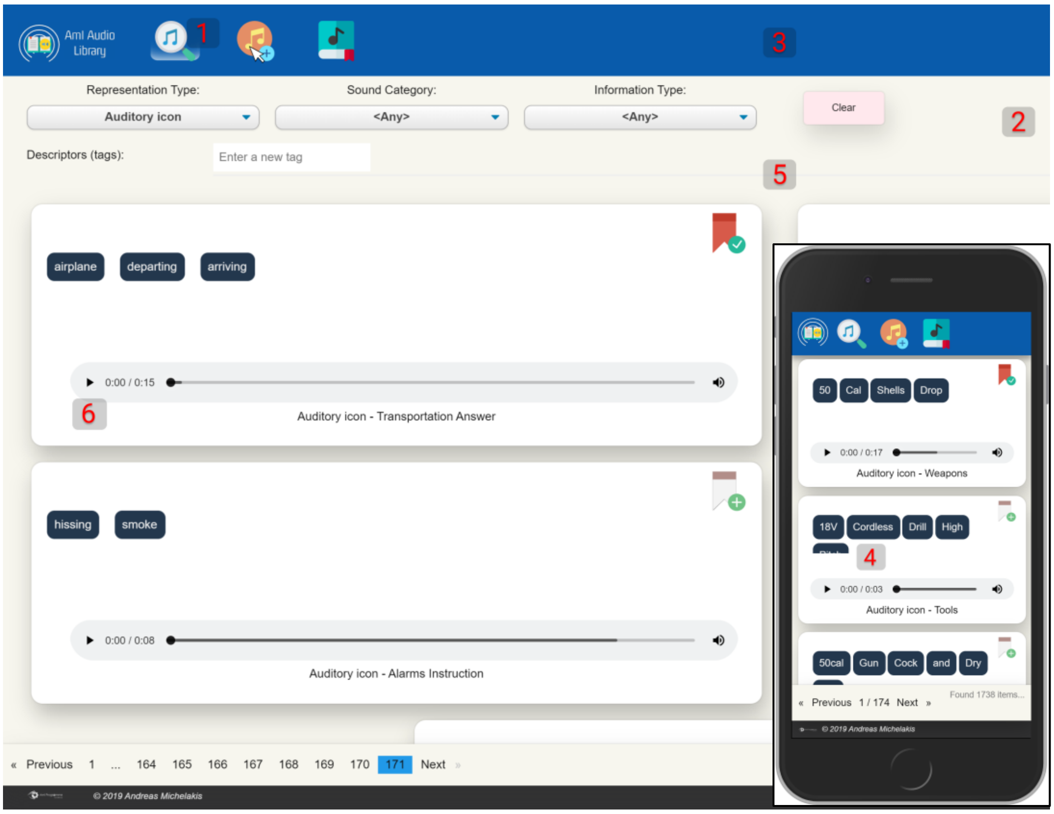


Figure 37 - Major usability issues: Home Page

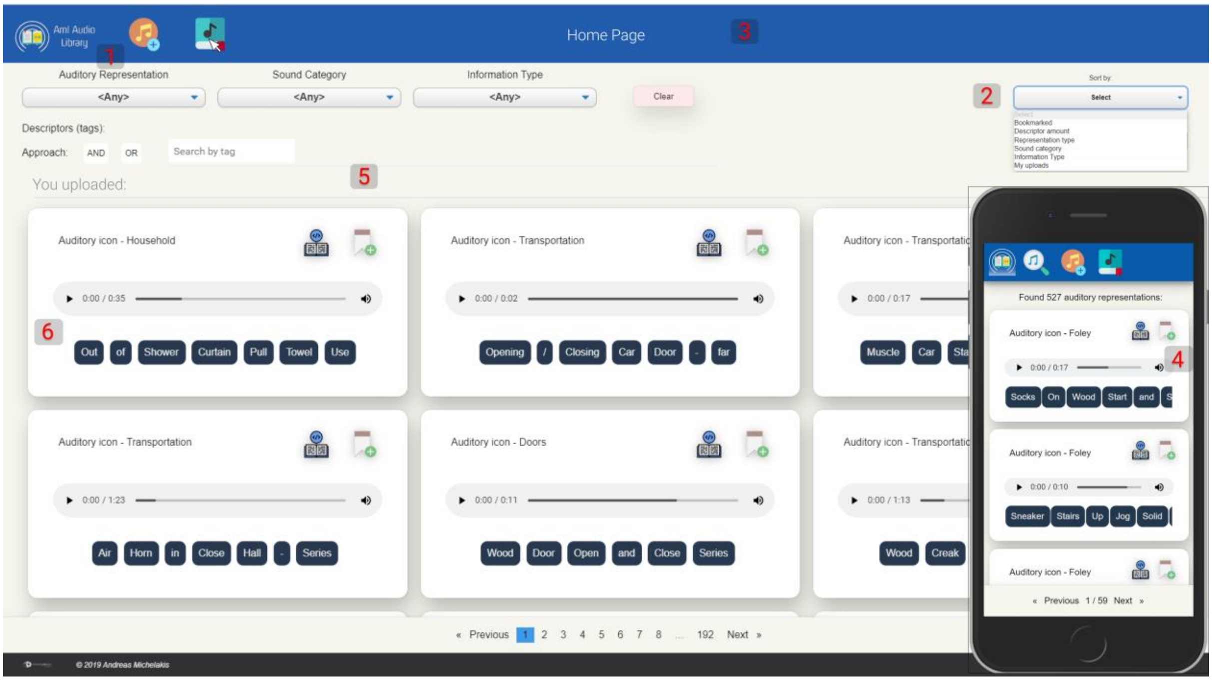


Figure 38 - Solutions: Home page

5.1.2.3.2 Create page

Only one major usability issue was detected for the Create page. Users could not see their uploaded auditory representations. It was solved by displaying uploaded items in a similar manner as in the home page.

Table 6 - Major usability issues: Create page

ii. Create page				
No	**Component**	**Issue**	**Violates Principle**	**Recommendation**
1	PAGE	Cannot see previously uploaded/created representations	1	*Show previously uploaded representations here*

Figure 40 - Major usability issues: Create page

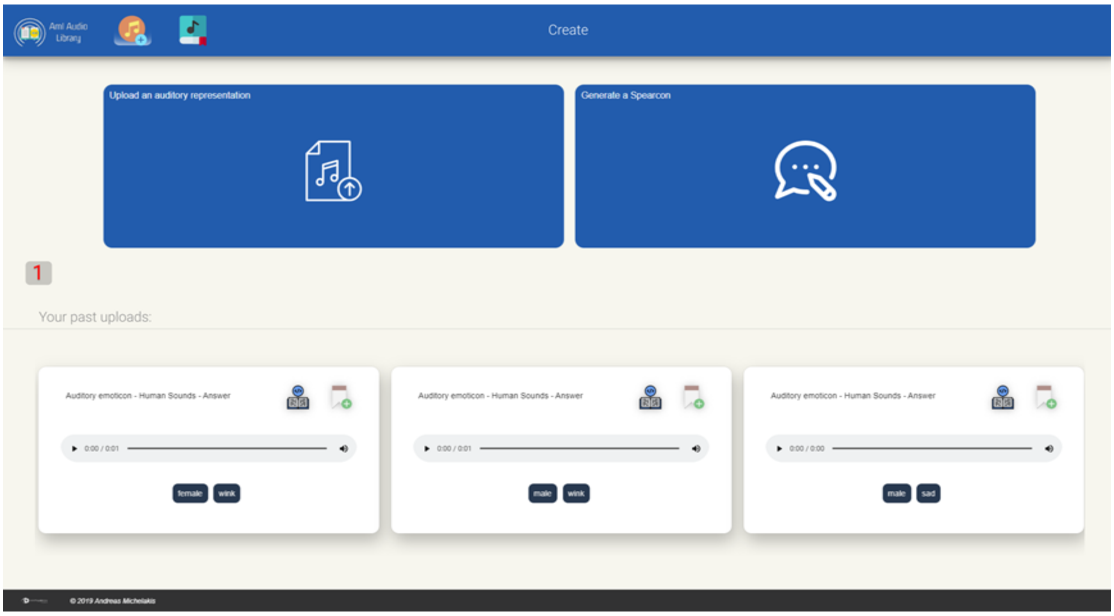


Figure 39 - Solutions: Create Page

5.1.2.3.3 Upload page

Four major usability issues were detected for the Upload page. All issues were solved according to recommendations. Issue 1 was solved by accepting only audio file formats, issue 2 was solved by adding a remove button, issue 3 by expanding the clickable area, and issue 4 by marking missing fields with a red border color.

Table 7 - Major usability issues: Upload page

iii. Upload page				
No	**Component**	**Issue**	**Violates Principle**	**Recommendation**
1	FORM →FILE SELECTION	Files other than audio are allowed (e.g. .pdf)	5	*Disallow selection and upload of such files*
2	SUCCESSFUL UPLOAD	Remove button not available (undo)	3	*Add remove button*
3	FORM → FILE SELECTION	Clicking to upload constrained on text	4	*Expand clickable area to fill box*
4	FORM → ERROR MESSAGE	No additional indication to the specific problem/missing field	9	*Mark fields with red border*

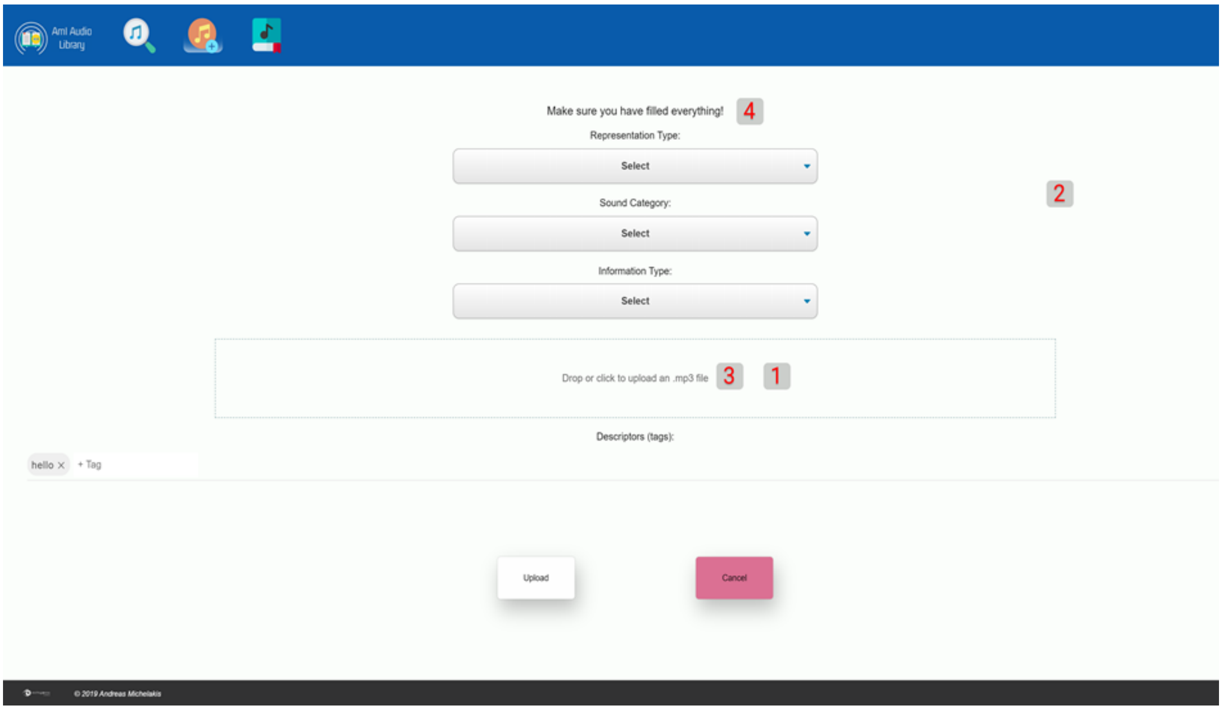


Figure 41 - Major usability issues: Upload page

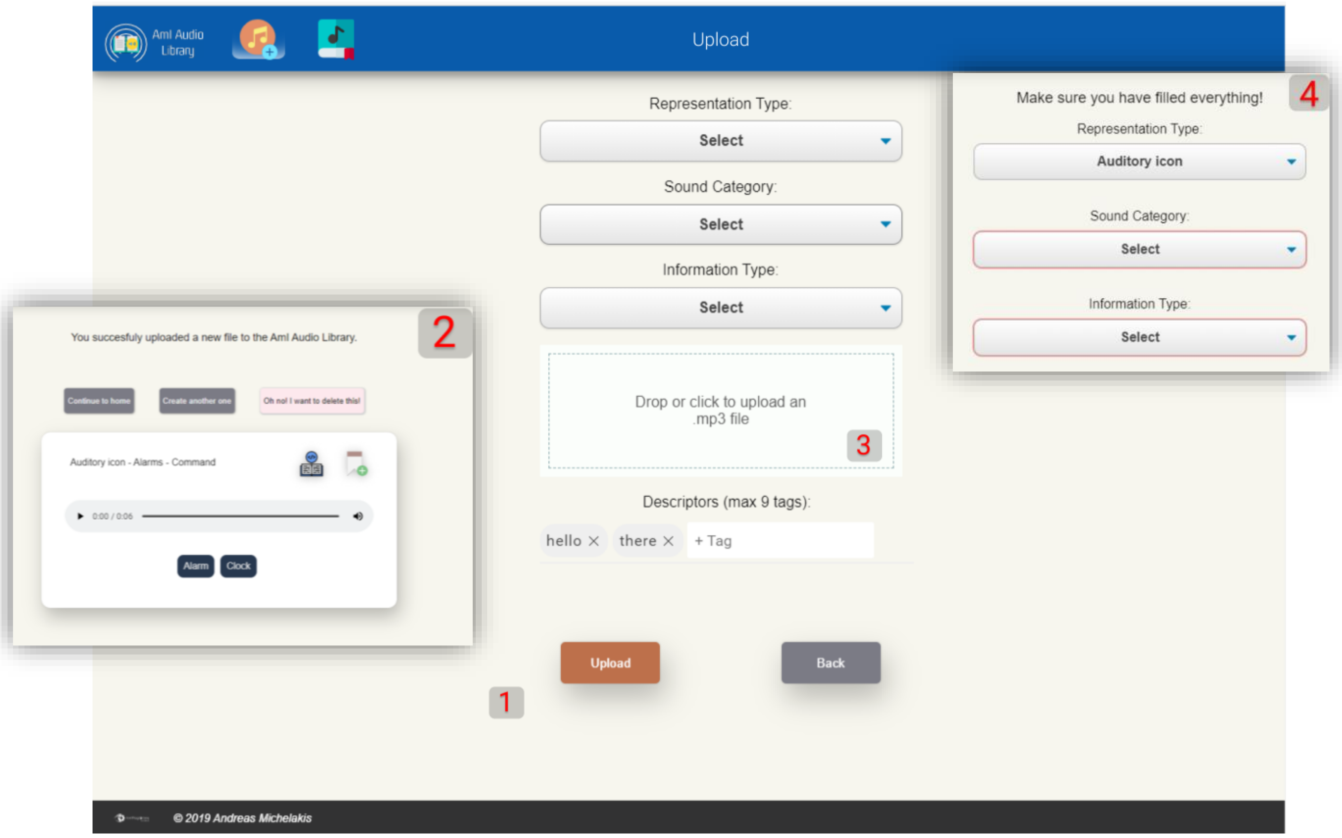


Figure 42 - Solutions: Upload Page

5.1.2.3.4 TTS generator page

Six detected usability issues received an average rating of three for the home page. Table 8 numbers these issues as they were identified by evaluators. The table also shows which heuristic principle (rule) is violated and the recommendations received from the evaluators.

Table 8 - Major usability issues: TTS generator page

iv. TTS generator page				
No	**Component**	**Issue**	**Violates Principle**	**Recommendation**
1	GENERATE	Same text-language combination is allowed for multiple generation (duplicates allowed)	5	*Restrict duplicates*
2	RESULTS	Invisible due to no indication for newly generated result (occurs after second generation)	7	*Scroll to Result*
3	SINGLE RESULT	Cannot directly upload to DB	7	*Add direct upload button*
4	DROPDOWN→ LANGUAGE	Language title does not describe content well	2	*Should be changed to 'Accent', since voices can read any language*
5	TEXT-BOX	No indication for maximum characters allowed (currently allowed 250)	6	*Indicate and restric maximum input characters*

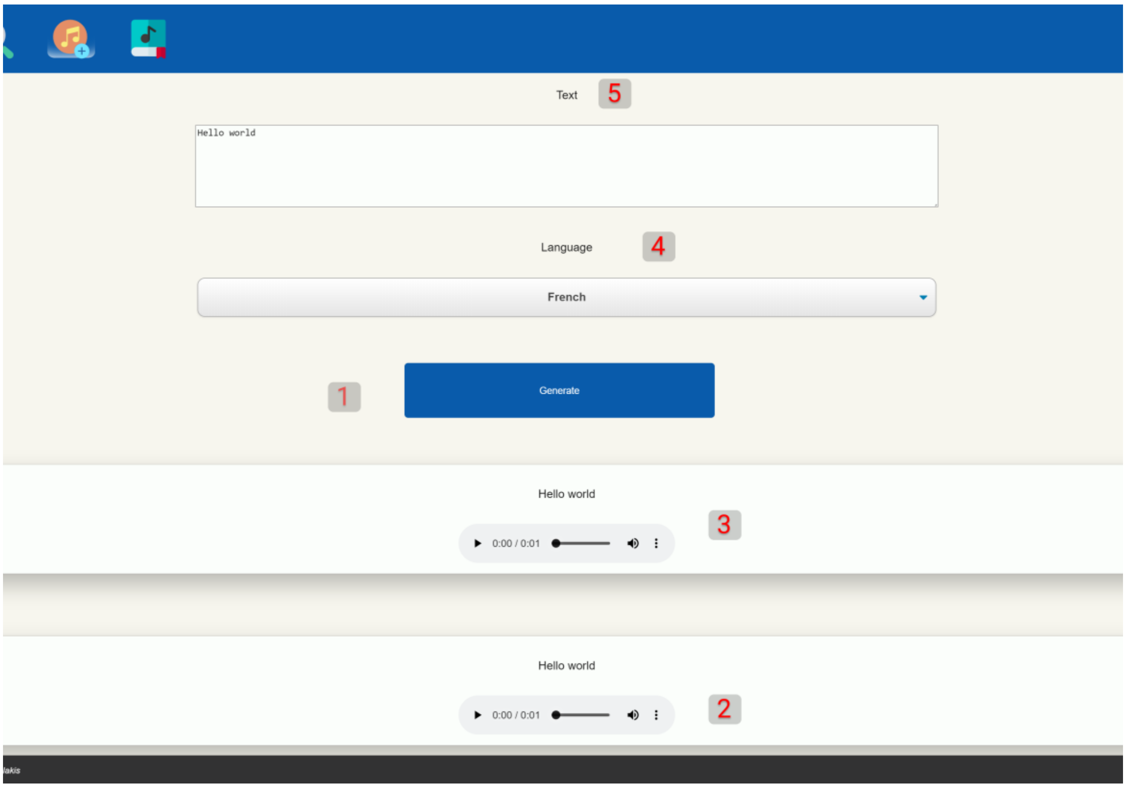


Figure 44 - Major usability issues: TTS generator page

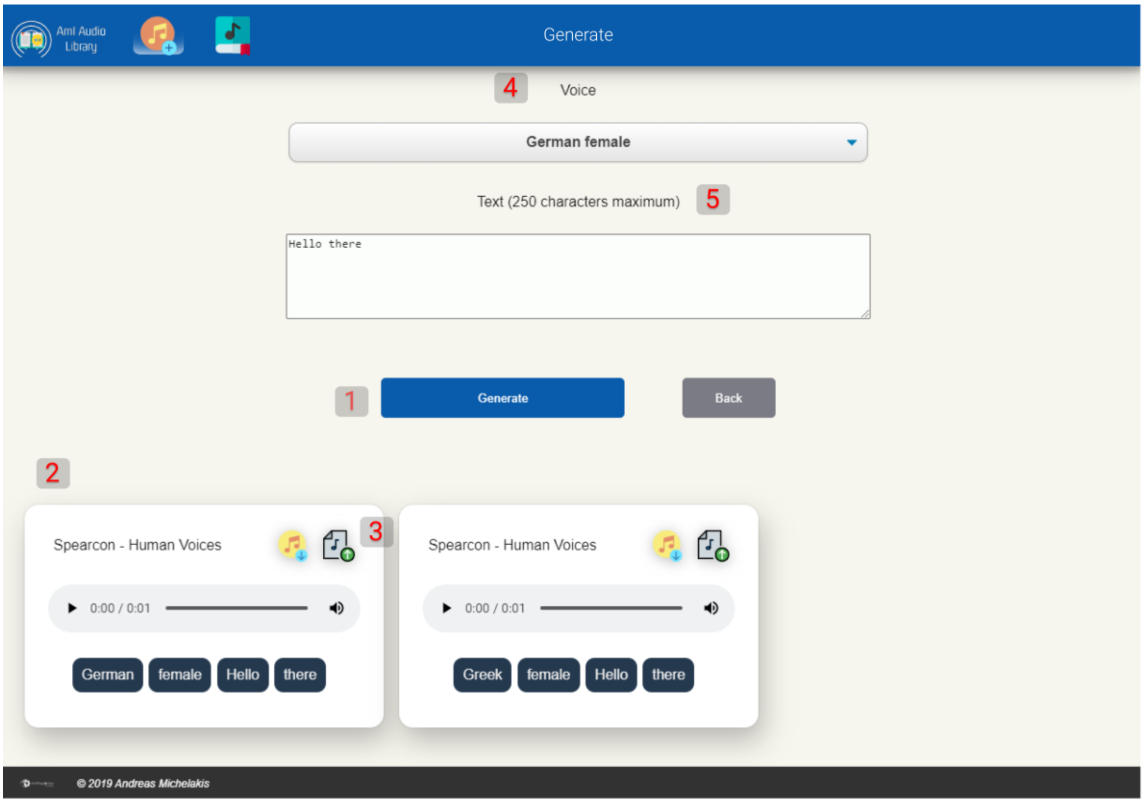


Figure 43 - Solutions: TTS generator page

5.1.2.4 Major usability issues: Speakers explorer

The Speakers Explorer interface only suffered from one major usability issue. No search and sorting functions were implemented. To solve that issue, users can now use interface elements to sort displayed information by a number of different sorting functions as well as search for a specific location or speaker by simply typing into the search bar.

Table 9 - Major usability issues: Speakers Explorer

Speakers explorer page				
No	**Component**	**Issue**	**Violates Principle**	**Recommendation**
1	ALL INFORMATION	Search and sorting functions are unavailable	7	*Add search and sort functions for locations and devices*

Figure 45 - Major usability issues: AmI Speakers Explorer

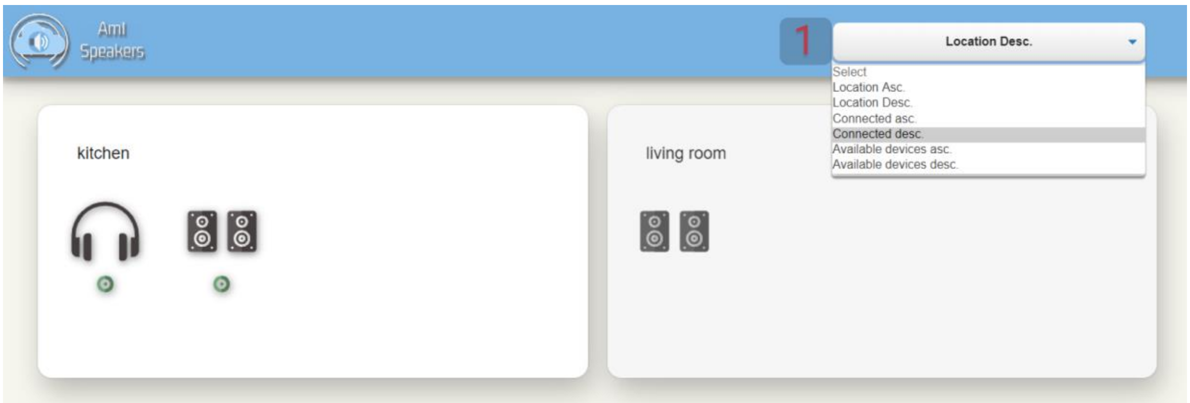


Figure 46 - Solutions: Speakers Explorer

5.2 Case study

A case study was prepared as the basis for the user-based evaluation. Testing the complete workflow of ACOUSMA required following the various user types and their respective tasks in the process of integrating auditory displays in an AmI application. The selected case study was to follow the creation of a weather application for an AmI environment. For every user type, specific tasks were formed that had to be followed by participants. A total of seventeen people agreed to participate in the evaluation. In the following subsections, the specific method, use cases and tasks will be presented, as well as results.

5.2.1 Method

ACOUSMA is designed for three target user types. The following table summarizes which interface is used by which type of user:

Table 10 - Type of user per user interface

Type of user	Interface
Auditory display expert	AmI Audio Library Explorer, AmI Speakers Explorer
AmI application developer	API client library module
AmI environment inhabitant	AmI application with ACOUSMA features

For each type of user – interface combination, a use case was composed and participants were given specific tasks to complete. Participants were carefully selected so that their real-life occupation approximates the role assigned. For example, it would be impossible to evaluate tasks for AmI application developer interfaces if a participant had no prior knowledge of programming. Specifically, seven people were assigned to the role of the auditory display expert, six people were assigned to the role of the AmI environment inhabitant and four people to the role of AmI application developer. All participants received a full brief regarding their role and equipment before running given tasks for the evaluation.

Consider the following scenario:

“A construction company composed by architects, designers and programmers, builds apartments based on Ambient Intelligence (AmI) technologies. With each purchased apartment, the company offers an interactive platform through which residents can install AmI applications. Those applications utilize shared resources and services provided by the apartment and display useful information in a variety of methods (audio, video) facilitating and improving the day-to-day lives of users. The company is currently utilizing a team of Auditory Display experts and AmI application developers to create weather forecast application. The application, using audio, will promptly inform a user of the weather conditions at the location of a scheduled event on their calendar. For this purpose, the company uses the ACOUSMA system. ACOUSMA is a system that manages the prompt deployment of personalized auditory displays in AmI environments. Moreover, the company has a simulation space (simulating apartment rooms) to test applications before release. When complete, the application will be available for use through the apartments’ interactive platform.”

Three use cases derive from this scenario:

- An auditory display expert using ACOUSMA to generate and upload auditory representations for various weather conditions. Also, the expert using ACOUSMA to test his design upon the simulation space.
- An AmI application developer using the API client library module to initiate user-tailored auditory displays for various weather conditions.
- An AmI environment inhabitant (apartment resident), experiencing the auditory result of the weather forecast application with the ability to critique.

Those use cases were the basis to formulate precise tasks for each user type. Each participant had to complete a number of tasks based on their role. Moreover, each role received an adjusted version of the aforementioned scenario to match their role. In the next section, the specific method and tools followed for each role will be analyzed.

5.2.1.1 Setup

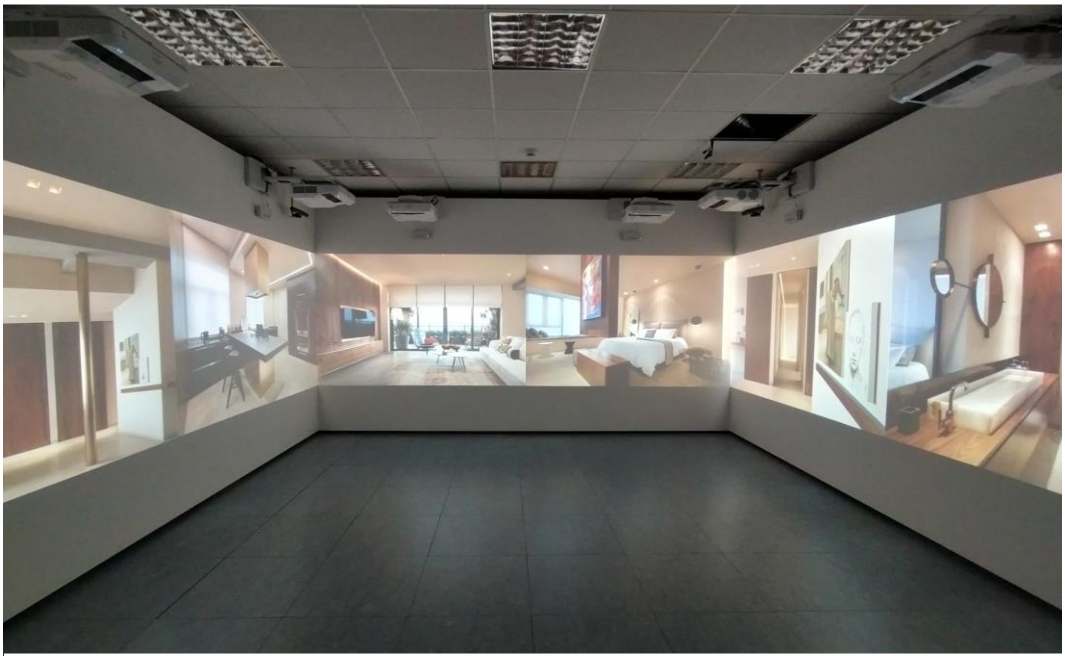


Figure 47 - "White room" simulation space

In order to reenact the process of Auditory Display Experts and AmI application developers working on a weather forecast application using ACOUSMA and the constructing company's apartment simulation space, a room (see Figure 47) with AmI infrastructure and services was utilized at the premises of the ICS-FORTH Ambient Intelligence Programme [120]. The room, named "white room", was used to offer an immersive experience for participants. The space offers a position tracking system and service, six projectors with touch capabilities as well as stereo speakers on each projector. For the purposes of this evaluation, AmI Audio Clients were set up to enable deployment of auditory displays to each projector. In addition to that, the available position tracking service was used to trigger events during the evaluation tasks for participants with the AmI environment inhabitant role. The infrastructure can be seen on Figure 48.

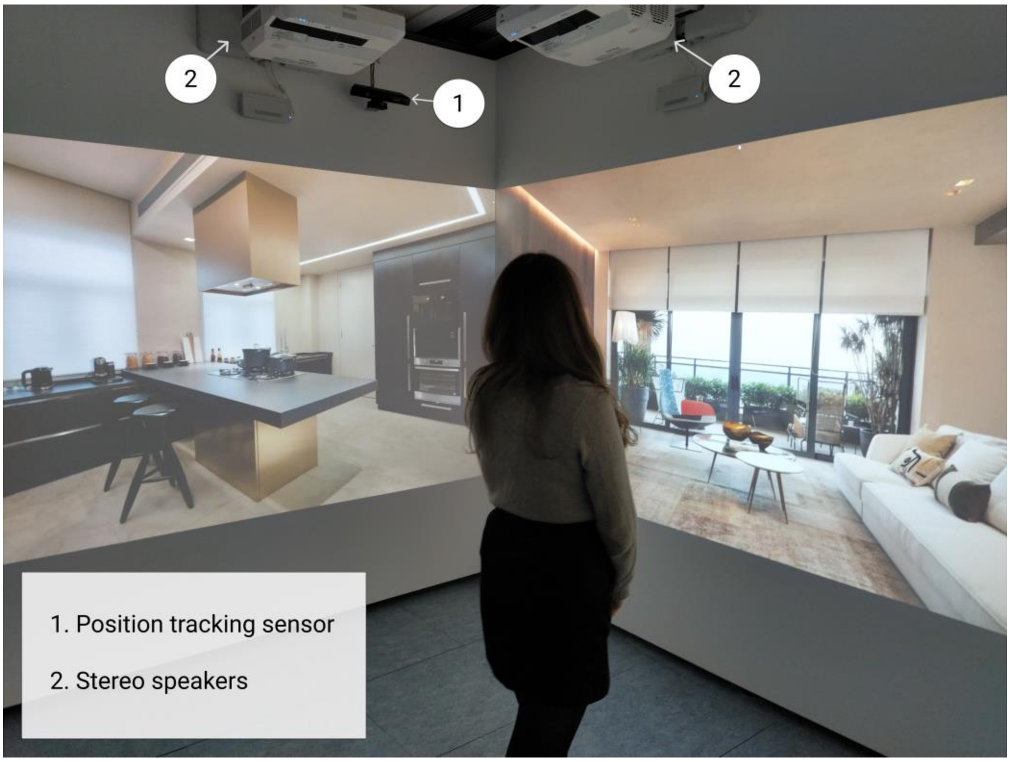


Figure 49 - "White room" simulation space: infrastructure

For interacting with ACOUSMA's interfaces, participants used a desktop computer with a web-browser as seen on Figure 48. The computer was

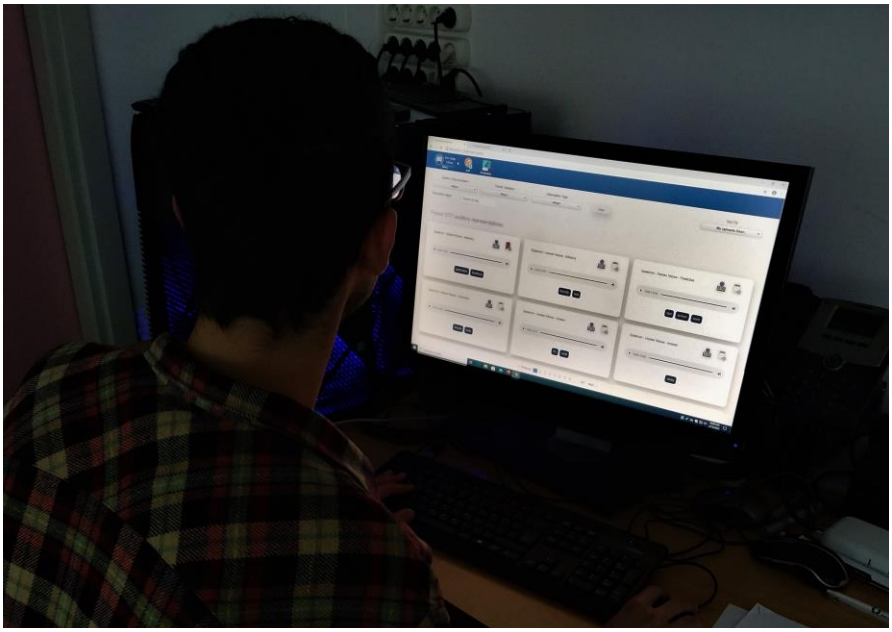


Figure 48 - Using the AmI Audio Library Explorer

placed in the "white room" so that participants could deploy and listen to auditory displays at the simulation space.

5.2.1.2 Auditory display expert

The auditory display expert uses ACOUSMA's AmI Audio Library Explorer interface to generate, upload and share created auditory representations with their team. Moreover, this type of user may choose to test their designed representations over specific locations of an AmI environment. For the latter, the AmI Speakers Explorer is used. The user-based evaluation aimed in examining usability of those two interfaces. The evaluation followed the think-aloud method [121] encouraging participants to openly share anything they thought during their interaction with ACOUSMA's interfaces.

Following the use case scenario of the weather forecast application, participants were given a role-adjusted version of the original scenario, mentioning that they are a member of that constructing company and that their team was assigned to design and develop the weather forecast application:

"*You are a member of a construction company where architects, designers…The company has commissioned your team to develop a weather forecast application. …"*

Along with that, a specific description of their role was given:

"As an auditory display expert, you have been given the task to design auditory representations describing a list of weather conditions. Using an interface provided by ACOUSMA, you will generate and upload auditory representations. Moreover, using a test interface of ACOUSMA, you will run listen at the company's simulation space, some of the auditory representations you have designed and uploaded."

This introduction was done so that participants had a full understanding of their role as well as to create a feeling of being in-context while completing tasks with each interface. Specific tasks were formed to evaluate usability for each interface.

For the AmI Audio Library Explorer, its four basic functions had to be evaluated: (1) generating, (2) uploading, (3) searching and (4) bookmarking auditory representations. For that purpose, participants had to complete the following tasks:

A. Adding auditory representations

1. Generate a Spearcon to describe the weather condition “sunny” using the English (UK) female voice.
2. After you generated that Spearcon, use the option for direct upload to upload the auditory representation to the database. Assume the information type is Predictive.
3. Repeat the same steps (1&2) to generate and upload Spearcons describing the following:
    a. Thunderstorm
    b. Heavy rain
    c. Snow

Using other sonification tools, you have exported two audio files that are now located at your desktop:

4. Upload the first file, named thunderstorm.mp3 to the database. Assume that the file is an Auditory Icon of sound category Weather and information type Predictive.
5. Upload the second file, named heavy_rain.mp3 to the database. Assume that the file is an Auditory Icon of sound category Weather and information type Predictive.

B. Searching & Bookmarking

6. Search for the Auditory Icon you uploaded at the previous task (5.) and add it to your bookmarks.
7. Search for the Spearcon you generated uploaded at tasks 1 & 2 and add it to your bookmarks.
8. Search for an Auditory Icon that a member of your team has uploaded and that describes “strong wind”.
9. From your bookmarks find the Auditory Icon describing “heavy rain”. View the code snippet and copy the item’s identity number (ID).

For the AmI Speakers Explorer, its two basic functions had to be evaluated: (1) testing an auditory representation and (2) monitoring AmI Audio Clients. For that purpose, participants had to complete two additional tasks:

C. Testing at the simulation space

10. Using the ID you just copied, test the auditory representation at the speakers located at room “kitchen” of the simulation space.
11. Stop the test you just initiated.

Following a direct observation approach, while participants were completing tasks, the evaluator was keeping detailed notes of participants’ comments as well as completion time and completion rate for tasks. After completing all tasks, participants were asked to fill out the System Usability Scale (SUS) (Table 12) questionnaire [122] [123], as well as a number of additional questions (Table 11). The SUS was selected as it can yield reliable results on small sample sizes and is an established tool for differentiating between a usable and unusable interface. Additional questions had a form similar to SUS and aimed at gathering statements regarding user’s perception of the various interfaces functions, generating, uploading & searching for the AmI Audio Library Explorer; monitoring and testing for the AmI Speakers Explorer interface.

Table 11 - Additional questionnaire

	Strongly Disagree	Somewhat Disagree	Neutral	Somewhat Agree	Strongly Agree
1. **It was easy to generate and upload Spearcons.**	☐	☐	☐	☐	☐
2. **I had no problem finding specific representations**	☐	☐	☐	☐	☐
3. **It was easy to test a representation**	☐	☐	☐	☐	☐
4. **It was hard to find and monitor locations and speakers**	☐	☐	☐	☐	☐

Table 12 - J.Brooke's system usability scale (SUS) [126]

		Strongly Disagree	Somewhat Disagree	Neutral	Somewhat Agree	Strongly Agree
1.	**I think I would like to use this tool frequently.**	☐	☐	☐	☐	☐
2.	**I found the tool unnecessarily complex.**	☐	☐	☐	☐	☐
3.	**I thought the tool was easy to use.**	☐	☐	☐	☐	☐
4.	**I think that I would need the support of a technical person to be able to use this system.**	☐	☐	☐	☐	☐
5.	**I found the various functions in this tool were well integrated.**	☐	☐	☐	☐	☐
6.	**I thought there was too much inconsistency in this tool.**	☐	☐	☐	☐	☐
7.	**I would imagine that most people would learn to use this tool very quickly.**	☐	☐	☐	☐	☐
8.	**I found the tool very cumbersome to use.**	☐	☐	☐	☐	☐
9.	**I felt very confident using the tool.**	☐	☐	☐	☐	☐
10.	**I needed to learn a lot of things before I could get going with this tool.**	☐	☐	☐	☐	☐

5.2.1.3 AmI application developer

AmI application developers use the API client library module and its functions to integrate auditory displays into their applications. The aim of the user-based evaluation for this user-type was to gather comments regarding documentation and overall usability of the API. Participants were given a template project of a web-application written in the programming language typescript. Participants were then asked to complete certain tasks at their own pace and time. That meant that participants used their preferred IDE to load the project and complete tasks. Moreover, participants had to setup an AmI Audio Client. When all tasks were finished, participants were interviewed individually and filled the SUS.

Following the use case scenario of the weather forecast application, participants were introduced to their role via a role-adjusted version of the original scenario:

"*You are a member of a construction company where architects, designers…The company has commissioned your team to develop a weather forecast application.…*"

And a specific description of their role:

"*As a programmer, you were given the task to create a testing prototype for the weather forecast application. The prototype should test functionality of the ACOUSMA system upon the simulation space.*"

Tasks given were centered around the idea of filling functionality of the template project. The interface was provided as seen on Figure 50 and participants had to program corresponding functions. A sheet was given to each participant that included several examples of usage for the API as well as the following tasks:

A. Program a function that would be bound to the 'Get Recommendation' button of the interface:
    1. Create a function that would use an appropriate recommender method from the ACOUSMA API. The function should return a recommended sound for the selected user (selectedUser) and the selected weather forecast descriptors (selectedWeatherDescriptors)
    2. Use the helper method 'log' to display the API's response on the interface.
    3. Use the helper method 'setAudioStream' to make the recommended sound audible at the interface.

4. Bind the function you created to the appropriate <button> element on the interface.
5. Check functionality of the button and read/listen to the response for available user profiles.

B. Program a function that would be bound to the 'Play at user's location' button of the interface:

6. Create a function that would use an appropriate recommender method from the ACOUSMA API to retrieve a sound recommendation. The function should play a recommended sound to the selected user's (selectedUser) location describing the selected weather forecast descriptors (selectedWeatherDescriptors)
7. Use the helper method 'log' to display the API's response on the interface.
8. Use the helper method 'setAudioStream' to make the recommended sound audible at the interface
9. Bind the function you created to the appropriate <button> element on the interface.

C. Program a function that would be bound to the 'Stop playback' button of the interface:

10. Create a function that would use an appropriate recommender method from the ACOUSMA API. The function should stop playback of a recommended sound to the selected user's (selectedUser) location.
11. Use the helper method 'log' to display the API's response on the interface.
12. Bind the function you created to the appropriate <button> element on the interface.

D. Use the function addUser of the resident service, to add a new test resident profile. The username should be your name and the location, the location at which you set up the AmI Audio Client i.e. this.residentService.addUser('Yourname','yourname_home')

13. Using the interface select the test resident profile with your name as well as a weather forecast description.
14. Test the functionality of all buttons of the interface. Sound should be audible at the AmI Audio Client location.

Individual interviews were carried out after completion of tasks by participants. First, the original SUS was filled and then several open-ended questions were asked to gather comments about the usability of the API as well as recommendations for possible improvements:

- Did you think functions were well documented?
- Did you have trouble setting the input for ACOUSMA's functions?
- What do you think should change?

Usability ratings collected by using the SUS would not be used as a validation of the API's usability, but as a guide towards usability areas that the API may be prone to improvements. Finally, time needed to complete all tasks was asked and whether any of the tasks were difficult and therefore skipped.

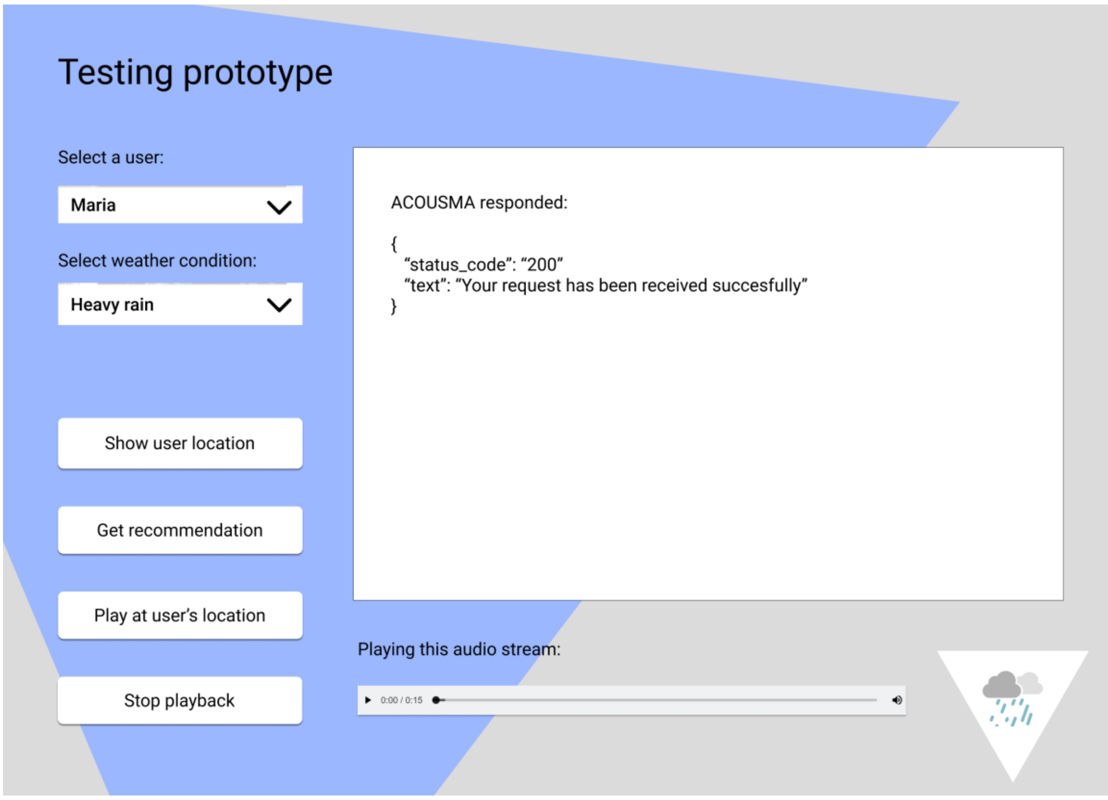


Figure 50 - User interface of the testing prototype

5.2.1.4 AmI environment inhabitant

An AmI environment inhabitant may listen to auditory displays emerging from various AmI applications. ACOUSMA will prioritize and initiate personalized auditory displays according to environment and user context. The purpose of the user-based evaluation for this role was to measure the effectiveness of the Auditory Display Recommender module as well as to measure the perceived usability of an application enhanced with ACOUSMA's features. To that end, for every participant a blank preference profile was initiated in the recommender, letting the participant dynamically shape their profile through interaction with the weather forecast application and ACOUSMA.

Following the use case scenario of the weather forecast application, participants were introduced to their role via a role-adjusted version of the original scenario:

"*You have bought an apartment from a construction company where…Recently, you installed a weather forecast application that promises to inform you, using audio, about the weather conditions at the location of your next calendar meeting. This happens promptly, when the apartment recognizes you are about to leave for the meeting.*"

The specific description of their role:

"*As the resident of a smart apartment, you will follow certain everyday scenarios where audio will be used to inform you about the weather forecast. This will occur when you are about to depart for a scheduled meeting. You will then have to use a provided interface to state your preferences regarding sounds used.*"

Five scenarios were used to represent tasks. According to each scenario, the participant would trigger events at the simulation space, initiating characteristic soundscapes. Soundscapes were created to simulate real-life acoustic conditions for scenarios. The weather forecast application would inform the participant about weather conditions, promptly, according to each scenario.

The general workflow of the evaluation was as follows:

1. First, a scenario was read to participants.
2. Then, they would interact with the simulation space, moving according to the scenario and listening to sounds.
3. After the scenario, a single question was asked by the evaluator. The question regarded to the participant's understanding of the weather forecast auditory display.

5.2.1.4.1 Scenario A.

A. Monday: Preparing for work

   It is Monday and you are preparing breakfast at the apartment's kitchen. Soon, your taxi will arrive to take you to your workplace. Just when you start eating, the apartment's virtual assistant informs you that your taxi is going to arrive soon to take you to work.

- You listen to a sound informing you about the weather forecast.

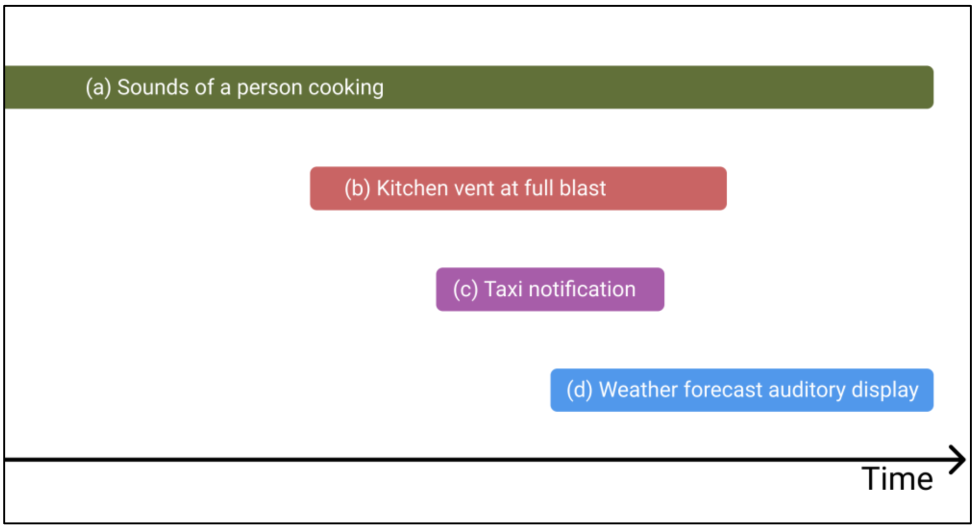


Figure 51 - Kitchen soundscape

For the first scenario, the participant would move and stand near the simulated kitchen room listening to various everyday sounds that normally would occur when a person prepares breakfast. Then, a notification would play and the virtual assistant (using text-to-speech) would inform the resident about their taxi arriving soon; saying "Your taxi will be arriving in 10 minutes. Be sure to get ready". Right after that, the weather forecast application would use ACOUSMA to play a recommended auditory display for the "heavy rain" weather condition. The succession of sounds played in this scenario are depicted in Figure 48. With the completion of the first scenario participants were asked about their understanding of the provided weather forecast auditory display. Participants were free to comment on their understanding of the display as well as other comments before starting the second scenario.

5.2.1.4.2 Scenario B.

B. Monday: Returning from work

> That same day, in the afternoon you come back from work. You quickly pass through the living room and towards the bedroom. You do not hear the notification sound that plays from your tablet at the living room. Thirty minutes later, wearing comfortable clothes, you sit at the living room couch

and ask the apartment's virtual assistance for updates. Amongst others, the assistant informs you that the weather forecast application requires your feedback about the sound it used that morning. The interface that the application provides can be accessed using the tablet.

- You use the tablet to state your preferences regarding the sound used by ACOUSMA.

The second scenario was prepared so that participants could use the critiquing mechanism of the Auditory Display Recommender to adjust the provided auditory display to their preferences. A simple critiquing interface was created for the weather forecast application, that participants could use to state the preferences (see Figure 49). Using that interface and a pair of headphones, participants listened to the auditory representation as originally selected by the recommender and then answered to a simple yes or no question about their preference. If they would choose "yes", their preference would be recorded, adjusting their profile and specific case. If they would choose "no", a simple form could be used to find a different auditory representation for the specific weather forecast condition i.e. "I would prefer a sound of the sound category 'Ambiences', containing the tags 'heavy', 'rain', 'summer'". According to user input, a list of fifteen recommendations was generated and a favorite could be chosen. The selected favorite, would then be stored as preferable for that case at the recommender's case history and user profile would be adjusted.

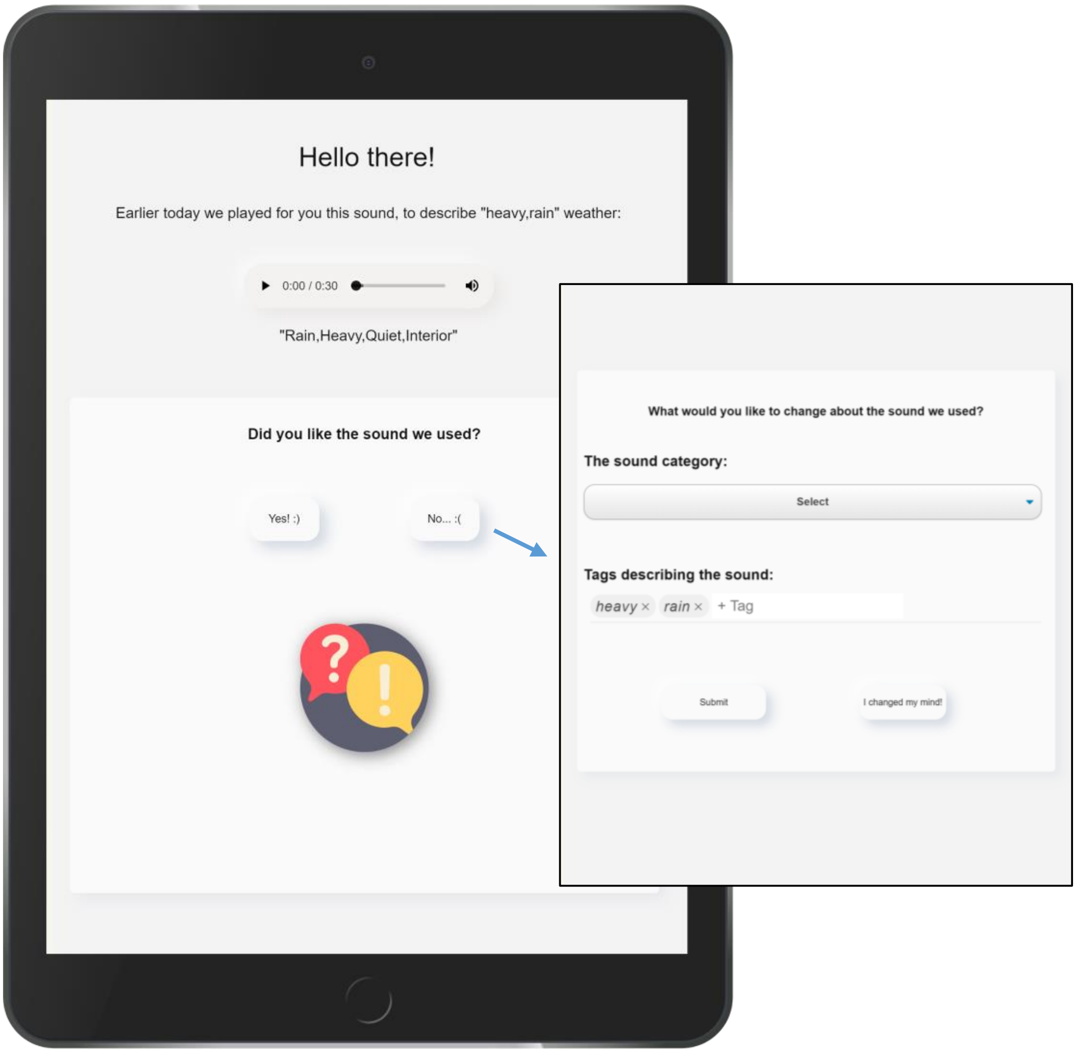


Figure 52 - Critiquing interface

5.2.1.4.3 Scenario C. & D.

The third and fourth scenarios were slightly altered versions of the first and second scenario. The major difference was that the weather forecast auditory display informed the resident about "sunny weather". This way, by the end of the fourth scenario, the participant would have stated (in total) their preference about auditory displays of two weather conditions "heavy rain" and "sunny weather".

5.2.1.4.4 Scenario E.

B. Weekend: An unexpected meeting

It is Sunday and you are not working. You do not have any plans for the day so you have decided to stay at the apartment. It is early afternoon and you are watching a documentary on your AmI TV. Suddenly, the volume drops and you are informed by the virtual assistant that your friend Anna has texted you. She wants to go biking at the nearby park.

- You listen to a sound informing you about the weather forecast.

During this last scenario, the resident would be informed about "heavy rain". This scenario was made so that the participant could listen to their personalized auditory display as it was adjusted during the second scenario. Then, it was measured whether the participant understood the personalized weather forecast auditory display.

5.2.1.4.5 Measurements

Aside from the three questions regarding understandability of the provided weather forecast auditory display (asked at the end of scenario A, C and E), participants' comments were also collected. Moreover, completing all scenario, participants were asked to fill the usability metric for user experience (UMUX) [124]. The UMUX questionnaire was chosen as it is an established tool for measuring perceived usability with high correlation to the SUS [125] [126]. As seen on Table 13, the UMUX consists of four statements with the ability to rate from one, stating strong disagreement, to seven, stating strong agreement. UMUX statements reflect to fundamental measures of user experience (based on ISO 9241-11) which are effectiveness, satisfaction, the efficiency and overall usability.

Table 13 - UMUX

		1	2	3	4	5	6	7
1.	**ACOUSMA's capabilities meet my requirements**	☐	☐	☐	☐	☐	☐	☐
2.	**Using ACOUSMA is a frustrating experience**	☐	☐	☐	☐	☐	☐	☐
3.	**ACOUSMA is easy to use**	☐	☐	☐	☐	☐	☐	☐
4.	**I have to spend too much time correcting things with ACOUSMA**	☐	☐	☐	☐	☐	☐	☐

5.2.2 Results

The user-based evaluation yielded positive evidence towards the usability of the ACOUSMA's interfaces. All participants could successfully complete all given tasks while their comments and overall interaction with the system provided useful insights for improving ACOUSMA. In the next sections, results for each user type will be presented and discussed.

5.2.2.1 Results: Auditory display expert

Based on participants' interaction and comments, as well as their answers to given questions and the SUS, usability of the AmI Audio Library Explorer and the AmI Speakers Explorer was evaluated. An average of forty minutes was needed to conclude each session. At the end of each session participants filled the standard SUS as well as stated their agreement ('Strongly Disagree' to 'Strongly Agree') for four statements (EQ). Each statement regarded to a specific functionality of the two interfaces. Their answers were categorized into three adjective ratings, 'positive', 'negative' and 'neutral'. For statements, one to three (EQ1, EQ2 & EQ3), selected options were recorded as follows:

- 'Strongly Disagree' and 'Somewhat Disagree' were recorded as 'negative'
- 'Neutral' was a recorded as 'neutral'

- 'Somewhat Agree' and 'Strongly Agree' were recorded as 'positive'

For the fourth statement (EQ4), mappings were inverted as it is a negative statement. As seen in Figure 53, recorded statements were mainly positive, with a small percentage of users being neutral about EQ3 – the function of testing and EQ4 – the function of monitoring AmI Audio Clients using the AmI Speakers Explorer.

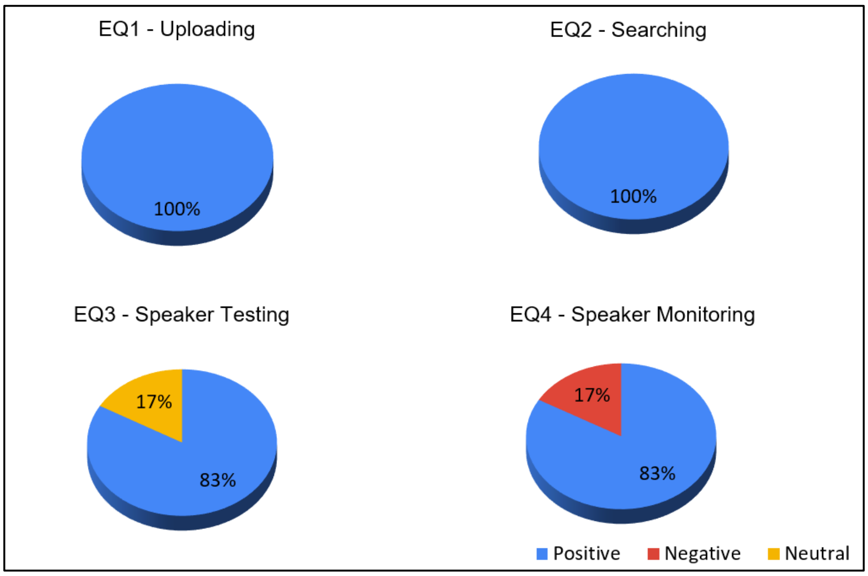


Figure 53 - Ratings of additional statements

Regarding the SUS, high scores were recorded from all participants, as seen in Table 14. The final calculated score was converted from the original 0 to 40 range to a 0 to 100 range. The calculated mean of scores received a value of 94.16 out of 100 as seen in Figure 54. An error bar was set to represent a 95% confidence interval, calculated using standard deviation. According to [127], many studies have accumulated large data sets with thousands of individual SUS questionnaires and hundreds of studies in order to provide grading scales for their interpretation. One such grading scale was provided in [128] and can be seen on Table 15. According to that grading scale, ACOUSMA's interfaces would be graded with an "A+".

Table 14 - SUS results

PARTICIP.	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	SUS SCORE
U1	4	1	5	1	5	1	4	1	4	1	92.5
U2	4	1	4	1	4	2	5	1	5	1	90
U3	5	1	5	1	5	1	4	1	5	1	97.5
U4	5	1	5	1	4	1	5	1	4	1	95
U5	5	1	5	2	5	1	5	1	5	1	97.5
U6	5	1	5	1	5	1	4	1	4	2	92.5
AVERAGE	**4,7**	**1,0**	**4,8**	**1,2**	**4,7**	**1,2**	**4,5**	**1,0**	**4,5**	**1,2**	**94,2**

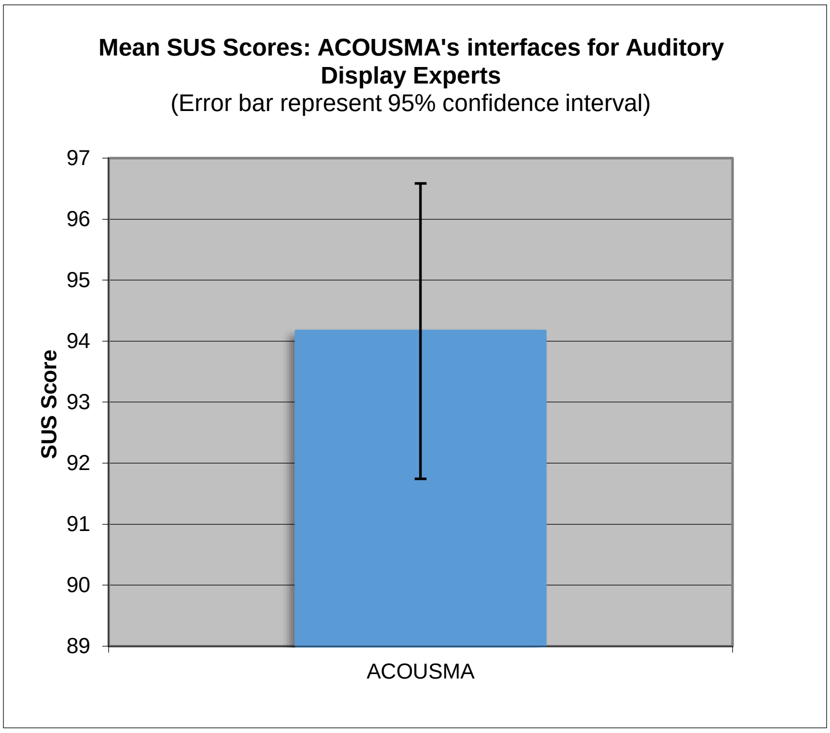


Figure 54 - Mean SUS scores

Table 15 - Grading scale for SUS [128]

Letter grade	SUS score range
A+	84.1–100
A	80.8–84.0
A-	78.9–80.7
B+	77.2–78.8
B	74.1–77.1
B-	72.6–74.0
C+	71.1-72.5
C	65.0–71.0
C-	62.7–64.9
D	51.7–62.6
F	0–51.6

Task completion times as well as recorded comments offered great insights for possible improvements to both interfaces. Here, those are going to be analyzed grouped by the interface function they evaluated.

5.2.2.1.1 Generating & uploading

Tasks 1 to 5 aimed to evaluate the function of generating and uploading auditory representations using the AmI Audio Library Explorer. Specifically, task 1&2 and 3 required participants to generate specific Spearcons and upload the to the AmI Audio Library database. Execution times were recorded for all tasks. Since the think-aloud method was followed, participants would often pause executing tasks and talk about different aspects of the interface. For consistency, times recorded where participants were not actively executing tasks were excluded from results. Also, note that times for task 3 were measured for each subtask (a, b & c) as seen on Table 16. Each subtask of task 3, was a repetition of tasks 1 & 2. Table 16 provides recorded times for tasks 1 to 3 as well as calculated averages. As seen on Figure 55, average execution times were significantly reduced each time participants repeated the same group of tasks (1&2) - generating and uploading Spearcons.

Table 16 – Recorded execution times: Tasks 1 to 3

PARTICIPANTS	TASK 1	TASK 2	TASK 3 (A)	TASK 3 (B)	TASK 3 (C)
U1	56	33	50	49	40
U2	32	15	58	27	14
U3	116	28	80	74	47
U4	19	46	38	31	19
U5	40	20	23	30	14
U6	50	62	37	20	18
AVERAGE EXECUTION TIME (SECONDS)	**52.2**	**34.0**	**47.7**	**38.5**	**25.3**

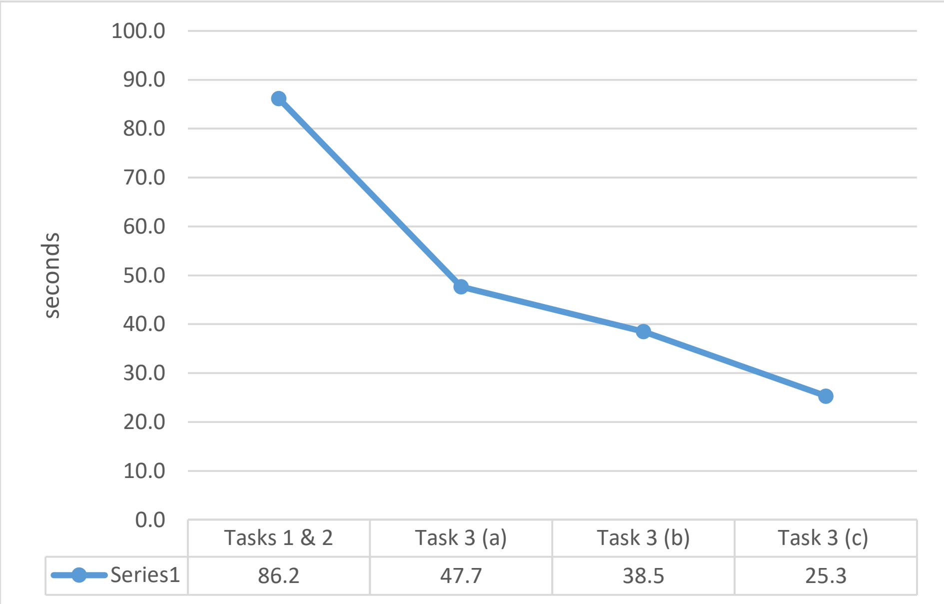


Figure 55 - Tasks 1 to 3: Decreasing execution time

Regarding tasks 4 and 5, participants were asked to upload two Auditory Icons (.mp3 files) from a desktop folder. A similar trend was recorded regarding execution times (see Table 17). Task 4 took longer to execute as participants haven't navigated from the home page to the upload form before. That was noticed both from task times and participants' comments and reactions.

Table 17 - Recorded execution times: Tasks 4 & 5

PARTICIPANTS	TASK 4	TASK 5
U1	82	33
U2	38	24
U3	59	27
U4	76	32
U5	70	20
U6	51	34
AVERAGE EXECUTION TIME (SECONDS)	**62.7**	**28.3**

For tasks 1-5, the following comments were aggregated:

- UC1 – The 'Add' text under the relevant navigation bar button is misleading.
- UC2 – The function of the direct upload button could be clearer.
- UC3 – Generated Spearcons should have a complete description (apart from descriptors)
- UC4 – Missing a feature to upload multiple Spearcons at once.
- UC5 – The auto play feature of the upload form is distracting.
- UC6 – Missing option to go directly to the Spearcon generation page after item has been uploaded

The majority of those comments can be mapped to specific UI elements of the AmI Audio Library Explorer, thus they constitute valuable guidelines for future improvements.

5.2.2.1.2 Searching & bookmarking

Tasks 6 to 9 aimed at evaluating the search functionality of the AmI Audio Library Explorer interface as well as the bookmark feature. Tasks 6 & 7 required participants to search for auditory representations the have uploaded in previous steps while task 8 required searching for an already uploaded item. Therefore, participants knew the specific attributes of the items they wanted to find in steps 6 & 7 while they only knew about the items descriptors for task 8. The effect of this is also noticeable by looking at the average execution times in Table 18.

Table 18 – Recorded execution times: Tasks 6 to 9

PARTICIPANTS	TASK 6	TASK 7	TASK 8	TASK 9
U1	15	68	180	12
U2	64	55	67	65
U3	76	27	75	29
U4	36	17	34	21
U5	30	84	20	20
U6	92	58	54	34
AVERAGE EXECUTION TIME (SECONDS)	**52.2**	**51.5**	**71.7**	**30.2**

Regarding task 9, which was to navigate to the bookmarks page and copy the identification number of a bookmarked item, participants instinctively executed the task; and that is also reflected in the average execution time.

Participants' comments as well as their reactions during interaction with the interface showed that the searching and filtering methods need to be improved both in terms of functionality as well as in terms of design. Specifically, participants commented:

- UC7 – Could not locate the search and filtering area right away.
- UC8 – The 'search by tag' option is unnoticeable.
- UC9 – The 'sort by' option is unnoticeable and options are unclear.
- UC10 – I would like more view options (e.g. list or grid view)
- UC11 – Missing filtering options at bookmarks page.
- UC12 – The code snippet button's functionality is unclear at first.

5.2.2.1.3 Testing & monitoring

The last two tasks regarded in testing and monitoring AmI Audio Clients using the AmI Speakers Explorer interface. Participants had to use an ID (that they copied in task 9) to test an auditory representation at a specific speaker of a specific simulation space location. The majority of participants did not have any problem in finding a specific speaker and deploying a test while one participant (U2) noted that it was fun to listen to the representation he previously uploaded, being played at the simulation space.

Table 19 – Recorded execution times: Tasks 10 & 11

PARTICIPANTS	TASK 10	TASK 11
U1	24	7
U2	31	5
U3	12	3
U4	46	1
U5	20	2
U6	7	1
AVERAGE EXECUTION TIME (SECS)	**23.3**	**3.2**

As indicated by task times (Table 19), initiating a test at a specific speaker and location of the AmI environment could be done rather quickly and stopping one could be done almost instantaneously. Participants provided useful comments:

- UC13 – I thought the speaker icons were interactive.
- UC14 – Functionality of the test button was unclear at first.
- UC15 – I would like to have a map view of the speaker's specific location in the environment.
- UC16 – Did not quite understand sorting names.

Those comments hinted towards new features as well as further improvements for the AmI Speakers Explorer interface.

5.2.2.2 Results: AmI Application Developer

Four participants had the role of AmI Application Developer and were asked to complete functionality of a user interface using ACOUSMA's API methods. Participants were asked to complete given tasks at their own time at home. Upon completion of tasks, a one-to-one interview was conducted. All participants had worked on AmI applications before but had different levels of expertise in the typescript programming language as mapped on Table 20 . Specifically, participants DEV1 & DEV3 were experts in the programming language with professional experience and had used the language before to implement AmI applications. For DEV2, it was their first time using the programming language. Although, DEV2 had a basic knowledge of JavaScript and professional experience in developing AmI applications. DEV4 had advanced knowledge of typescript but had no professional experience with the language. The selected sample of users aimed in capturing reactions and comments from developers with a wide range of expertise in the programing language.

Table 20 - AmI Developers: Level of expertise

Participant	Level of expertise
DEV1	Expert
DEV2	None (Basic knowledge of JavaScript)
DEV3	Expert
DEV4	Advanced

Regarding implementation, all participants successfully completed all given tasks and implemented all required functions using the API Client library module. Before each interview, SUS ratings were collected. During the interview sessions useful comments were recorded by each participant. Also, the time required to complete all tasks was asked.

Here, SUS ratings will be analyzed first. Then, reported completion times will be viewed and recorded comments will be analyzed.

5.2.2.2.1 SUS ratings

After completing all tasks and before each interview session was initiated, participants were asked to fill the standard SUS. Recorded results were used as helpful indicators to improve overall usability of the API. As seen on Table 21, the average SUS score recorded was 84,4%. Of course, this score is not validation of the API's usability as the purpose of the questionnaire here was to compliment findings of the interviews and to help indicate areas for usability improvement. Individual results were controversial. A significant low score was given by participant DEV2

(57,5%) while a perfect score was received by DEV4 (100%). The low score given by DEV2 can be attributed to two factors. Their confusion with certain aspects of the API, evident by commented issues (discussed in the next section) and their inexperience with the programming language that may have magnified those issues.

Table 21 - AmI Developers: SUS scores

Particip.	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	SUS Score
DEV1	5	2	4	1	5	1	4	1	5	1	92,5
DEV2	4	4	3	3	4	2	4	2	3	4	57,5
DEV3	4	1	5	2	5	1	4	1	4	2	87,5
DEV4	5	1	5	1	5	1	5	1	5	1	100
Average	**4,5**	**2,0**	**4,3**	**1,8**	**4,8**	**1,3**	**4,3**	**1,3**	**4,3**	**2,0**	**84,4**

Regardless, recorded SUS scores will be useful in future evaluations of the API as learnability and usability are better measured with higher number of participants and multiple iterations of the evaluation process.

5.2.2.2.2 Interview questions, comments & completion times

Beginning each interview session, the evaluator would ask the estimated completion time for all tasks. Participants would give an approximation of time needed to complete functionality for the interface. All participants gave a similar approximation, being around 10 minutes (min) as seen in Table 22.

Table 22 - AmI Developers: Completion times

Participant	Completion time (estimate)
DEV1	<10 min
DEV2	<10 min
DEV3	<10 min
DEV4	~=10 min

Completion time was similar for all participants regardless of their level of expertise in the programming language. That can be correlated with findings of the direct observation sessions where observed developers needed ten minutes to read documentation and successfully call a playback function (see 3.1.2.3 Direct Observation). Still, evaluation participants had to call more than one playback function as well as additional provided helper functions.

Regarding interview questions, three broad questions were asked to capture comments for three different aspects of the API. As seen in Table 23, the aim was to record participants' opinions on the documentation of the API, the input required by API functions and possible changes they thought should be made to the API. All questions were also used as a basis for constructive conversation where useful comments were captured for various other aspects of the API and the evaluation itself.

Table 23 - Questions to API Aspects

API Aspect	Question
Documentation	Did you think functions were well documented?
Input	Did you have trouble setting the input for ACOUSMA's functions?
Changes	What do you think should change?

Next, comments received in each question with respect to the aforementioned aspects will be analyzed and mapped to participants.

Regarding documentation, the following comments were received:

- DC1 – (DEV1, DEV2, DEV3): The API is well documented.
- DC2 – (DEV1, DEV2, DEV3, DEV4): The API documentation is understandable and was helpful in setting function input.
- DC3 – (DEV4): Documentation could be less formal, certain terms used could be formulated with a friendlier vocabulary.
- DC4 – (DEV4): Output of functions could be better documented.

Three out of four participants commented that the API is well documented (DC1) while all participants found the API to be understandable and helpful when it comes to setting function input (DC2). Participant DEV4 stated that they disliked the formal language used in some parts of the documentation and commented that they would prefer if some words would be changed to other, less formal. Specifically, they thought it would be simpler if 'descriptors' would be changed to 'tags'. The same participant commented that the output of each function could be better documented. He mentioned that a short description could be added to each element of the output interface.

Regarding function input, the following comments were received:

- IC1 – (DEV1, DEV3, DEV4): I had no problem setting input for functions.
- IC2 – (DEV2): I had trouble finding the right function to use. Some global functions seemed suitable but had different input than needed. That leaded to finding that required functions were under. recommender.
- IC3 – (DEV2): I liked the general idea of having several generic, easy-to-use functions and some more advanced functions.

Most participants had no problem in setting function input (IC1). DEV1 and DEV3 also mentioned that they really liked how the IDE would display hints during function input, significantly aiding in input completion. DEV2 discussed that initially proceeded in completing tasks without reading the documentation or provided examples of usage for the API. DEV2 commented that it was hard to find the right function to use, as some global functions (expert functions – see 4.3.3.2 API Client Library) seemed to be suitable (PlayAtUser & PlayAtLocation). The mismatch on task requirement and function input led DEV2 to finding that all required API functions were under '. recommender'. Even though DEV2 had a frustrating experience at first, he commented that he liked the idea that more advanced functions (expert functions) existed along with easy-to-use functions (recommender functions) for developers.

Regarding possible changes, participants suggested the following:

- CC1 – (DEV1): Recommender should be a separate module to avoid writing 'acousma.recommender'.
- CC2 – (DEV1): A function to play a recommended sound to multiple users would be welcome.
- CC3 – (DEV1): I would like to have sound recommendation for a specific location, not just for a specific user.

- CC4 – (DEV1): I would prefer to use a single string instead of an array of strings as input for the descriptors property.
- CC5 – (DEV2): It would be preferable if the '.recommender' attribute would be used only for recommender purposes (i.e. get a recommendation) and then use recommender response with global playback functions.
- CC6 – (DEV2): The PlayAtUser (global) input could be changed to allow both experts in auditory displays and novice users to deploy sounds. Parameters could have default values that expert users could change if need be. A specific structure could be implemented for input, e.g. a JSON object.
- CC7 – (DEV3) - I do not think any change is necessary. I think reading the documentation is enough to quickly and efficiently use functions.
- CC8 – I would like to have a recommender function return just the stream url of the recommended sound.

From their suggestions it was easily noticeable that changes should be made to the structure of the API module (CC1, CC5, CC6), and specifically to the placement of recommender functions. Participants also noted that they would like to have additional recommender functions (CC2, CC8). Moreover, participants suggested that changes should be made to the functions input (CC4, CC6). All of their comments pointed towards improvements for the API module as well as possible new features for ACOUSMA.

The fact that all participants encountered no problem in completing all tasks can be considered as positive evidence that the API client library module is usable. Although, through their comments and helpful indications, participants made clear that the API needs to be reworked and further evaluated to ensure user requirements are met.

5.2.2.3 Results: AmI environment inhabitant

Participants with the AmI environment inhabitant role followed five everyday scenarios emulating a smart apartment resident using an AmI application (weather forecast application) with ACOUSMA features

(Auditory Display Recommender). Each session lasted an average of thirty minutes. During each session, the evaluator would ask a question after scenarios A., C. & E., that regarded to the understandability of the recommended auditory display. At the end of each session, participants were asked to fill the standard UMUX questionnaire stating their agreement (1 to 7) for four statements. Also, an open discussion was initiated with each participant, to gather their opinions about the overall experience and the recommendations provided. Gathered results showed positive evidence towards the usability of the Auditory Display Recommender as well as overall user experience of an ACOUSMA enhanced AmI application.

Each statement of the UMUX aims to assess a unique user experience aspect. The first statement (UMUX1) measures effectiveness, the second (UMUX2) measures satisfaction, the third (UMUX3) overall usability and the fourth (UMUX4) efficiency. For calculating scores, odd items must be scored as [score -1] and even items as [7-score], effectively recoding original values to a 0-6 range (where 6 is the optimum) [134]. The final UMUX score was calculated in the 0-100 range. That is done by dividing the total with twenty-four (which is the maximum sum of all statements) and then multiplying by one-hundred.

The ACOUSMA enhanced AmI application scored positively in all four measures as seen in Table 24, gathering an average UMUX score of 89.3 out of 100. Although positive, the poorest score was recorded in UMUX1 and UMUX4 both with an average score of 5,3 out of 6. The latter hinted towards the need for improvements in effectiveness and efficiency of ACOUSMA's recommender.

Table 24 - UMUX scores

PARTICIPANT	UMUX 1	UMUX 2	UMUX 3	UMUX 4	UMUX SCORE
P1	**5**	**6**	**5**	**5**	**87.5**
P2	**5**	**6**	**6**	**6**	**95.8**
P3	**5**	**4**	**6**	**5**	**83.3**
P4	**6**	**6**	**5**	**4**	**87.5**
P5	**5**	**5**	**6**	**6**	**91.7**
P6	**6**	**6**	**6**	**6**	**100**
P7	**5**	**5**	**4**	**5**	**79.2**
AVERAGE SCORE	**5.3**	**5.4**	**5.4**	**5.3**	**89.3**

Data were also recorded from participants' interaction with the provided critiquing interface, as well as answers to questions posted at the end of scenarios.

5.2.2.3.1 Post-scenario questions & comments

Through scenarios, the same workflow was followed by each participant:

- During Scenario A. the recommender would choose an auditory representation for the information 'heavy rain', tailored for a blank user profile (as the participant had yet to provide any feedback). The participant would listen to that display.
- During Scenario B. the participant would provide feedback to the recommender regarding the 'heavy rain' auditory display using the provided critiquing interface (a tablet device). Using the interface, the participant could listen to the display and view the information it was trying to describe.
- During Scenario C. the recommender would choose an auditory representation for the information 'sunny weather', tailored for

their user profile as shaped in Scenario B. Participants would then listen to that display.

- During Scenario D. participants would provide feedback to the recommender regarding the ‘sunny weather’ auditory display using the provided critiquing interface (a tablet device). Using the interface, the participant could listen to the display and view the information it was trying to describe.
- During Scenario E. the recommender would retrieve from the case history, the auditory display for ‘heavy rain’ as that was adjusted in Scenario B. for the specific participant. The participant would listen to that display.

After the end of scenario A. C. & E., the evaluator would ask the participant about their understanding of the provided display (“What was the weather forecast about?”). Recorded answers were categorized into three categories depending on the participants understanding of the information being transmitted via the auditory display:

- ‘Exact’ was recorded when a participant would fully and precisely understand the information.
- ‘Partial’ was recorded when a participant would understand the context of information but not the precise information.
- ‘None’ would be recorded when a participant did not understand any information from the auditory display.

As seen in Figure 56, all participants understood the information being transmitted, especially after adjusting auditory displays to their preferences. At the end of Scenario A., only 71.4% of participants understood that the display was about 'heavy rain" and recorded as 'exact'. The rest understood it was about 'rain', but did not quantify the information, thus recorded as 'partial'. After providing feedback to the recommender, increased understandability was recorded. At the end of Scenario E, 85,7% (6/7) of participants were able to precisely distinguish that the recommended display was about 'heavy rain'. The increase of understandability could be accounted to participants having the chance to use the critiquing mechanism in Scenario B. Regarding Scenario C, a similar trend was followed with 85,7% of participants precisely understanding the auditory display to be about 'sunny weather'.

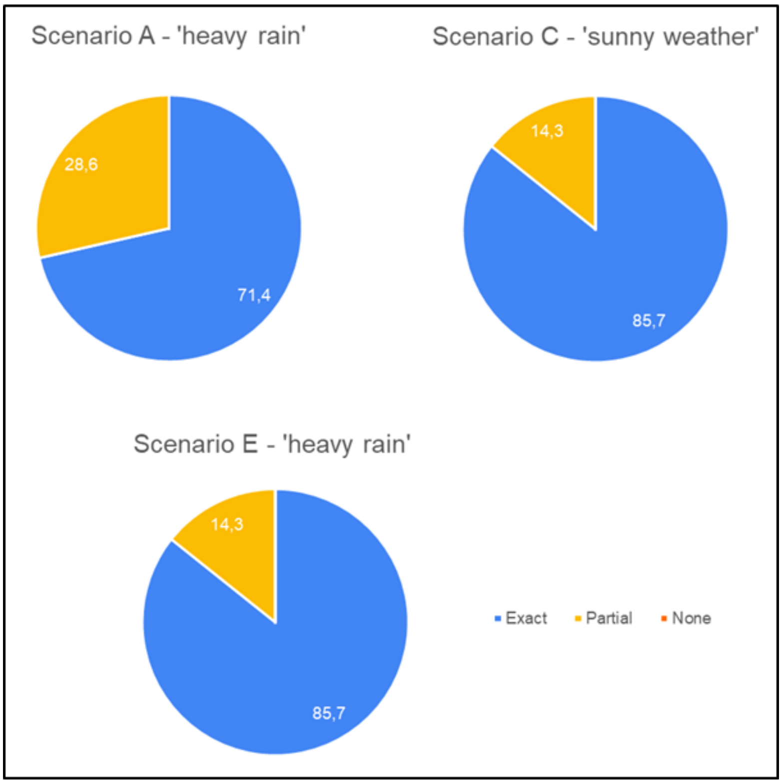


Figure 56 - Understandability of recommended auditory displays

Using the critiquing mechanism helped participants map the auditory display to the precise information it was used to describe. Also, by providing feedback, it helped the recommender provide personalized results.

During Scenario B, all participants responded 'yes' to the 'yes or no' question about their preference of the recommended auditory display (seen on the critiquing interface). That signified that the recommender's initial selection for the blank user profile was sufficient, understandable and liked. According to that feedback, the recommender adjusted the user profile and successfully provided personalized auditory displays in Scenarios C. & E. That was noted since most participants could precisely understand the recommended displays during those scenarios and by the fact that 6 out of 7 participants chose 'yes' to the same feedback question during Scenario D. Only one participant chose 'no' during Scenario D and used the critiquing mechanism to retrieve a list of recommendations. That participant noted that she liked most of the recommendations and that she had a hard time choosing a favorite. Eventually, she chose a preferred auditory display for the information 'sunny weather' and the recommender adjusted her profile and the specific case in the case history.

At the end of the evaluation, through an open discussion, participants were asked to share their thoughts about the overall experience and the provided auditory displays. Participants noted the following:

- PC1 – I would like to be able to comment on the timing, sounds (auditory displays) were provided. For example, I would prefer if I would listen to the weather forecast before the taxi notification during Scenario A. & C.
- PC2 – I liked the simulation space and felt like I was in a real apartment. I would really like to have such a weather forecast application at home.
- PC3 - I found the sounds to be really close to my preferences.
- PC4 - I would like to be able to comment on the volume a sound played.
- PC5 – I would like to be able to adjust my profile on demand.

- PC6 – I liked the timing the recommender chose to play the weather forecast (in Scenario A)
- PC7 - It would like to be able to upload my own sounds to the system.

Based on their reactions as well as comments, participants seemed to have enjoyed the recommended auditory displays as well as the evaluation process. From their comments, participants gave great insights on possible new features for the recommender and the critiquing mechanism. A clear participant demand towards that direction (noted by three participants) was to have the ability to comment on the timing ACOUSMA provided a recommended auditory display.

Chapter 6
6. Conclusions & Future work

6.1 Conclusions

This Thesis has presented the design, implementation and first steps towards the usability evaluation of ACOUSMA, a platform that encapsulates all functions necessary to standardize the integration of context-aware, personalized auditory displays in Ambient Intelligence environments. Challenges revealed by related work and by discussing with and eliciting requirements for target user groups, pointed towards the need for a tool to simplify and promote the use of personalized, context-aware, auditory feedback in AmI environment applications. Through careful design, the blueprints of a micro-service, modular system were constructed to meet all functional and non-functional user requirements. Using ACOUSMA's interfaces and modules, auditory display experts can generate, upload, store and share auditory representations; AmI application developers can efficiently integrate meaningful auditory displays that are personalized and context-aware; AmI environment inhabitants can use the intelligent mechanism module to critique and adjust auditory displays to their preferences at any time. The implementation of ACOUSMA surpassed all challenges and objectives set (see 2.6 Objectives) while the user-based evaluation yielded positive evidence that user requirements were successfully met. It is of course evident that, in order to ensure the fulfilment of all user requirements and usability of ACOUSMA, AmI applications must be developed to integrate its features. In this regard, future work has already been planned and is analyzed in the next section.

6.2 Future Work

The next step of this work is to further evaluate the usability for all ACOUSMA's modules through usage of ACOUSMA in AmI applications of various domains. Results of the user-based evaluation has shown that ACOUSMA is ready to be used by AmI applications to enhance any AmI environment with auditory displays.

In addition, three individual research projects are planned. Every project will explore the applicability of ACOUSMA into different contexts. Research results from these applications may be used to upgrade and improve the system but may also provide significant insights into yet undiscovered directions.

A variety of successful AmI applications for many domains have been implemented and are demonstrated by the ICS-FORTH Ambient Intelligence Programme [128]. Some of those applications may benefit from providing auditory feedback. Examining the use of ACOUSMA for enhancing those applications with auditory displays can aid in measuring the applicability and ease of integration of the system in various application domains.

In order to investigate performance of creating a new AmI application that uses ACOUSMA, an application for the smart home will be designed and implemented. The application will use ACOUSMA combined with other services to promptly notify residents for various events. This work can help understand how ACOUSMA affects the workflow of designing and developing an AmI application as well as an auditory display application.

The intelligent mechanism module has already shown potential for different use cases. For example, in the context of auditory displays, it could be extended to recommend a sonification technique based on a given dataset. From evaluation results, it was made clear that users would like the critiquing interface to including commenting on timing auditory displays are initiated. This may lead to extending the recommender as well as the queue manager to facilitate that. An interesting challenge would be to adapt the module to recommend visual displays. With this notion, an analogy-system to ACOUSMA could be designed with the ability automatically select an appropriate method (from both visual and auditory channels) for displaying information based on context and user preferences.

7. References and Bibliography

[1] E. Aarts and R. Wichert, "Ambient intelligence," *Technology Guide,* p. 244–249, 2009.

[2] J. Krumm, Ubiquitous Computing Fundamentals, 2009.

[3] F. Adelstein, S. K. Gupta and G. Richard, "Fundamentals of Mobile and Pervasive Computing, McGraw-Hill 2005," *,* 2015.

[4] D. J. Cook, J. C. Augusto and V. R. Jakkula, "Ambient intelligence: Technologies, applications, and opportunities," *Pervasive and Mobile Computing,* vol. 5, p. 277–298, 8 2009.

[5] P. L. Emiliani and C. Stephanidis, "Universal access to ambient intelligence environments: Opportunities and challenges for people with disabilities," *IBM Systems Journal,* vol. 44, p. 605–619, 2005.

[6] J. C. Augusto, "Past, Present and Future of Ambient Intelligence and Smart Environments," *Communications in Computer and Information Science,* p. 3–15, 2010.

[7] K. Groß-Vogt, M. Weger, R. Höldrich, T. Hermann, T. Bovermann and S. Reichmann, "Augmentation of an Institute's Kitchen: An Ambient Auditory Display of Electric Power Consumption," 2018.

[8] R. Tünnermann, J. Hammerschmidt and T. Hermann, "Blended Sonification –Sonification For Casual Information Interaction," *,* 2013.

[9] E. Aarts and J. Encarnação, "Into Ambient Intelligence," *True Visions,* p. 1–16, 2006.

[10] K. Ducatel, U. européenne. Technologies de la société de l'information, U. européenne. Institut d'études de prospectives technologiques and U. européenne. Société de l'information conviviale, "Scenarios for ambient intelligence in 2010," 2001.

[11] E. H. L. Aarts and J. L. Encarnação, "True Visions: The Emergence of Ambient Intelligence (Frontiers Collection)," *,* 2006.

[12] K. Dohsaka, "Information-Report on," *Ambient Intelligence Symposium,* vol. 4, p. 64, 2006. 2006.

[13] C. Ramos, "Ambient Intelligence – A State of the Art from Artificial Intelligence Perspective," *Progress in Artificial Intelligence,* p. 285–295.

[14] J. Rech and K.-D. Althoff, "Artificial Intelligence and Software Engineering: Status and Future Trends.," *KI,* vol. 18, pp. 5-11, 2004.

[15] A. Vasilakos and W. Pedrycz, "Ambient Intelligence, Wireless Networking, And Ubiquitous Computing," *,* 2006.

[16] A. Schmidt, M. Kranz and P. Holleis, "Interacting with the ubiquitous computer," 2005.

[17] Y. Baram and A. Miller, "Auditory feedback control for improvement of gait in patients with Multiple Sclerosis," *Journal of the Neurological Sciences,* vol. 254, p. 90–94, 3 2007.

[18] L. Shams and A. R. Seitz, "Benefits of multisensory learning," *Trends in Cognitive Sciences,* vol. 12, p. 411–417, 11 2008.

[19] E. Kohler, "Hearing Sounds, Understanding Actions: Action Representation in Mirror Neurons," *Science,* vol. 297, p. 846–848, 8 2002.

[20] B. C. J. Moore, An Introduction to the Psychology of Hearing, 1977.

[21] W. W. Gaver, "What in the World Do We Hear?: An Ecological Approach to Auditory Event Perception," *Ecological Psychology,* vol. 5, p. 1–29, 3 1993.

[22] M. De Lucia, C. Camen, S. Clarke and M. M. Murray, "The role of actions in auditory object discrimination," *NeuroImage,* vol. 48, p. 475–485, 11 2009.

[23] M. M. Murray, "Rapid Brain Discrimination of Sounds of Objects," *Journal of Neuroscience,* vol. 26, p. 1293–1302, 1 2006.

[24] J. V. G. Robertson, T. Hoellinger, P. Lindberg, D. Bensmail, S. Hanneton and A. Roby-Brami, "Effect of auditory feedback differs according to side of hemiparesis: a comparative pilot study," *Journal of NeuroEngineering and Rehabilitation,* vol. 6, 12 2009.

[25] G. Rizzolatti and L. Craighero, "THE MIRROR-NEURON SYSTEM," *Annual Review of Neuroscience,* vol. 27, p. 169–192, 7 2004.

[26] G. Kramer, B. Walker, T. Bonebright, P. Cook and J. H. Flowers, "Sonific ation Report: Status of the Field and Research Agenda," *,* 2010.

[27] D. R. Perrott, K. Saberi, K. Brown and T. Z. Strybel, "Auditory psychomotor coordination and visual search performance," *Perception & Psychophysics,* vol. 48, p. 214–226, 5 1990.

[28] R. B. Welch and D. H. Warren, "Immediate perceptual response to intersensory discrepancy.," *Psychological Bulletin,* vol. 88, p. 638–667, 1980.

[29] Á. Csapó and G. Wersényi, "Overview of auditory representations in human-machine interfaces," *ACM Computing Surveys,* vol. 46, p. 1–23, 11 2013.

[30] S. Serafin, K. Franinovic, T. Hermann, G. Lemaitre, M. RInott and D. Rocchesso, The Sonification Handbook, 2011.

[31] W. Gaver, "Auditory Icons: Using Sound in Computer Interfaces," *Human-Computer Interaction,* vol. 2, p. 167–177, 6 1986.

[32] W. Gaver, "The SonicFinder: An Interface That Uses Auditory Icons," *Human-Computer Interaction,* vol. 4, p. 67–94, 3 1989.

[33] W. W. Gaver, "Sound Support For Collaboration," *Proceedings of the Second European Conference on Computer-Supported Cooperative Work ECSCW '91,* p. 293–308, 1991.

[34] M. Blattner, D. Sumikawa and R. Greenberg, "Earcons and Icons: Their Structure and Common Design Principles," *Human-Computer Interaction,* vol. 4, p. 11–44, 3 1989.

[35] S. Brewster, "Providing a Structured Method for Integrating Non-Speech Audio into Human-Computer Interfaces," *Heslington, York: University of York,* 1994.

[36] G. Leplâtre and S. A. Brewster, "An Investigation of Using Music to Provide Navigation Cues," 1998.

[37] B. N. Walker, A. Nance and J. Lindsay, "SPEARCONS: SPEECH-BASED EARCONS IMPROVE NAVIGATION PERFORMANCE IN AUDITORY MENUS," *,* 2006.

[38] B. N. Walker, J. Lindsay, A. Nance, Y. Nakano, D. K. Palladino, T. Dingler and M. Jeon, "Spearcons (Speech-Based Earcons) Improve Navigation Performance in Advanced Auditory Menus," *Human Factors: The Journal of the Human Factors and Ergonomics Society,* vol. 55, p. 157–182, 7 2012.

[39] D. K. Palladino and B. N. Walker, "LEARNING RATES FOR AUDITORY MENUS ENHANCED WITH SPEARCONS VERSUS EARCONS," *,* 2007.

[40] T. Dingler, J. Lindsay, B. N. Walker, L.-M.-U. München and F. Medieninformatik, "LEARNABILTIY OF SOUND CUES FOR ENVIRONMENTAL FEATURES: AUDITORY ICONS, EARCONS, SPEARCONS, AND SPEECH," *,* 2008.

[41] G. Wersenyi, "EVALUATION OF USER HABITS FOR CREATING AUDITORY REPRESENTATIONS OF DIFFERENT SOFTWARE APPLICATIONS FOR BLIND PERSONS," *,* 2008.

[42] M. Jeon and B. N. Walker, ""Spindex": Accelerated Initial Speech Sounds Improve Navigation Performance in Auditory Menus," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting,* vol. 53, p. 1081–1085, 10 2009.

[43] M. Jeon and B. N. Walker, "Spindex (Speech Index) Improves Auditory Menu Acceptance and Navigation Performance," *ACM Transactions on Accessible Computing,* vol. 3, p. 1–26, 4 2011.

[44] P. Froehlich and F. Hammer, "Expressive Text-to-Speech: A user-centred approach to sound design in voice-enabled mobile

applications," in *Proc. Second Symposium on Sound Design*, 2004.

[45] G. Wersényi, "EVALUATION OF AUDITORY REPRESENTATIONS FOR SELECTED APPLICATIONS OF A GRAPHICAL USER INTERFACE," *,* 2009.

[46] G. Wersényi, "Auditory Representations of a Graphical User Interface for a Better Human-Computer Interaction," *Auditory Display,* p. 80–102, 2010.

[47] G. Nemeth, G. Olaszy and T. G. Csapo, "Spemoticons: Text to Speech Based Emotional Auditory Cues," *,* 2011.

[48] G. Kramer, Auditory Display: Sonification, Audification, And Auditory Interfaces, 1994.

[49] G. Parseihian and B. F. G. Katz, "Rapid Auditory System Adaptation Using a Virtual Auditory Environment," *i-Perception,* vol. 2, p. 805–805, 10 2011.

[50] M. McGee-Lennon, M. Wolters, R. McLachlan, S. Brewster and C. Hall, "Name that tune," 2011.

[51] M. McGee-Lennon and S. Brewster, "Reminders that Make Sense: Designing Multimodal Notifications for the Home," 2011.

[52] R. McLachlan, M. McGee-Lennon and S. Brewster, "The sound of musicons: investigating the design of musically derived audio cues," *,* 2012.

[53] A. Ankolekar, T. Sandholm and L. Yu, "Play it by ear," 2013.

[54] R. D. Patterson, Guidelines for auditory warning systems on civil aircraft, Civil Aviation Authority, 1982.

[55] R. D. Patterson, "Auditory warning sounds in the work environment," *Philosophical Transactions of the Royal Society B,* vol. 327, no. 1241, pp. 485-492, 1990.

[56] J. Edworthy, S. Loxley and I. Dennis, "Improving Auditory Warning Design: Relationship between Warning Sound Parameters and Perceived Urgency," *Human Factors: The Journal of the Human Factors and Ergonomics Society,* vol. 33, p. 205–231, 4 1991.

[57] E. J. Hellier, J. Edworthy and I. Dennis, "Improving Auditory Warning Design: Quantifying and Predicting the Effects of Different Warning Parameters on Perceived Urgency," *Human Factors: The Journal of the Human Factors and Ergonomics Society,* vol. 35, p. 693–706, 12 1993.

[58] E. C. Haas and J. Edworthy, "Designing urgency into auditory warnings using pitch, speed and loudness," *Computing & Control Engineering Journal,* vol. 7, p. 193–198, 8 1996.

[59] B. N. Walker and G. Kramer, "Ecolological Psychoacoustics and Auditory Displays," *Ecological Psychoacoustics,* p. 149–174, 2004.

[60] B. N. Walker and M. A. Nees, "Theory of sonification," *The sonification handbook,* p. 9–39, 2011.

[61] B. N. Walker and G. Kramer, "Human Factors and the Acoustic Ecology: Considerations for Multimedia Audio Design," *Journal of The Audio Engineering Society,* 1996.

[62] K. M. Franklin and J. C. Roberts, "A path based model for sonification".

[63] L. Brown, S. Brewster, R. Ramloll, W. Yu and B. Riedel, "BROWSING MODES FOR EXPLORING SONIFIED LINE GRAPHS," *,* 2002.

[64] T. Hermann and A. Hunt, "Guest Editors' Introduction: An Introduction to Interactive Sonification," *IEEE MultiMedia,* vol. 12, p. 20–24, 4 2005.

[65] A. de Campo, "A data sonification design space map," in *Proc. of the 2nd International Workshop on Interactive Sonification, York, UK*, 2007.

[66] F. Dombois and S. Birlinghoven, "Auditory seismology on free oscillations, focal mechanisms, explosions and synthetic seismograms," *,* 2002.

[67] S. D. Speeth, "Seismometer Sounds," *The Journal of the Acoustical Society of America,* vol. 33, p. 909–916, 7 1961.

[68] F. Grond and J. Berger, "Parameter mapping sonification," in *The sonification handbook*, 2011.

[69] T. Bovermann, T. Hermann and H. Ritter, "TANGIBLE DATA SCANNING SONIFICATION MODEL," in *Proceedings of the International Conference on Auditory Display*, 2006.

[70] B. A. Smith and S. K. Nayar, "The RAD: making racing games equivalently accessible to people who are blind.," in *Proceedings of the 2018 CHI conference on human factors in computing systems, ACM*, 2018.

[71] I. C. o. A. D. Proceedings, "Georgia Institute of Technology," Georgia Tech SMARTech , [Online]. Available: https://smartech.gatech.edu/handle/1853/49750.

[72] B. N. Walker and J. T. Cothran, "SONIFICATION SANDBOX: A GRAPHICAL TOOLKIT FOR AUDITORY GRAPHS," *,* 2003.

[73] B. K. Davison and B. N. Walker, "SONIFICATION SANDBOX RECONSTRUCTION: SOFTWARE STANDARD FOR AUDITORY GRAPHS," *,* 2007.

[74] R. J. Winton, T. M. Gable, J. Schuett and B. N. Walker, "A sonification of Kepler space telescope star data," *,* 2012.

[75] R. M. Candey, A. M. Schertenleib and W. L. D. Merced, "Xsonify sonification tool for space physics," *,* 2006.

[76] S. Phillips and A. Cabrera, "Sonification workstation," 2019.

[77] F. Dombois, O. Brodwolf, O. Friedli, I. Rennert and T. Koenig, "SONIFYER A Concept, a Software, a Platform," *,* 2008.

[78] A. Schoon and F. Dombois, "Sonification in Music," *,* 2009.

[79] D. Worrall, M. Bylstra, S. Barrass and R. T. Dean, "Sonipy : the design of an extendable software framework for sonification research and auditory display," in *ICAD 2007 : Immersed in Organized Sound : Proceedings of the 13th International Conference on Auditory Display, June 26-29 2007, Montreal, Quebec, Canada*, 2007.

[80] D. Worrall, "The Sonipy Framework: Getting Started," *Human–Computer Interaction Series,* p. 181–211, 2019.

[81] V. Lazzarini, S. Yi, J. Ffitch, J. Heintz, Ø. Brandtsegg and I. McCurdy, Csound: A Sound and Music Computing System, 2016.

[82] G. Schmitz, J. Bergmann, A. O. Effenberg, C. Krewer, T.-H. Hwang and F. Müller, "Movement Sonification in Stroke Rehabilitation," *Frontiers in Neurology,* vol. 9, 6 2018.

[83] S. Wilson, D. Cottle and N. Collins, "The SuperCollider Book," *,* 2011.

[84] M. Ballora, B. Pennycook, P. C. Ivanov, L. Glass and A. L. Goldberger, "Heart Rate Sonification: A New Approach to Medical Diagnosis," *Leonardo,* vol. 37, p. 41–46, 2 2004.

[85] N. Collins, "Sonification of the Riemann Zeta function.," in *(2019). 25th International Conference on Auditory Display (ICAD 2019). : Georgia Tech Library*, 2019.

[86] R. Tünnermann, C. Leichsenring, T. Bovermann and T. Hermann, "Upstairs: A calm auditory communication and presence system," *,* 2015.

[87] D. Lockton, F. Bowden, C. Brass and R. Gheerawo, "Powerchord: Towards Ambient Appliance-Level Electricity Use Feedback through Real-Time Sonification," *Ubiquitous Computing and Ambient Intelligence. Personalisation and User Adapted Services,* p. 48–51, 2014.

[88] A. Harman, H. Dimitrov, R. Ma, S. Whitehouse, Y. Li, P. Worgan, T. Omirou and A. Roudaut, "NotiFall," 2016.

[89] C. Leichsenring, J. Yang, J. Hammerschmidt and T. Hermann, "Challenges for smart environments in bathroom contexts," 2016.

[90] S. Bakker, E. van den Hoven and B. Eggen, "Exploring Interactive Systems Using Peripheral Sounds," *Haptic and Audio Interaction Design,* p. 55–64, 2010.

[91] N. Anyfantis, E. Kalligiannakis, A. Tsiolkas, A. Leonidis, M. Korozi, P. Lilitsis, M. Antona and C. Stephanidis, "AmITV," 2018.

[92] N. Partarakis, M. Antona and C. Stephanidis, "Adaptable, Personalizable and Multi User Museum Exhibits," *Curating the Digital,* p. 167–179, 2016.

[93] E. Zidianakis, K. Stratigi, D. Ioannidi, N. Partarakis, M. Antona and C. Stephanidis, "The Farm Game: A Game Designed to Follow Children's Playing Maturity," *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering,* p. 20–28, 2017.

[94] E. Sykianaki, A. Leonidis, M. Antona and C. Stephanidis, "CaLmi," 2019.

[95] M. Antona, S. Ntoa, I. Adami and C. Stephanidis, "User Requirements Elicitation for Universal Access," *Human Factors and Ergonomics,* p. 1–14, 6 2009.

[96] C. Stephanidis, The Universal Access Handbook, 2009.

[97] M. Yousuf and M. A. M.Asger, "Comparison of Various Requirements Elicitation Techniques," *International Journal of Computer Applications,* vol. 116, p. 8–15, 4 2015.

[98] J. Nehmer, M. Becker, A. Karshmer and R. Lamm, "Living assistance systems: an ambient intelligence approach," in *Proceedings of the 28th international conference on Software engineering*, 2006.

[99] L. Chung and J. C. P. Leite, On Non-Functional Requirements in Software Engineering, 2009, pp. 363-379.

[100] E. U. Warriach, "State of the Art: Embedded Middleware Platform for A Smart Home," *International Journal of Smart Home,* vol. 7, no. 6, pp. 275-294, 2013.

[101] F. Fernandez and G. Pallis, "Opportunities and challenges of the Internet of Things for healthcare," in *4th International Conference on Wireless Mobile Communication and Healthcare - "Transforming healthcare through innovations in mobile and wireless technologies"*, 2014.

[102] F. P. Lim, "A Review-Analysis of Network Topologies for Microenterprises," in *Circuits, Control, Communication, Electricity, Electronics, Energy, System, Signal and Simulation 2016*, 2016.

[103] J. Buck, "Visual displays," p. 195–231, 1983.

[104] O. Bones, T. J. Cox and W. J. Davies, "Sound Categories: Category Formation and Evidence-Based Taxonomies," *Frontiers in Psychology,* vol. 9, 7 2018.

[105] A. H. N. G. A. N. D. A. M. MFONDOUM, M. TCHINDJANG, J. M. Mfondoum and I. Makouet, "Eisenhower matrix* Saaty AHP= Strong actions prioritization? Theoretical literature and lessons drawn from empirical evidences," *IAETSD-Journal for Advanced Research in Applied Sciences,* vol. 6, p. 13–27.

[106] P. Batra, "Eisenhower Box for Prioritising Waiting List of Orthodontic Patients," *oral health and dental management,* vol. 2017, no. 1, pp. 0-0, 2017.

[107] A. V. Kirillov, D. K. Tanatova, M. V. Vinichenko and S. A. Makushkin, "Theory and Practice of Time-Management in Education," *Asian Social Science,* vol. 11, no. 19, pp. 193-204, 2015.

[108] R. Burke and M. Ramezani, "Matching Recommendation Technologies and Domains," *Recommender Systems Handbook,* p. 367–386, 10 2010.

[109] M. Montaner, B. López and J. L. de la Rosa, ""A taxonomy of recommender agents on the internet."," *Artificial Intelligence Review,* vol. 19, p. 285–330, 2003.

[110] I. Fernández-Tobı□as, I. Cantador, M. Kaminskas and F. Ricci, "Cross-domain recommender systems: A survey of the state of the art," in *Spanish conference on information retrieval*, 2012.

[111] F. Ricci, L. Rokach and B. Shapira, "Introduction to Recommender Systems Handbook," *Recommender Systems Handbook,* p. 1–35, 10 2010.

[112] R. Burke, "Hybrid Web Recommender Systems," *The Adaptive Web,* p. 377–408.

[113] C. C. Aggarwal, "Knowledge-Based Recommender Systems," *Recommender Systems,* p. 167–197, 2016.

[114] A. V. Aho, J. E. Hopcroft and J. D. Ullman, "On Finding Lowest Common Ancestors in Trees," *SIAM Journal on Computing,* vol. 5, p. 115–132, 3 1976.

[115] D. Harel and R. E. Tarjan, "Fast Algorithms for Finding Nearest Common Ancestors," *SIAM Journal on Computing,* vol. 13, p. 338–355, 5 1984.

[116] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," in *Advances in Neural Information Processing Systems 26*, 2013.

[117] J. Nielsen, "Usability inspection methods," 1994.

[118] J. Nielsen and R. Molich, "Heuristic evaluation of user interfaces," 1990.

[119] J. Nielsen, "Severity Ratings for Usability Problems," *,* 2006.

[120] "ICS-FORTH Ambient Intelligence Programme," [Online]. Available: http://ami.ics.forth.gr/domain.

[121] M. W. v. Someren, "The think aloud method : a practical guide to modelling cognitive processes," *,* 1994.

[122] j. Brooke, "SUS: A 'Quick and Dirty' Usability Scale," *,* pp. 207-212, 1996.

[123] J. Brooke, "SUS: a retrospective," *Journal of Usability Studies archive,* vol. 8, no. 2, pp. 29-40, 2013.

[124] K. Finstad, "The Usability Metric for User Experience," *Interacting with Computers,* vol. 22, p. 323–327, 9 2010.

[125] M. I. Berkman and D. Karahoca, "Re-assessing the usability metric for user experience (UMUX) scale," *Journal of Usability Studies archive,* vol. 11, no. 3, pp. 89-109, 2016.

[126] J. R. Lewis, "Critical Review of 'The Usability Metric for User Experience'," *Interacting with Computers,* vol. 25, p. 320–324, 3 2013.

[127] J. R. Lewis and J. Sauro, "Item benchmarks for the system usability scale," *Journal of Usability Studies archive,* vol. 13, no. 3, pp. 158-167, 2018.

[128] J. Sauro and J. R. Lewis, "Quantifying user research," *Quantifying the User Experience,* p. 9–18, 2016.

[129] C. C. Aggarwal, "An Introduction to Recommender Systems," *Recommender Systems,* p. 1–28, 2016.

[130] J. B. Schafer, D. Frankowski, J. Herlocker and S. Sen, "Collaborative Filtering Recommender Systems," *The Adaptive Web,* p. 291–324.

[131] M. Sharma and S. Mann, "A Survey of Recommender Systems: Approaches and Limitations," *,* 2013.

[132] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering,* vol. 17, p. 734–749, 6 2005.

[133] J. Bobadilla, F. Ortega, A. Hernando and A. Gutiérrez, "Recommender systems survey," *Knowledge-Based Systems,* vol. 46, p. 109–132, 7 2013.

[134] M. Papagelis, D. Plexousakis and T. Kutsuras, "Alleviating the Sparsity Problem of Collaborative Filtering Using Trust Inferences," *Lecture Notes in Computer Science,* p. 224–239, 2005.

[135] M. J. Pazzani and D. Billsus, "Content-Based Recommendation Systems," *The Adaptive Web,* p. 325–341.

[136] P. Lops, M. de Gemmis and G. Semeraro, "Content-based Recommender Systems: State of the Art and Trends," *Recommender Systems Handbook,* p. 73–105, 10 2010.

[137] L. O. Colombo-Mendoza, R. Valencia-García, A. Rodríguez-González, G. Alor-Hernández and J. J. Samper-Zapater, "RecomMetz: A context-aware knowledge-based mobile recommender system for movie showtimes," *Expert Systems with Applications,* vol. 42, p. 1202–1222, 2 2015.

[138] A. Felfernig, G. Friedrich, D. Jannach and M. Zanker, "Developing Constraint-based Recommenders," *Recommender Systems Handbook,* p. 187–215, 10 2010.

[139] D. Bridge, M. H. Göker, L. McGinty and B. Smyth, "Case-based recommender systems," *Knowledge Engineering Review,* vol. 20, no. 3, pp. 315-320, 2005.

[140] F. Lorenzi and F. Ricci, "Case-Based Recommender Systems: A Unifying View," *Lecture Notes in Computer Science,* p. 89–113, 2005.

[141] B. Smyth, "Case-Based Recommendation," *The Adaptive Web,* p. 342–376.

[142] R. Burke, "Hybrid Recommender Systems: Survey and Experiments," *User Modeling and User-adapted Interaction,* vol. 12, no. 4, pp. 331-370, 2002.

Appendix A - Acronyms

AmI	Ambient Intelligence
API	Application Programming Interface
CB	Content-based
CF	Collaborative-filtering
HCI	Human Computer Interaction
GUI	Graphical User Interface
IOT	Internet of Things
KB	Knowledge-based
SUS	System Usability Scale
TTS	Text-to-speech
UMUX	Usability Metric for User Experience
UX	User Experience
UI	User Interface

Appendix B - Recommendation systems

This section of the Appendix offers a literature review of the current state-of-the-art in recommendation techniques. Common recommendation techniques are discussed along with their benefits and issues.

Please refer to 3.2.3.3.3 An intelligent mechanism module for a discussion of the interaction design of ACOUSMA's recommender and 3.2.4 Auditory Display Recommender for the in-depth analysis of the rationale behind selecting a specific recommendation approach. The implemented recommender module is discussed in 4.3.3.3 Recommender while its usability evaluation is examined in 5.2 Case study.

Techniques for producing recommendations

Recommendation systems are classified based on a variety of recommendation techniques. According to [111] and [129], those include:

- Collaborative-filtering (CF)
- Content-based (CB)
- Knowledge-based (CF)
- Demographic-based
- Utility-based
- Context-aware
- Trust-aware based
- Fuzzy-based
- Social network-based
- Group-based
- Hybrid techniques

Here the focus will be set in common techniques, used by most recommender systems in a variety of domains. Those are CF, CB and KB and hybrid techniques. Other advanced techniques are used in specific contexts and for specific purposes, thus are not going to be focused upon.

Collaborative-filtering (CF)

Collaborative-filtering (CF) is one of the most popular recommendation techniques [111] and works by recommending items to the active user that other users with similar tastes have liked in the past. Similarity in preferences between users is calculated according to similarity found in the item rating history of users [130]. Similarity measures commonly used are correlation and cosine-based [131]. Ratings are used to measure the degree of interest for an item by a user. A widely used algorithm in the CF technique is that of k Nearest Neighbors [130] [132] [133] that is used to calculate which users have similar preferences with the active user and generate a list of recommendations. The recommendation process that is usually followed in collaborative filtering is:

1. Identify user ratings
2. Compute user similarity
3. Form neighborhood
4. Calculate predictions
5. Generate recommendation results

The major drawbacks of this technique is the ‘new user, new item’ problem (or ‘cold start’ problem) and the ‘rating sparsity’ problem [134]. The cold-start problem occurs in scenarios where reliable recommendations cannot be generated due to lack of ratings either by a new user who has not yet rated any items or due to a new item that has not received any ratings [130] [132]. Rating sparsity occurs when only a few users have rated the same item, creating a gap where no overlap of rating preferences can be made with the target user [134].

Content-based (CB)

The CB technique is used to recommend items that are similar in terms of content features to the ones that a user has liked in the past. CB recommendation systems calculate similarity based on associated features in compared items [112] [135]. Content-based recommender systems use relatively simple retrieval models, such as keyword matching and Vector Space Model (VSM) with basic Term Frequency-Inverse Document Frequency (TF-IDF) weighting. TF-IDF is used to compute the overall importance of keywords in a document [136]. The main advantages of this type of recommenders are:

- User independence – the recommender does not rely on other profiles to function.
- Transparency – recommendations can be easily explained.
- New unrated item recommendation – unrated items can be recommended solely based on common features with other items.

Drawbacks of this technique are limited content analysis, overspecialization, lack of serendipity and the cold-start problem [111]. Limited content analysis refers to scenarios where useful and reliable information cannot be extracted from heterogeneous data formats (video, audio and images), thus directly affecting the quality of generated recommendations. Overspecialization occurs when a user receives recommendations only for items very similar to their preferences (what they liked before), hence they lack the feeling of serendipity – making discoveries of items that are not similar to what they usually like but are likely to be of interest to them. The cold-start problem occurs, as a high amount of ratings must be accumulated before quality recommendations can be produced.

Knowledge-based (KB)

Knowledge-based recommenders are systems that recommend items to users based on domain knowledge about how items meet user preferences

[137] [113]. Three types of knowledge must be established in this type of recommenders [132]:

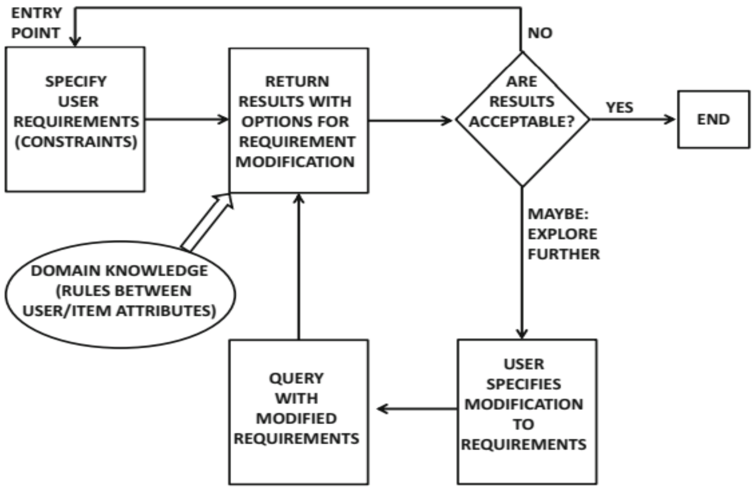
- About users
- About items
- About mappings between item and user's needs.

In contrast to the aforementioned approaches (CF & CB), this technique does not suffer from cold-start problems [132] or the rating sparsity problem, since the generated recommendations make use of domain knowledge and are not bound to user ratings. Due to these positive characteristics, knowledge-based approaches are suitable for hybridization with other techniques. The main drawback of KB approaches is the requirement of in-depth domain knowledge and knowledge engineering skills [112]. Knowledge-based (KB) recommenders can be categorized by the interaction methodology and knowledge base used to facilitate interaction [113]. Two types of knowledge-based recommenders exist in literature, namely Constrained-based and Case-based.

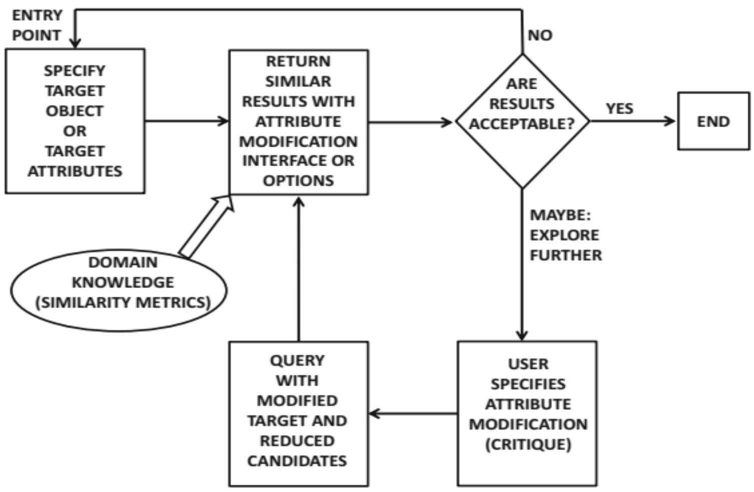
KB recommenders using the Constrained-based approach require users to specify constraints (e.g. an upper limit) on a target item's attributes. Domain specific rules are used to match user requirements and constrains to item attributes [138]. Through these rules, domain-specific knowledge is represented in the form of constraints or requirements defined by the user (e.g. "Houses with at least two floors that are built after the seventies"). The recommender may also internally create rules to relate item attributes to user attributes (e.g. "Younger buyers do not prefer houses that are built before the seventies"). Using those rules in conjunction with domain specific rules, specific queries are created (e.g. floors>=2 AND year >= 1970) that are used to filter search results and produce recommendations. A drawback of the constraint-based approach is that some queries may produce an empty set of results. For that, KB recommenders enable exploration and adjustment of results by allowing users to add or relax constrains after a list of recommendations is displayed. Repeatedly doing so, users interactively reach to desired results.

Case-based KB recommendation systems, allow users to create specific target cases that the system tries to match, providing similar results [139] [140] [141]. Similarity metrics are defined upon item attributes and are used by the system to produce recommendations. Therefore, similarity metrics incorporate domain-knowledge and must be effectively designed. Using similarity metrics, the system returns recommended items, which approximate or directly match the user's target case. If results are not satisfactory to the user, they can adjust the target case using critiquing methods. Critiquing methods provided by Case-based recommenders allow users to either specify directional or replacement critiques. Directional critiques adjust a target case by defining a filtering adjective towards provided results (e.g. "a – cheaper - house"), while replacement critiques replace target case attributes (e.g. "house of – different color –"). At any given time, a user can define one or multiple critiques towards recommendations. Through such interactive exploration of the item-space (critiquing cycles), users can reach the desired results. Successful results are held in a case history and are used to create a user profile and personalize the critiquing mechanism, as well as the recommendations.

Both Case-based and Constrained based approaches offer similar interactive processes to provide recommendations. An overview is provided in [113] and is shown in Figure 57. The difference between the two, lies in the way users specify their recommendation requirements. In the Constraint-based approach, users must define constraints that are domain-specific rules to filter and adjust recommendation results, i.e. in an online hardware store "a solid state drive with >= 500gb storage space". This entails that users have a general idea of the recommendation domain in order to efficiently reach the desired results. Often, however, users are not able to state their exact requirements in a complex product domain, although through exploration and interaction with the system, users can learn about the domain and receive meaningful recommendations. In the Case-based approach, recommendation results are produced by determining a target query with optimal requirements that may then be modified through user interaction and critiquing. A more conversational approach is used for reaching the desired results with users stating with simple terms how recommendation results should change (e.g. "a bigger solid state drive" or "a house with a spacious living room"). Therefore, users can explore a complex domain without having specific domain knowledge. Of course, this requires careful design of the interaction process and critiquing mechanism.



(a) Constraint-based interaction



(b) Case-based interaction

Figure 57 - Overview of interactive process in Knowledge-based recommenders [113]

Hybrid techniques

Recommendation systems have been developed that use the hybridization approach, combining features of two or more recommendation techniques with the goal of benefiting from each technique's strengths and improving the overall performance [112]. Using a hybrid approach can aid overcome limitations introduced by individual techniques and produce recommendation of higher quality [142].