DEPARTMENT OF
COMPUTER SCIENCE

NUNO RENATO NASCIMENTO GUERREIRO

BSc in Computer Engineering Sciences

CONSTRUCTING AN AUDITORY NOTATION IN SOFTWARE ENGINEERING: UNDERSTANDING UML MODELS WITH VOICE AND SOUND

MASTER IN COMPUTER SCIENCE

NOVA University Lisbon
March, 2023

DEPARTMENT OF
COMPUTER SCIENCE

CONSTRUCTING AN AUDITORY NOTATION IN SOFTWARE ENGINEERING: UNDERSTANDING UML MODELS WITH VOICE AND SOUND

NUNO RENATO NASCIMENTO GUERREIRO

BSc in Computer Engineering Sciences

Adviser: Vasco Miguel Moreira do Amaral
Associate Professor, FCT-NOVA

Examination Committee

Chair: Hervé Miguel Cordeiro Paulino
Associate Professor, FCT-NOVA

Adviser: Vasco Miguel Moreira do Amaral
Associate Professor, FCT-NOVA

Member: Jácome Miguel Costa da Cunha
Associate Professor, FEUP

MASTER IN COMPUTER SCIENCE

NOVA University Lisbon
March, 2023

Constructing An Auditory Notation in Software Engineering: Understanding UML Models With Voice And Sound

This document was created with the (pdf/Xe/Lua)LaTeX processor and the NOVAthesis template (v6.9.5) [1].

Dedicated to "avô Tomécas", for kindling my relentless desire to learn and remain curious. You are always in my mind. To my family and friends.

Acknowledgements

I would like to express my gratitude for all the support and encouragement during my academic path, precious for the realisation of this work. This path has not always been easy, but I emerge on the "other side" a more fulfilled and well-rounded person.

First of all, my sincere thanks to my supervisor, Professor Dr Vasco Miguel Moreira do Amaral, for placing his trust in my work and competencies, constantly pushing me towards the achievement of promising and pioneering work.

I want to specifically thank my colleagues, who became much more than that: Rafa, Pedrocas, Diogo, Camponês, António, Valente, Martinha, Pinto, Almeida, Gameiro, Hugo, Carol, and all the amazing people that, if I keep going, this page won't be enough to mention. There is never a dull moment with you all, and even in the hardest of moments, there was always a reason to laugh. As they say: "when the going gets tough, the tough get going". I will always be eternally grateful for your readily available presence and tireless help, regardless of the time of day, the constant exchange of knowledge and the bonds we have made. Thank you for opening my horizons and unknowingly encouraging me to constantly strive to match your expertise and standards, without you it would not have been possible.

To my girlfriend, for her patience, kindness, affection and ability to always come up with the right words. You're the best.

To all my friends, band-mates and especially to my lifelong pals André and Daniel, for their motivation and friendship, giving me the strength to carry on. Thank you for holding me accountable to always give my best and achieve my full potential and beyond.

To my brothers, whose advice and assistance have been crucial throughout my life.

Last but not least, I want to thank my grandparents and parents, who despite occasionally not understanding the challenges and struggles I faced, always had the best intentions and my interests in mind. Thank you for the values you instilled in me and for your cooperation, allowing me to give this effort my whole attention.

“A gem cannot be polished without friction, nor a man perfected without trials.” (Seneca)

Abstract

Sound is crucial to how we interact with the world around us, providing feedback and contextualising information. However, when discussed in software, it is not given the same importance as vision. Neglecting this channel results in untapped possibilities that could enhance the user experience, and the exclusion of many visually impaired people from activities related to software engineering, since the visual notations and well-accepted tools in this field are not supportive of audio, such as UML.

Several technologies have been developed and integrated into prototypes. Still, it became evident during our research that, among other factors, their usability is greatly impacted by unsuitable choices of sound and voice symbolism, as well as wrong interaction dialogues that can become too cumbersome to be used. Sound should be analysed in the context of software engineering, as it has the unexplored potential to significantly contribute to how we construct and interact with the software while allowing blind and visually impaired people to be part of these activities.

For this purpose, we are interested in building a foundational framework to substantiate decisions when designing an auditory notation, and a tool that performs diagrammatic readings in UML, intended to validate these proposals. Supported by the semiotics of the audible field and music symbology, combined with the insights provided by Moody's *Physics of Notations*, the findings of other research and developed tools concerning these topics, along with experimental studies that we carried out and are presented in this document.

We believe that this work can be instrumental in creating a structured and intuitive auditory notation for software engineering, complemented by a tool built in the right direction for accessibility. Furthermore, it is an approach that aims to join both the visual and hearing senses in a manner that benefits a large and diverse population of experienced software engineers and novices alike, heightening the visual notation in the process.

Keywords: foundational framework, auditory notation, diagrammatic readings, guiding principles, software engineering, UML, modelling software, the physics of notations, *SoundUML*, *Model-By-Voice*, *Modelio*

Resumo

O som é crucial para a forma como interagimos com o mundo ao nosso redor, fornecendo *feedback* e contextualizando informação. No entanto, quando este é discutido em software, não lhe é dada a mesma importância que à visão. Negligenciar este canal resulta em possibilidades inexploradas que poderiam melhorar a experiência do utilizador, e na exclusão de pessoas com deficiências visuais de actividades relacionadas com engenharia de software, uma vez que as notações visuais e as ferramentas bem aceites neste domínio não suportam áudio, como é o caso do UML.

Diversas tecnologias foram desenvolvidas e integradas em protótipos, mas durante a nossa pesquisa tornou-se evidente que, para além de outros factores, a sua usabilidade é bastante impactada por escolhas inadequadas de simbolismo relativamente ao som e voz, bem como diálogos de interação errados que se podem tornar demasiado incómodos para serem utilizados. O som deve ser analisado no contexto de engenharia de software, pois tem o potencial de contribuir para a forma como construímos e interagimos com software, permitindo ainda que pessoas com deficiências visuais façam parte destas atividades.

Para esta finalidade, queremos construir uma *framework* para fundamentar decisões na construção de uma notação auditiva, em conjunto com uma ferramenta que efectua leituras diagramáticas em UML, destinada à validação destas propostas. Tendo por base a compreensão das semióticas do domínio audível e simbologia musical, combinado com os conhecimentos fornecidos pelo *Physics of Notations* de Moody, as descobertas de outros trabalhos e ferramentas desenvolvidas neste âmbito, juntamente com estudos experimentais que realizámos, apresentados neste documento.

Acreditamos que este trabalho possa ser fundamental na criação de uma notação auditiva estruturada e intuitiva para engenharia de software, complementada por uma ferramenta construída na direção certa para a acessibilidade. Além disso, é uma abordagem que visa a união dos sentidos de visão e audição, de forma a beneficiar uma ampla e diversa população de engenheiros de software experientes e novatos, elevando a notação visual no processo.

Palavras-chave: notação auditiva, leituras diagramáticas, princípios orientadores, engenharia de software, software de modelação, *SoundUML*, *Modelio*

Contents

List of Figures **xii**

List of Tables **xv**

Acronyms **xviii**

1 Introduction **1**

1.1 Context 1

1.2 Motivation 2

1.3 Problem Statement 3

1.4 Predicted Contributions 3

1.5 Document Structure 4

2 Background **6**

2.1 Semiotics 6

2.1.1 Peirce's Categories 7

2.1.2 The Three Trichotomies 8

2.1.3 Semiotics of the Audible Field 10

2.2 Model-Driven Software Engineering 14

2.2.1 Metamodels 14

2.2.2 The Model-Driven Acronyms 15

2.2.3 Modelling Languages 15

2.2.4 Unified Modelling Language (UML) 16

2.3 Voice Computing 17

2.3.1 Natural Language Processing 17

2.3.2 Voice Synthesis 18

2.3.3 Speech Recognition 18

3 Systematic Literature Review **21**

3.1 Establishment Of The Research Question 21

3.2 Search Strategy and Selection of Primary Studies 22
3.3 Data Extraction and Results 23
3.3.1 Sound in a Software Context 23
3.3.2 UML Related Solutions For The Visually Impaired 26
3.3.3 Speech Recognition - Software Modelling Using Voice Commands 27
3.4 Final Observations 30

4 Related Work 31
4.1 The "Physics" of Visual Notations 31
4.2 Other Relevant Works 33
4.2.1 Audible Browser - UML Diagrams For Blind Programmers . . . 34
4.2.2 ModelByVoice Tool 35

5 Principles For The Construction Of Auditory Notations & Useful Guidelines 36
5.1 Revisiting The "Physics" Of Notations: Principles For An Auditory Notation 36
5.2 Design Guidelines 38
5.2.1 Guidelines For Sound Design In The Context Of Software 38
5.2.2 Scientific Method For Sound Design - A Conceptual Framework 41
5.2.3 Guidelines For Voice User Interfaces 42

6 Detecting Patterns In UML Diagrammatic Readings: An Experimental Study 44
6.1 Planning 44
6.1.1 Goals 44
6.1.2 Participants 45
6.1.3 Tasks 45
6.1.4 Defining The Hypothesis 45
6.1.5 Preparing The Evaluation Session 46
6.1.6 Experimental Material 49
6.2 Procedure Of The Evaluation Sessions 49
6.3 Results 51
6.3.1 Transcription Procedure 52
6.4 Discussion Of Results 52
6.4.1 Class Diagram Section 53
6.4.2 State Machine Diagram Section 58
6.4.3 Summary 62
6.5 Proposals For The Construction Of Diagrammatic Reading Tools 64
6.5.1 Proposals For Class Diagrams 64
6.5.2 Proposals For State Machine Diagrams 66
6.6 Threats To Validity 66

7 A Catalogue Of Sounds For UML Class Diagrams 68
7.1 Defining The Catalogue Of Sounds 68

7.2 Specifying The Catalogue Of Sounds 69
7.3 Experimental Study 74
7.3.1 Planning 75
7.3.2 Procedure Of The Evaluation Sessions 81
7.3.3 Results 83
7.3.4 Discussion Of Results 92
7.3.5 Threats To Validity 98

8 Validation In The Context Of Use: Auditory Diagrammatic Readings 99
8.1 SoundUML: A Tool For Auditory Readings Of UML Class Diagrams . . 99
8.2 SoundUML Architecture 100
8.3 Essential Tools For The Development Of SoundUML 102
8.3.1 Eclipse Modeling Framework (EMF) 102
8.3.2 Modelio 103
8.3.3 Maven 104
8.3.4 JFace 104
8.3.5 Javax - Clip API and LineListener 105
8.3.6 FreeTTS 105
8.4 Implementation Of SoundUML 106
8.4.1 Module Development Structure: A General Overview 106
8.4.2 Project Overview 107
8.5 Installing SoundUML: A Step-By-Step Guide 113
8.6 Using SoundUML 114
8.7 Experimental Study: Evaluating The Diagrammatic Readings & SoundUML 115
8.7.1 Planning 116
8.7.2 Procedure Of The Evaluation Sessions 120
8.7.3 Results 122
8.7.4 Results On The Evaluation Of SoundUML 132
8.7.5 Discussion Of Results 139
8.7.6 Threats To Validity 147
8.8 Identified Shortcomings 149

9 Conclusion 151
9.1 Summary 151
9.2 Contributions 153
9.3 Shortcomings 154
9.4 Future Work 155

Bibliography 158

Appendices

A Appendix 1 - Detecting Patterns In UML Diagrammatic Readings: An Experimental Study 163
A.1 UML Diagrammatic Reading: Experimental Study 164

B Appendix 2 - A Catalogue Of Sounds For UML Class Diagrams 175
B.1 A Catalogue Of Sounds: Experimental Study 176

C Appendix 3 - Validation In The Context Of Use 197
C.1 SoundUML Class Diagram 198
C.2 UML Class Diagrams From The Experimental Study 199
C.3 Support Document 201
C.4 Faculty Diagram Form 204
C.5 Library Diagram Form 207
C.6 GUI Diagram Form 211
C.7 Personal Info + Usability + General Feedback Form 214

D Appendix 4 - Scientific Paper 225

Annexes

I Annex 237

List of Figures

2.1 Semiotic Triangle 7
2.2 Relationship between the different Model-Driven Acronyms. Taken from [7] 15
2.3 UML models taxonomy. Taken from [7] 16
2.4 A functional diagram of a TTS system. Adapted from [30] 18
2.5 A typical speech-recognition system. Taken from [31] 19

4.1 ModelByVoice II Architecture Overview. Taken from [11] 35

6.1 The document's introductory text, presented to the participants. 48
6.2 Example of a set of Class diagram questions in the document presented to the participants. 51
6.3 Class Diagram from Question 1. Taken from [54]. 53
6.4 Class Diagram from Question 2. 54
6.5 Class Diagram from Question 3. Adapted from [54]. 58
6.6 State Machine Diagram from Question 1. Adapted from [54]. 59
6.7 State Machine Diagram from Question 2. Adapted from [55]. 61

7.1 Semiotic Triangle For The Catalogue Of Sounds 69
7.2 Typical view for how the sounds were created in *REAPER*, along with the homepage of the website used to download the free basic sounds. 77
7.3 The form's introductory text, presented to the participants. 78
7.4 Structure of a typical set of questions in the form. 82
7.5 Results regarding the participants' gender. 83
7.6 Results regarding the participants' hearing condition. 84
7.7 Results regarding the participants' level of understanding of UML. 84
7.8 Results regarding the participants' last contact with UML. 84
7.9 Results regarding the participants' context of usage of UML. 85
7.10 Results of the first question. 85
7.11 Results of the second question. 86
7.12 Results of the third question. 86

7.13 Results of the fourth question. 87
7.14 Results of the fifth question. 87
7.15 Results of the sixth question. 88
7.16 Results of the seventh question. 88
7.17 Results of the eighth question. 89
7.18 Results of the ninth question. 89
7.19 Results of the tenth question. 90
7.20 Results of the eleventh question. 90
7.21 Results of the relevance of each proposed principle for Auditory notations, as rated by participants. 91
7.22 Results regarding how participants rated the overall relevance of the principles for constructing auditory notations. 92

8.1 General outline of SoundUML's architecture. 101
8.2 The *SoundUMLModule* Life Cycle. Adapted from [67] 107
8.3 Summarised SoundUML's Class Diagram, detailing the most important classes, their respective relations and packages. 108
8.4 SoundUML Install Guide: Steps 1 to 3 - Adding SoundUML to the program Modelio. 114
8.5 SoundUML Install Guide: Steps 4 and 5 - Deploying SoundUML to a specific user project. 114
8.6 SoundUML Install Guide: Steps 6 to 8 - Deploying SoundUML to a specific user project. 115
8.7 Example of a typical SoundUML window performing a diagrammatic reading, showing the reading of an abstract class. 115
8.8 Results regarding the participants' gender. 123
8.9 Results regarding the participants' hearing condition. 123
8.10 Results regarding the participants' level of understanding of UML. 123
8.11 Results regarding the participants' last contact with UML. 124
8.12 Breakdown of how the participants experienced each version of the Faculty diagram. 124
8.13 Results of Question 1 From The Faculty Diagram, for each of the versions . 125
8.14 Results of Question 2 From The Faculty Diagram, for each of the versions . 125
8.15 Results of Question 3 From The Faculty Diagram, for each of the versions . 126
8.16 Breakdown of how the participants experienced each version of the Library diagram. 127
8.17 Results of Question 1 From The Library Diagram, for each of the versions . 127
8.18 Results of Question 2 From The Library Diagram, for each of the versions . 128
8.19 Results of Question 3 From The Library Diagram, for each of the versions . 128
8.20 Results of Question 4 From The Library Diagram, for each of the versions . 129

8.21 Breakdown of how the participants experienced each version of the GUI diagram. 130
8.22 Results of Question 1 From The GUI Diagram, for each of the versions . . . 130
8.23 Results of Question 2 From The GUI Diagram, for each of the versions . . . 131
8.24 Results of Question 3 From The GUI Diagram, for each of the versions . . . 131
8.25 Results of the first SUS question concerning the *SoundUML* tool. 132
8.26 Results of the second SUS question concerning the *SoundUML* tool. 133
8.27 Results of the third SUS question concerning the *SoundUML* tool. 133
8.28 Results of the fourth SUS question concerning the *SoundUML* tool. 133
8.29 Results of the fifth SUS question concerning the *SoundUML* tool. 134
8.30 Results of the sixth SUS question concerning the *SoundUML* tool. 134
8.31 Results of the seventh SUS question concerning the *SoundUML* tool. 134
8.32 Results of the eighth SUS question concerning the *SoundUML* tool. 135
8.33 Results of the ninth SUS question concerning the *SoundUML* tool. 135
8.34 Results of the tenth SUS question concerning the *SoundUML* tool. 135
8.35 Results of how participants rated their experience with the *SoundUML* tool. 136
8.36 Results on whether participants thought that the readings performed by the tool were intuitive enough to understand the diagrams. 136
8.37 Results on the type of reading preferred by the participants. 137
8.38 Results regarding the participants' opinion on whether SoundUML could be useful to complement and enrich the existing visual notation. 137
8.39 Results concerning the participants' opinion on if the SoundUML tool could be useful for people with visual impairments who want to carry out UML modelling. 138
8.40 Results regarding whether participants would recommend the SoundUML tool to visually impaired people. 139
8.41 Box plot with SoundUML's SUS score. 145

A.1 Full sized class Diagram from Question 2. 174

C.1 Summarised SoundUML's Class Diagram, detailing the most important classes, their respective relations and packages. 198
C.2 Visual Version of Diagram A - Faculty, as constructed and presented in Modelio. 199
C.3 Visual Version of Diagram B - Library, as constructed and presented in Modelio. 199
C.4 Visual Version of Diagram C - GUI, as constructed and presented in Modelio. 200

I.1 Interactions between the principles proposed by Moody. Taken from [8] . . 237
I.2 Auditory Variables defined by Krygier. Taken from [52] 238
I.3 First version of the iconography for the framework's components and their respective categories. Taken from [46] 238

List of Tables

2.1 meaning of triadic sign terms in semiotic theory and in popular language . 7
2.2 Peirce's Categories and their relation to the sign components and trichotomies 9
2.3 The Ten Classes Of Signs. Adapted from [18] 10
2.4 Semiotics Of The Audible Field. Adapted from [20]. 12
2.5 Listener's cues for understanding product sound. Taken from [4]. 13

3.1 Inclusion & Exclusion Criteria. 22
3.2 Research query building. 23

5.1 The 9 principles proposed by Moody with our auditory equivalent proposal. 37

7.1 Catalogue Of Sounds for UML Class Diagram Elements 70
7.2 The Catalogue Of Bad Practices with the Unsatisfied Auditory Principles . 80
7.3 Summary of the results regarding the choices for the sounds. 91
7.4 Average score regarding the relevance of each principle, as rated by the participants. 92
7.5 Option with the most votes in the questions regarding the catalogues of sounds. 93

8.1 Table with the 12 possible combinations of the order in which the diagrams were presented. 121
8.2 Table with the results of Question 1 From The Faculty Diagram 125
8.3 Table with the results of Question 2 From The Faculty Diagram 126
8.4 Table with the results of Question 3 From The Faculty Diagram 126
8.5 Table with the results of Question 1 From The Library Diagram 127
8.6 Table with the results of Question 2 From The Library Diagram 128
8.7 Table with the results of Question 3 From The Library Diagram 129
8.8 Table with the results of Question 4 From The Library Diagram 129
8.9 Table with the results of Question 1 From The GUI Diagram 130
8.10 Table with the results of Question 2 From The GUI Diagram 131
8.11 Table with the results of Question 3 From The GUI Diagram 132

8.12 Percentage of correct answers (Accuracy) for the auditory versions of the diagrams. 140
8.13 Percentage of correct answers (Accuracy) for the 3 versions of the diagrams. 142
8.14 Calculated Mean Accuracy of the answers for each of the versions of the diagrams. 144
8.15 Descriptive Statistics Of SoundUML's System Usability Scale 145

Acronyms

API Application Programming Interface 19

DSLs Domain-Specific Languages 15
DSP Digital Signal Processing 18

EMF Eclipse Modeling Framework 102, 104

GPMLs General-Purpose Modelling Languages 16
gRPC Google Remote Procedure Call 19
GUI Graphical User Interface 109

MBE Model-Based Engineering 15
MDA Model-Driven Architecture 15
MDD Model-Driven Development 15
MDE Model-Driven Engineering 15
MDSE Model-Driven Software Engineering 14

NLP Natural Language Processing 17

OMG Object Management Group 15

POM Project Object Model 106

SLR Systematic Literature Review 21
STT Speech-to-text 18
SUS System Usability Scale 119, 132
SWT Standard Widget Toolkit 104

TTS Text-to-speech 18

UML Unified Modelling Language 1, 16

VDM Voice-Driven Modelling 29

VUIs Voice-User Interfaces 4, 20, 153

1

Introduction

1.1 Context

Hearing is one of the five human senses and, as such, is crucial to how we interact with the world around us. Every day, there are numerous situations where sound is employed, ranging from communication through spoken language to entertainment, art, or even small cues that provide feedback or contextualise information. Blake and Cross explore this relation with sound more thoroughly, as humans utilise and experience sound in varied, complex and dynamic forms. They state that "Research in linguistics and music psychology continues to suggest that humans have developed a range of sensitivities to sound through the production and perception of affective nonverbal vocalisations, speech, and music" [2].

In 2021, the International Agency for the Prevention of Blindness released a compilation of the latest eye health data. The report revealed that approximately 295 million people were living with moderate-to-severe visual impairment, of which 43 million people were blind [3]. Sound is inevitable, and its usage is even more significant when vision is affected. However, it is not treated with the same importance as vision, mainly when discussed in software. Blauert affirms that product-sound design became an issue during its infancy, with sound being treated mostly as secondary. In addition, sound design itself did not have quite the same expansion and attention as other design areas like visual design [4], while Kenwright stresses that sound should be designed, whether intentionally or not [5].

Visual notations are well accepted in Computer Science [6] and the tools that accommodate this diagrammatic communication are not supportive of sound in the context of said notations. This negligence regarding sound leads to the exclusion of many visually impaired or blind people from activities related to computer education programs and the software development industry. One such example is the Unified Modelling Language (UML). Created as a way to standardise notational systems in software design, UML is a general-purpose, widely adopted modelling language and, as such, an integral part of the software development process [7].

Sound should be analysed in the context of software engineering, as it has the unexplored potential to significantly contribute to how we construct and interact with models and software in general, while also allowing blind and visually impaired people to be integrated into these activities.

1.2 Motivation

The main motivation is to bring diagrammatic languages, used as a means of abstraction in Software Engineering, from something based on the visual sense to the auditory one, as we believe this approach has unstudied capabilities, while also being potentially relevant for accessibility. The Automated Software Engineering group at the research institution NOVA LINCS at NOVA School of Science and Technology started a project to automatically generate language editors that deal with domain-specific languages using sound as the input and output channel. However, as the main focus was on the technological solution, there was not much effort in systematising knowledge or guiding principles that could substantiate the decisions taken concerning the usage of sound.

Other works examined in this document also began to approach this problem mainly from a technological standpoint through the development of tools, without considering the fundamental part of establishing and structuring good symbolism for sounds, resulting in bad choices for how audio was used. Thus, to fully realise one of the main goals of this work and assemble an effective framework, it is necessary to understand what influence different acoustic cues have through their cataloguing and systematisation. It is also shown that organised sounds such as music, play a role in memory creation and recollection, and auditory cues influence human behaviour [2]. This is relevant, as we also intend to propose an auditory reading of UML class diagrams that must be cognitively manageable, realised through creating a user-friendly tool that does not require much effort on behalf of its users.

The most popular and comprehensive recommendation for assuring cognitively effective notations [6] is *The "Physics" of Notations* [8], in the visual realm. This research by Daniel Moody contains nine principles based on the perceptual properties of notations, focusing on addressing and raising awareness about the importance of specific visual representation issues that software engineering researchers and notation designers have overlooked. The work presented in this document seeks to give continuity to the previously conducted research that has addressed some of the topics here considered, while analysing and discussing their findings, with the main focus being the construction of a framework that among other aspects, consists in establishing auditory principles, similarly to Moody's aforementioned study.

1.3 Problem Statement

In the aforementioned ongoing project, the usage of model-driven technology and approaches to building software prototypes that work as sound editors of diagrammatic languages were previously explored. Other technologies were also investigated and integrated into proof-of-concept prototypes. However, in usability studies, it became evident that along with non-functional requirements of the processing speed in these tools [9], and problems of accuracy in speech recognition tools [9] [10] [11], the usability of such languages is greatly impacted by unsuitable choices of sound and voice symbolism, as well as wrong interaction dialogues that can become too cumbersome to be used.

We are interested in building a foundational framework that can substantiate decisions when designing an auditory notation. This framework consists in establishing auditory principles, detecting patterns in diagrammatic readings, and creating a catalogue of sounds for diagram elements. This will be based on the semiotics of the audible field and music symbology, combined with the insights provided by Moody's *Physics of Notations*, the findings of other research and developed tools regarding these topics, along with experimental studies that will be carried out. Finally, to validate this framework in the context of use, we also intend to develop a tool that performs auditory readings of UML Class diagrams.

We believe this framework could be an appropriate first step to achieving systematised knowledge concerning these topics while offering greater accessibility regarding the inclusion of visually impaired people in activities related to software engineering. Furthermore, it is an approach that aims to join both the visual and hearing senses in a manner that achieves the maximum benefit for a large and diverse population of experienced software engineers and novices alike, heightening the visual notation in the process.

Ultimately, we can announce our research question as **Which guiding principles should we consider to use sound and voice as modelling notations, applied in the context of modelling with UML?**

1.4 Predicted Contributions

The following is a brief overview of what will be done to tackle the previously described problem, as well as the expected contributions:

- **Analysis Of Semiotics**, from the perspective of computer engineering, which will allow for the justification of certain sound design choices when creating an auditory notation.

- **Systematic Literature Review concerning the topics related to the construction of auditory notations in software engineering**, enabling the identification, evaluation and interpretation of all available research relevant to the topics of this thesis, while offering insights that can be applied to this work.

- **Elaboration of principles for constructing auditory notations**, consisting in the transposition of the principles proposed by Moody for visual notations, in a form that is analogous to sound. This will allow for the construction of structured and more intuitive auditory notations and substantiate the choices made in the creation of a catalogue of sounds for diagram elements.

- **Collection of Design Guidelines and Frameworks for sound design in the context of software and Voice-User Interfaces (VUIs)**. Having these summarised in one locale facilitates their examination and further discussion while helping to justify some of the decisions made during the implementation phase along with the created catalogue of sounds.

- **Proposals for the construction of tools that perform auditory readings of diagrams**, through an experimental study, focused on detecting patterns regarding how different people carry out diagram readings, by observing the most common actions among the participants. The results serve as the basis for the auditory reading of our prototype.

- **Construction of a catalogue of sounds for UML Class diagram elements**, based on the study of semiotics, the proposed auditory principles and collected guidelines. This catalogue was followed by an experimental study with the goal of qualitatively determining whether the construction of a catalogue that follows the elaborated auditory principles and semiotics is more effective than one that disregards them.

- **Auditory readings of UML Class diagrams, realised through the development of a proof of concept tool**, implementing in practice, the conclusions drawn from the constructed framework and the conducted experimental studies. This tool has the intended purpose of performing the reading of UML Class diagrams to its users, through the usage of voice synthesis and sound playback, and aims to enrich the existing visual notation, while contributing to the inclusion of blind and visually impaired people in activities related to software engineering.

1.5 Document Structure

This document is divided into nine unique chapters, each one containing the following information:

- **Introduction** - In Chapter 1, the problem and its general topics are considered and explored, in addition to the context, motivations and predicted contributions.

- **Background** - Chapter 2 provides context to the information examined in this document, discussing in depth the general topics related to this work. Furthermore, semiotics is also analysed from the perspective of computer engineering in this chapter.

- **Systematic Literature Review** - Chapter 3 focuses on elaborating a Systematic Literature Review and discussing its findings, aiming to identify and evaluate and interpret all relevant research to the topics of this thesis.

- **Related Work** - Chapter 4 addresses the works that serve as the primary sources of insight related to the topics of this thesis more concretely, while their key ideas and limitations are analysed.

- **Principles For The Construction Of Auditory Notations & Useful Guidelines** - Chapter 5 contains the elaboration of the auditory principles and design guidelines that will serve as the foundation for our intended framework and subsequent work of this dissertation.

- **Detecting Patterns In UML Diagrammatic Readings: An Experimental Study** - In Chapter 6, an experimental study was conducted to understand any emerging patterns regarding how different people carry out diagrammatic readings of UML Class and State Machine Diagrams. The planning, execution and results are discussed, along with the elaboration of proposals for the construction of tools that perform auditory readings of diagrams.

- **A Catalogue Of Sounds For UML Class Diagrams** - Chapter 7 proposes a catalogue of sounds for each of the different elements that constitute a UML Class diagram, following a more methodical approach, where the choices for sounds adhere to the proposed auditory principles and the study of the semiotics of the audible field. An experimental study was prepared as well, where another catalogue was defined, and its sounds were chosen arbitrarily, disregarding said auditory principles and semiotics. The planning, execution and results of this study are explained.

- **Validation In The Context Of Use: Auditory Diagrammatic Readings** - Chapter 8 seeks to validate and discuss the drawn conclusions regarding the created framework and consequently, our proposals for an auditory diagrammatic reading of UML class diagrams. We present the construction and implementation process of our developed prototype, a proof of concept named *SoundUML*, as well as an experimental study with the purpose of evaluating the effectiveness of our proposals and the tool's usability.

- **Conclusion** - Chapter 9 details the final conclusions by summarising the efforts of this dissertation, along with the achieved contributions, shortcomings and suggestions for future work.

2

Background

This section will establish the context of this work, providing general information about its topics. In Chapter 2.1, the study of signs and symbols and their signification is presented. Chapter 2.2 defines the various concepts of Model-Driven Software Engineering in the context of this work, while Chapter 2.3 details the aspects of the discipline of Voice Computing.

2.1 Semiotics

As previously mentioned, to assemble a useful notation, it is necessary to understand the influence different acoustic cues have on assigning meaning to sound effectively. Semiotics is the study of any activity that involves the usage of signs and symbols and their signification. At first glance, we might be inclined to believe that signs are merely what we routinely refer to as 'signs' in everyday life - such as road signs or pub signs - and that they lean towards the visual realm, but this is a misconception. Generally, signs can be understood as a "stand-in" representation for a particular concept. Signs can also be drawings, paintings, photographs, words, sounds, or body language. Examples of signs include *emojis* used in electronic communication, traffic lights and logos. The ones receiving the information need background knowledge to bridge the gap between the sign itself and the concept that it is meant to represent.

In semiotic theory, the sign is made up of the relation between three components: the **(semiotic) object** - what the sign represents/encodes, and can be anything thinkable, for example, a fact or a law; the **signifier** - the form that represents the denoted *object*, giving it meaning. It can be, for example, a word or image and is also sometimes called representamen. Finally, the **signified** - also called interpretant, is the concept that a *signifier* refers to. Essentially being what is evoked in mind - a mental concept [12].

With basis on the brief theoretical explanation presented here, along with the foundational work elaborated in Chandler's *Semiotics: The Basics* [12] and *Conceptualization, measurement, and application of semantic transparency in visual notations* [6], figure 2.1 aims to visually aid the brief explanation of these concepts, while table 2.1 details for each one of these terms, a concise semiotic meaning, a popular language meaning (how people colloquially perceive one of the terms), and a concrete example.

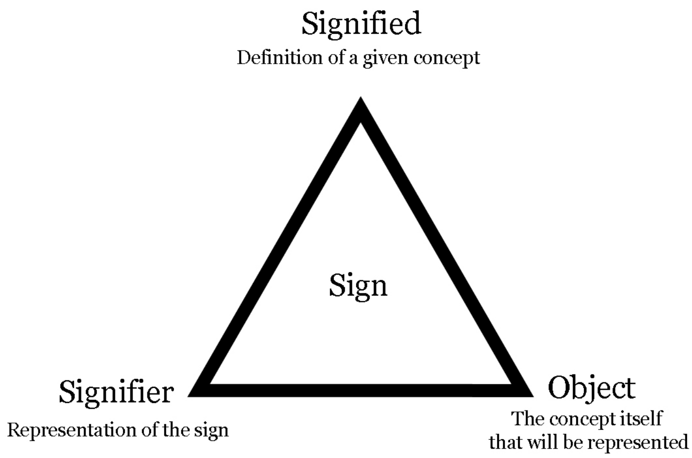


Figure 2.1: Semiotic Triangle

Table 2.1: meaning of triadic sign terms in semiotic theory and in popular language

Term	Semiotic Meaning	Popular language meaning	Example
Signifier	Also called representamen, is the form the sign takes, giving it meaning.	The terms "representation" and "depiction" are used as synonyms.	Drawing of a red heart
Signified	Also called interpretant, is the concept that the signifier refers to.	The terms "concept" and "definition" are used as synonyms.	A symbol of love and affection (Signifies the red heart)
Sign	The unity of what is represented (the object), how it is represented (signifier), and how it is interpreted (signified).	An object, quality, or event whose presence or occurrence indicates something else's probable presence or occurrence.	Logos, Traffic lights

2.1.1 Peirce's Categories

Charles Sanders Peirce further subdivided each of these three terms of semiosis into three different categories, calling them *cenopythagorean categories* [13]. Commonly referred to as Peirce's Categories or Peircean phenomenological categories of experience, they are distinguished as firstness, secondness and thirdness, with Peirce believing that these categories are necessary and sufficient to account for all of the human experience [14].

As Everaert-Desmedt explains [14], **firstness** is a conception of being that is independent of anything else, corresponding to emotional experience. For example, the general sensation of hurt, before wondering exactly whether the sensation comes from a headache, a burn or emotional pain. **Secondness** is the mode of being that is in relation to something else and corresponds to practical experience and intellectual categorisation. Peirce illustrates this concept through the example of how one may see a beam of light. In firstness, one may sense the fundamental experience of only the beam of light, while in secondness, one may perceive the beam of light as ’God’s creation’, which is a secondary, derivative interpretation [15] [16]. Lastly, **thirdness** corresponds to intellectual experience and is the category of thought, language, representation, and the process of semiosis. It belongs to the domain of rules and laws, bridging the signifier and the object. Everaert-Desmedt states that the law of gravity is an example of thirdness, as it allows us to predict that each time a stone is dropped, it will fall on the ground [14].

2.1.2 The Three Trichotomies

Semioticians differentiate amongst various types of signs, emphasising that signs differ in how arbitrary/conventional they are [17]. Specifically, semioticians distinguish Peirce’s categories mentioned above of firstness, secondness and thirdness by what stands as the sign (signifier), the signifier-object relations, and how the signified denotes the object. These are called the three trichotomies [14].

2.1.2.1 The First Trichotomy - The Signifier

The signifier can be a **qualisign** (firstness), meaning a quality that functions as a sign, qualitative; a **sinsign** (secondness) - a specific singular thing or event that has both spatial and temporal qualities that functions as a sign; or a **legisign** (thirdness), which is a conventional sign, lawlike. Everaert-Desmedt exemplifies that legisigns can be passwords, insignias or words of a language. Legisigns cannot act until being embodied as sinsigns (which are "replicas"). For example, the article "the" is a legisign in the English language, and it will occur many times in a text. In all of these occurrences, it is the same legisign, but every single instance is a replica. The replica is a sinsign, but also includes qualisigns. Qualisigns can be the intonation of the oral replica or the shape of the letters of the written replica [14].

2.1.2.2 The Second Trichotomy - The Object

According to Everaert-Desmedt, a signifier can refer to its object by virtue of Peirce’s categories. As stated by Pierce, this trichotomy emphasises the different ways in which the sign refers to its object. As such, a sign can be an **icon** (firstness) - where the sign is perceived as resembling or imitating the object, being similar in having some of its qualities. An icon may have as its signifier a qualisign, a sinsign or legisign. Examples of icons are sound effects in radio drama, a portrait or a scale model;

A sign can also be an **index** (secondness) - a mode in which the sign is affected by the object, the link between the signifier and signified can be observed or inferred, with both being directly connected in some way. This can be ascertained in examples like ’natural signs’ - such as smoke, thunder, footprints or echoes; medical symptoms - pain, a rash, pulse rate, or ’signals’ - like a knock on a door or a phone ringing.

Lastly, a sign can also be a **symbol** (thirdness) - a mode in which, as Chandler affirms, the signifier does not resemble or mirror the signified [12]. This means that their relationship is arbitrary and must be agreed upon and learned. Examples of this include national flags, Morse code, or punctuation marks [6].

2.1.2.3 The Third Trichotomy - The Signified

In this trichotomy, whenever a sign is understood in terms of suggested qualities an object may have, a signified is created that qualifies as a **rheme** (firstness). Examples are not straightforward, but they can be thought of as unsaturated predicates like "— is a dog" or "— is happy" and so on. The sign is a **dicisign** (secondness) if it is understood as referring to the existential features of an object. These signs can be thought of as saturated predicates or propositions like "Fido is a dog" or "Larry is happy". Finally, the sign is an **argument** (thirdness) if a signified is determined by focusing our understanding on some conventional or lawlike features employed in signifying the object. In sum, a rheme can be thought of as an unsaturated predicate, the dicisign as a proposition, and the argument as an argument or rule of inference [18]. To summarise these concepts, table 2.2 was devised, through this description and its sources, Peirce's works [16] and Mittelberg's efforts [19].

Table 2.2: Peirce's Categories and their relation to the sign components and trichotomies

Categories	**Typical Characterisation**	**As Universe of Experience**	**1st Trichotomy**	**2nd Trichotomy**	**3rd Trichotomy**
Firstness	Quality of feeling	Ideas, chance, possibility	qualisign	icon	rheme
Secondess	Reaction, experience	Brute facts, actuality	sinsign	index	dicisign
Thirdness	Representation, mediation	Habits, laws, necessity	legisign	symbol	argument

Each sign is then classifiable as some combination of each of its three elements, that is, as either one of the three types of signifier/representamen, plus one of the three types of object, plus one of the three types of signified/interpretant. Initially, this seems to result in twenty-seven possible combinations, but in semiosis, there is a hierarchy principle amongst the categories. Firstness includes nothing other than itself, whereas secondness includes firstness, and thirdness includes both secondness and firstness. This means that a signifier (a first) cannot refer to an object (a second) from a higher category, and the signified (a third) cannot belong to a category higher than that of the object [14], there are, in fact, only ten types of signs. Table 2.3 details these ten permissible combinations, with a brief example for each one [18].

Table 2.3: The Ten Classes Of Signs. Adapted from [18]

Signifier (Representamen)	**Object**	**Signified (Interpretant)**	**Examples [16]**
Qualisign (1)	Icon (1)	Rheme (1)	"A feeling of red", General sensasation of hurt
Sinsign (2)	Icon (1)	Rheme (1)	"An individual diagram"
Sinsign (2)	Index (2)	Rheme (1)	"A spontaneous cry", An involuntary shout
Sinsign (2)	Index (2)	Dicisign (2)	"A weather vane"
Legisign (3)	Icon (1)	Rheme (1)	Onomatopoeia
Legisign (3)	Index (2)	Rheme (1)	"A demonstrative pronoun"
Legisign (3)	Index (2)	Dicisign (2)	"A street cry"
Legisign (3)	Symbol (3)	Rheme (1)	A common noun like "apple"
Legisign (3)	Symbol (3)	Dicisign (2)	Propositions like "it's cold in here"
Legisign (3)	Symbol (3)	Argument (3)	"An argument"

2.1.3 Semiotics of the Audible Field

As Blauert states in *Communication Acoustics*, every sound can be regarded as a bearer of information and, therefore, be analysed as a sign carrier [4].

2.1.3.1 Sounds, Signs and Hearing: Towards a Semiotics of The Audible Field

This essay discusses how the process of hearing relates to sounds and meaning, with Capeller striving to draw a map of the audible field, rethinking the relationship between mimesis (the process of "imitation" and mimicry) and semiosis (the process that involves signs, including production of the meaning) [20].

As Capeller indicates, even though there are many other typologies of the audible field - based on hearing/listening processes and intrinsic features of sounds - the one developed by Michel Chion in *Audio-Vision: Sound on Screen* [21] has the advantage of avoiding the dependence on hearing and of the audible field to epistemic cognition[1] criteria, as these typologies tend to classify the audible field in terms of greater or lesser organisational complexity of sounds (considered by the authors) or of hearing, considered in its greater or lesser capacity as an emotive, energetic or intellectual *interpretant* of sounds.

Capeller illustrates that music is the privileged form of a "would-be linguistical paradigm" (or in Peircean terms, a legisign) for the arrangement of the audible field, pointing out that it is, however, a problematic solution when compared to other linguistic and semiotic paradigms, attributing this to the fact that music was the last of the classical arts to be fully codified and regulated by an ideal of artistic representation. Nonetheless, he affirms that the semiotics of the audible field can be reasonably articulated through an expanded concept of music, pointing out that the semiotician Lucia Santaella has further highlighted the correspondence between the three modes of hearing proposed by

[1]Term that refers to the understanding of the process of knowledge construction.

Michel Chion and the three Peircean phenomenological categories of experience. Lucia establishes a triad that, although not directly based on Peirce, presents a perfect correspondence with his categories. These are: **Reduced Hearing** (1), **Causal Hearing** (2) and **Semantical Hearing** (3). Capeller exemplifies that Reduced Hearing may bring to our minds some sounds and structures musically thought of by composers such as Xenakis (with many of his compositions being analysed as a succession of composed sounds, being called a "Sound sculptor" [22]). This mode can also pertain to situations where surrounding sounds cannot be located or identified or distinguish real or imaginary characteristics of sources. Causal hearing may refer to certain musical instruments or a singer's voice, but it also helps identify day-to-day noises or people's voices on the telephone. Semantical hearing may concern auditory abilities capable of understanding very complex musical structures. Still, it also refers to the central role played by organised verbal speech in all human communication. Capeller continues by expressing that **firstness** correlates to a special mode of hearing in which the listener's attention is reduced to following the free-floating amplitude and frequency variations of sounds (corresponding to its signifiers) in their random modulation and environmental propagation, therefore, reduced hearing. **Secondness** relates to causal hearing since sounds can act as traces or indexical marks of the supposed presence of real or imaginary sources or objects (visible or otherwise). **Thirdness** is correlated to semantical hearing, capable of activating the required *signifieds* for symbolically structured sounds, like a set of verbal phonemes[2] or musical notes and chords.

Still, Capeller states that the audible field presents a high rate of semiotic instability. As such, the correspondence mentioned above between Chion's three modes of hearing and Peirce's categories is not enough to lead to a comprehensive overview of all morphological aspects and complete mapping of the audible field. Thus, he states that the audible field must rely upon a theory of mimesis and that the principles of this theory can be found through Jacques Rancière's three regimes of distribution of the sensible. This solution is analogous to Pierce's triadic logic in the way that **Ethical Regime** is related to firstness, and the **Poetic-Mimetical regime** equates to secondness as it symbolises the representation of sounds. Lastly, **Aesthetical Regime** is intertwined with the Peircean concept of thirdness, meaning that the traditional poetic-mimetical codes of representation no longer guarantee the order of statements. Therefore, it needs continuous re-interpretation. By intertwining both Chion's three modes of hearing and Rancière's three ways of distributing the sensible, operated by Peirce's three categories, the mapping of the audible field can be deepened and amplified, thus obtaining a semiotical diagram of the audible field. Table 2.4 describes these relations and was adapted from Capeller's work [20].

This table can be read either vertically, through the three modes of hearing, or horizontally, through its three distinct audible regimes. **Sound objects** are considered pure

[2] A phoneme is a unit of sound that serves to distinguish one word from another in a language or dialect.

Table 2.4: Semiotics Of The Audible Field. Adapted from [20].

Audible Regimes / Modes Of Hearing	**Reduced Hearing** (1) Signifier	**Causal Hearing** (2) Object	**Semantical Hearing** (3) Signified
Ethical Regime of modulation and propagation of sounds (1)	**Sound Objects:** Variatons of frequency, phase and amplitude of sound waves	**Voice as object:** Vocalisation, phonation, intonation (phonoaudiology, psychoanalysis)	**Modal Music:** Noises, timbres, languages (Sociology, Anthropology, Etiquette)
Poetic-Mimetical Regime of codification and representation of sounds (2)	**Audio Signals:** Recording, editing and mixing sound tracks (radio, cinema and television)	**Voice as Chant:** Epics, lyrics, dramatics (recitals, poetics)	**Tonal Music:** Genre, styles, authors (rhythm, melody, harmony)
Aesthetical Regime of dissemination and interpretation of sounds (3)	**Soundtracks:** Atmospheres, art installations, sound machines (sound effects and sound design)	**Voice as Speech:** Diction, prosody and accent (linguistics, elocution)	**Discourse:** Enunciation, persuasion, interpretation (oratory, rhetorics and hermeneutics)

qualisigns since they result from a reduction of the audible field to its variations of wave amplitude, frequency and phase, as perceived by a human ear as volume, pitch and spatial localisation information. **Audio signals** are considered sinsigns and lead the audible field towards secondness since reproduced sounds are always experienced as imaginary doubles that indicate a correlated supposed real source or cause. **Soundtracks** can also be seen as sound design and tend to offer a wide range of articulated sound effects; these are presented as auxiliary legisigns that may be replicated to collaborate with signifying processes mainly conveyed through a visual or verbal form.

Capeller expresses that voices are the most important object for human hearing. Although causal hearing can be referred to as an infinite number of possible sound objects, its primary goal is to relate to a human voice through Peirce's iconic, indexical and symbolic categories, previously described in section 2.1.2.2. **Voice as object** is linked to ethical questions related to its iconic qualities, with its immediate objects referring to modulation, intonation and vocalisation, which are studied by phonoaudiology. In contrast, Dynamic objects are part of psychoanalysis and require persistent interpretation. **Voice as chant** is the most crucial sound object regarding the relationship between voice and language, as it is precisely in-between the mimetical expressive element of language and fully codified semiotics. There is no culture in which sound indexes do not strongly mark the vocal function. Its immediate object is the singing voice and its dynamic objects, including various methods of combining music and speech through verse. **Voice as speech** is the central object of causal hearing in its third symbolic layer. Its immediate object is language, and its dynamic objects are the non-discursive components of speech embedded in the sound materialisation of voices. This is a less semantic and more aesthetic mode of hearing a speaking voice, as it does not pay attention to the meaning of the discourse and its multiple possible interpretations. Instead, it focuses on elocution through its symbolic characteristics - diction, prosody and accent.

Capeller states that the audible field is also capable of generating autonomous codes of expression, which is what we call "music" while relating to Peirce's 3rd trichotomy of rhemes, dicisigns and arguments (previously described in section 2.1.2.3). **Modal music** is associated with Peircean's rheme, as it is still subordinated to a given set of verbal and visual expressions and has its signifieds (interpretants) outside the range of the audible field. **Tonal music** is at the boundary of the aesthetical regime, offering a representational image of a possible logical signified/interpretant of the audible field. It allows the complete development of pure musical ideas as sound organised propositions or dicisigns. Lastly, semantical hearing's most celebrated cognitive achievement is the ability to follow oral **discourse** and listen to a given set of arguments. This is the most abstract layer of the audible field, with oratory, rhetorics and hermeneutics being related to its emotive, energetic and intellectual signifieds (interpretants). The mimetic element of language often finds its way to verbal discourse through several sensuous and non-sensuous correspondences like puns, lapses, cacophonous sounds and undesirable rhymes.

2.1.3.2 Semiotics In the Context of Sound Design

Blauert states that when analysing the product sound from the perspective of the listener, three differentiations can be made [4]. The focal point is the listener's ambition to extract information via the perceived form characteristics of the product sound. The listener's cues for understanding can be related to the source, to the recipient and to the code, as summarised in the table 2.5.

Table 2.5: Listener's cues for understanding product sound. Taken from [4].

Source	Recipient	Code
Aesthetic cues	Imperative (re)-action	Index (causality)
Technical cues	Indicative (re)-action	Icon (similarity)
Functional cues	Suggestive (re)-action	Symbol (arbitrarity)

The author then asks if a sound refers to an object of experience where the listener has already processed comparable cues in the past. Sound cues are then separated into three categories of abstraction. This link to the universe of human experience matches the definition of the second trichotomy of the three Peircean categories of experience, described in Chapter 2.1.2.2.

According to Blauert, in **Index**, the sound refers the listener directly to its physical source. For example, let's consider the sound of a hammer banging on wooden material. If this event is an object of experience to the listener, wooden material will be associated with the sound. In **Icon**, the relation between the sound and its source is based on similarity, resembling the source by having some of its qualities. For example, an artificial sound contains reduced characteristics of natural sounds (preferably the most relevant ones) and avoids redundancy. Finally, in a **symbol**, the connection between a sound and its signified is based on conventions. Blauert gives the examples of a church bell indicating the hour and a warning signal in a car pointing the listener to not having fastened the seat belts.

2.2 Model-Driven Software Engineering

As the authors of the book *Model-Driven Software Engineering in Practice* [7] state: "Models are paramount for understanding and sharing knowledge about complex software". According to them, Model-Driven Software Engineering (MDSE) is a software development process that can be defined as a methodology for applying the advantages of modelling to software engineering activities. MDSE addresses software design with a modelling approach instead of a drawing one. These two approaches are distinguished because drawing is just about creating nice pictures, possibly conforming to some syntactical rules for describing design aspects. Modelling is a much more complex activity with implicit but unequivocally defined semantics, allowing precise information to be exchanged. Thus, compared to the drawing approach, modelling grants additional advantages like syntactical validation, model checking, model simulation, model transformations, model execution (either through code generation or model interpretation) and model debugging.

MDSE methodology seeks solutions through conceptualisation, which is responsible for defining conceptual models that describe reality, and implementation, which deals with mapping the models to existing or future running systems. The core concepts of MDSE are models and transformations (manipulation operations on models), meaning that there is some context to be modelled and some target for the models to be transformed into [7]. Both concepts need to be expressed in some notation, which in MDSE is called a modelling language (detailed in section 2.2.3).

Conceptualisation is applied at three levels [7]: application level, application domain level and the meta-level. The **application level** is where models of the applications are defined, transformations are performed upon the models, and actual running components are generated. In the **application domain level**, the modelling language, transformations and implementation platforms for a specific domain are defined. Lastly, in the **meta-level**, the conceptualisation of models and of transformations are defined [7].

Like conceptualisation, implementation also consists of three core levels: modelling, realisation, and automation. The **modelling level** is where the models are defined. In the **realisation level**, solutions are implemented through artefacts that are actually in use within the running systems (this consists of code in the case of software). Finally, the **automation level** is responsible for putting into place, the mappings from the modelling to the realisation levels.

2.2.1 Metamodels

The definition of a modelling language itself can be seen as a metamodel since they provide a way of describing the whole class of models that can be represented by that language. This procedure is referred to as **metamodelling**. In the same way that a model can be defined as an abstraction of phenomena in the real world, a metamodel can be defined as yet another abstraction, highlighting the properties of the model itself. At any level where metamodelling practice is considered, the model conforms to its metamodel in the way that a computer program conforms to the grammar of the programming language in which it is written [7].

2.2.2 The Model-Driven Acronyms

In the model-driven universe, there are a plethora of different acronyms. Figure 2.2 shows a visual overview of the relations between these. **Model-Driven Development (MDD)** is a development paradigm that uses models as the primary artefact of the development process. Usually, in MDD, the implementation is (semi)automatically generated from the models. **Model-Driven Architecture (MDA)** can be considered a subset of MDD, proposed by the Object Management Group (OMG), thus relying on the use of its standards, like modelling and transformation languages, to implement MDD. **Model-Driven Engineering (MDE)** can be examined as a superset of MDD since MDE goes beyond the pure development activities, encompassing other model-based tasks of a complete software engineering process (for example, the model-based evolution of the system or the model-driven reverse engineering of a legacy system). Finally, **Model-Based Engineering (MBE)** usually refers to a softer version of it, meaning that MBE is a process in which software models play an essential role but are not necessarily the key artefacts of the development (since they do not conduct the process as in MDE). An example of MBE would be a development process where designers specify the domain models of the system. Models still play an essential role in this process but are not the central part of the development process and may be less detailed than those in an MDD approach.

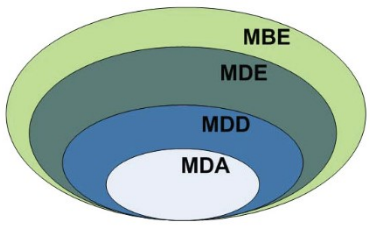


Figure 2.2: Relationship between the different Model-Driven Acronyms. Taken from [7]

2.2.3 Modelling Languages

Modelling languages are one of the main components of MDSE [7], and are the tool that allows for the definition of a concrete representation of a conceptual model, letting designers specify the models for their systems.

- **Domain-Specific Languages (DSLs)** are designed specifically to describe things in a determined domain. DSLs have been largely used in computer science even before creating the acronym. Some examples of this type of language include the established HTML (markup language for Web page development), MatLab for mathematics, and SQL for database access. If the language is aimed at modelling, it may also be referred to as Domain-Specific Modeling Language (DSML).

- **General-Purpose Modelling Languages (GPMLs)** - sometimes called **GMLs or GPLs** - alternatively represent tools that can be applied to any sector or domain, for modelling purposes. The typical example of this kind of language is the UML language suite (further detailed in 2.2.4) or state machines.

2.2.4 Unified Modelling Language (UML)

UML is a general-purpose, widely known and adopted modelling language. As such, it does not enforce any particular development method. It is a full-fledged language suite since it includes a set of different diagrams for describing a system from different perspectives. In figure 2.3, these different UML diagrams are categorised and classified. As stated by Brambila et al. [7], while seven different diagrams can be used to describe the system's static (structural) aspects, the remaining seven diagrams can be used for describing the dynamic aspects. Some of these diagrams describe the characteristics of classes, while others are used to describe the features and behaviour of individual items. The UML specification is maintained by the OMG [23].

UML provides a set of facilities for designing systems, enabling good design practices, such as Pattern-based design; using several integrated and orthogonal models together - meaning that diagrams share some symbols and allow cross-referencing between modelling artefacts; modelling at different levels of detail - allowing designers to choose the right quantity of information to include in diagrams depending on the purpose of the modelling and stage in the development process; extensibility - providing a set of extensibility features which allow the design of customised modelling languages if needed.

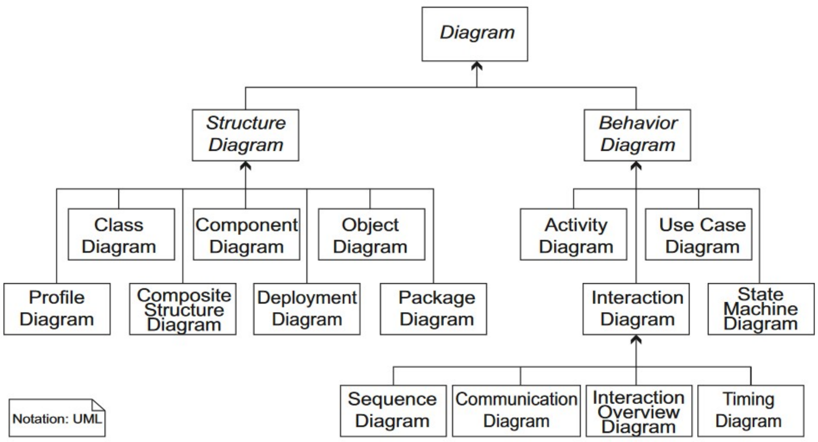


Figure 2.3: UML models taxonomy. Taken from [7]

2.3 Voice Computing

As expressed by Allen et al. [24], speech is the fundamental language representation present in all cultures. While text is often considered more durable than speech and more reliably preserved, text-based interaction with computers requires typing (and often reading) skills that many potential users do not possess for many reasons.

Schwoebel defines Voice Computing as a discipline that develops hardware or software to process voice inputs, stating that this field is booming, making it attractive to build and monetise voice-based applications [25]. Voice computing has become increasingly significant in modern times. As of 2017, 1 in 4 searches on Google are voice-enabled, and in 2021, Amazon Alexa passed 80 thousand skills[3]. Beyond these trends, Schwoebel claims that voice computing is having a potent effect on the global economy, driving innovation across a range of industries and fields, including machine learning infrastructure, natural language processing, audio engineering, data science, voice assistants, speech recognition, biometrics, text-to-speech software, connected home applications, cloud computing, interactive voice response systems, and many others [25].

2.3.1 Natural Language Processing

Natural Language Processing (NLP) is a subfield of linguistics and artificial intelligence that aims to build machines that understand and respond to text or spoken words in the same way a person does, complete with the speaker or writer's intent and sentiment. NLP combines computational linguistics (rule-based modelling of human language) with statistical, machine learning, and deep learning models. Some examples of NLP are: voice-operated GPS systems, digital assistants, and customer service chatbots [27].

The ultimate goal shared by many enthusiasts of Language Engineering is to write software using natural languages such as English. The problem with natural languages is that they are prone to misinterpretations, leading to fatal errors. In the case of aviation, some examples of miscommunication contributing to aviation accidents include the Tenerife accident in 1977 and the 1996 Charkhi Dadri mid-air collision, where 583 and 349 people died, respectively [28]. The International Civil Aviation Organization has acknowledged that "communications, or the lack thereof, has been shown by many accident investigations to play a significant role" [28]. Because of situations like this and safety concerns regarding the ability of pilots and air traffic controllers to communicate, "Aviation English" was devised. With around 300 terms, this language combines technical terms and plain English [29]. Allowing for quick communication ensures that important information is conveyed while limiting the possibility of misunderstandings. As an example of a domain-specific, restricted natural language, it was created to be precise, effective, unambiguous and serve a specific scope [29].

[3] *Skills are like apps that help you do more with Alexa.* - Taken from Amazon [26]

2.3.2 Voice Synthesis

Voice synthesis is the artificial production of human speech. A Text-to-speech (TTS) system converts normal language text into speech [24]. As Dutoit more concretely asserts, the ultimate goal of a TTS synthesiser is to read any text intelligibly and naturally through the production of new sentences. This differs from other voice response systems - such as the ones used in the announcement of arrivals in train stations - in the way that those are simply concatenating isolated words or parts of sentences, being applicable only when a limited vocabulary is required and when the sentences to be pronounced share a restricted structure. In the context of TTS synthesis, it is impossible to record and store all the words of the focus language, so it is more suitable to define it as "the production of speech by machines, by way of the automatic phonetisation of the sentences" [30]. The reverse process is called Speech Recognition, described in section 2.3.3.

In figure 2.4, a functional diagram of a general TTS synthesizer is presented. It is comprised of a natural language processing (NLP) module - capable of producing a phonetic transcription of the text to be read, together with the desired intonation and rhythm - and a Digital Signal Processing (DSP) module, which transforms the symbolic information it receives into natural-sounding speech [30]. The terms chosen for the intermediate information vary from one synthesiser to another. Here, as Dutoit explains, the term "narrow phonetic transcription" was chosen to emphasise that it is not composed of a mere sequence of phonemes but rather a list of allophone [4] names augmented with prosodic information [30].

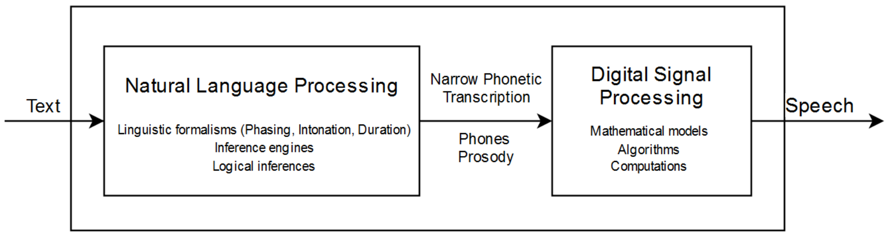


Figure 2.4: A functional diagram of a TTS system. Adapted from [30]

2.3.3 Speech Recognition

The task of speech recognition - also known as Speech-to-text (STT) - is to convert speech into a sequence of text by a computer program. Huang and Deng profess that the ultimate dream of speech recognition is to enable people to communicate more naturally and effectively [31]. Speech recognition focuses on the translation of speech from a verbal format to a text one, whereas voice recognition seeks to identify an individual user's voice [32]. As Huang and Deng explain, modern speech-recognition systems have been

[4] A phoneme that changes its sound based on how a word is spelt. For example, in English the aspirated t of *top* and the tt in the word *better* are allophones of the phoneme /t/

built invariably based on statistical principles. As figure 2.5 illustrates, the speaker's mind decides the source word sequence **W** that is delivered through the text generator. The source is then passed through a communication channel, consisting of the speaker's vocal apparatus that produces the speech waveform and the speech signal processing component of the speech recogniser. Finally, the speech decoder aims to decode the acoustic signal **X** into a word sequence $\hat{\mathbf{W}}$, ideally, as close to the original word sequence **W** [31].

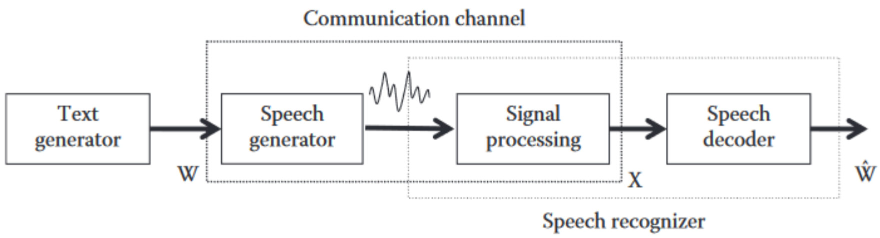


Figure 2.5: A typical speech-recognition system. Taken from [31]

As Huang et al. state: "Over the last four decades, there have been several breakthroughs in speech recognition technologies that have led to the solution of previously impossible tasks". The authors believe that in the next 40 years, speech recognition will pass the Turing test, expecting it to help bridge the gap between humans and machines. Nevertheless, they state that to reach this objective, some obstacles still need tackling [33]. Speech recognition is currently done on remote servers, with the results available within milliseconds on smartphones, making it difficult to adapt to the speaker and the environment dynamically. This has the potential to reduce the error rate by half [33]. Other problems include dealing with previously unknown words, recognising highly similar sounding words, and mixed-lingual speech - where phrases from two or more languages may be intertwined, often in countries where English is mixed with the native language. These problems have led to difficult user interface choices for which good enough solutions have been adopted [33].

Google's STT tool accurately converts speech into text using an Application Programming Interface (API) powered by Google's AI technologies, with STT having both REST and Google Remote Procedure Call (gRPC) methods for calling the API synchronous and asynchronous requests [34]. Through the **Synchronous Recognition**, audio data is sent to the API, where it performs recognition on said data and returns results after all audio has been processed. These requests are limited to audio data of 1 minute or less. Audio data is sent to the API through **Asynchronous Recognition**, and a *Long Running Operation* is initiated. Using this operation, the user can periodically poll for recognition results. Asynchronous requests are used for audio data of any duration up to 480 minutes. **Streaming Recognition** uses gRPC only and performs recognition on audio data provided within a gRPC bi-directional stream, meaning that the call is initiated by the client invoking the method and the server receiving the client metadata, method name, and deadline [35]. Streaming requests are designed for real-time recognition, such as capturing live audio from a microphone. This allows the results to appear while a user is still speaking [34].

2.3.3.1 Voice-User Interfaces (VUIs)

VUIs allow the users to interact with a system through voice or speech commands. Virtual assistants, such as Apple's Siri, Google Assistant and Amazon Alexa, are examples of VUIs [36]. Voice assistants recognise human speech and carry out commands pronounced by the users. They are powered by AI and base their performance on cloud storage with millions of words and phrases. Unlike the first voice recognition devices, modern assistants are not restricted by a specific language pattern or vocabulary. As mentioned in [36], applying the same design guidelines to VUIs as to graphical user interfaces is impossible. There are no visual elements in a VUI, and users do not have clear indications of what the interface can do or their options. Nowacki et al. [37] propose a set of guidelines tailored to VUIs, as a modified version of Bastien & Scapin's ergonomic criteria. These guidelines will be further analysed and discussed in Chapter 5.2.3.

3

Systematic Literature Review

In this chapter, a Systematic Literature Review (SLR) is presented. According to Kitchenham and Charters [38], an SLR is meant to identify, evaluate and interpret all available research relevant to a particular research question or topic area or phenomenon of interest. This methodology, as such, calls for a focused research question. It also includes a defined search strategy that delineates both the inclusion and exclusion criteria to limit the selection of papers solely to those that touch on the subjects relevant to this discussion. Afterwards, the primary studies concerning these topics of interest are collected, and their empirical evidence is summarised.

This chapter is structured as follows: establishment of the research question (addressed in Section 3.1); search strategy and selection of primary studies (Section 3.2), and data extraction and results (section 3.3), with each detailing the appropriate procedure.

3.1 Establishment Of The Research Question

The first step then is to establish a research question. As the primary goal of this work is to construct a foundational framework that can substantiate decisions when designing an auditory notation, while applying it in the context of UML diagram modelling, the following research question has been defined: *"Which guiding principles should we consider to use sound and voice as modelling notations, applied in the context of UML?"*. This specification will enable more accurate identification of primary studies that address this research question.

As proposed by the previously cited *Guidelines for performing Systematic Literature Reviews in Software Engineering* [38], the elaboration of this research question follows the PICOC criteria:

- Population: Modelling notations in general, and more concretely, UML diagrams.
- Intervention: Guiding principles for modelling notations.
- Comparison: Already established guiding principles for modelling notations.
- Outcome: Construction of an effective auditory notation, applied in the context of UML; Accessibility for visually impaired and blind users.

- Context: Research papers and experimental studies.

During the formalisation of the research question, other relevant topics arose. These questions will be able to help perform a quality assessment of the papers that should be selected further ahead. The questions are as follows:

- Q1. Aside from the principal focus on the auditory component, can it be used to heighten visual notation in general, and specifically, of UML modelling?
- Q2. What methods, principles and best practices have already been proposed to establish an auditory notation, applied in the context of UML?
- Q3. Are there preexisting tools/technologies already trying to implement said methods or principles? If so, what is the usability evaluation of said tools?
- Q4. Are the editors flexible from a configuration standpoint? Can new sounds and new notations be created, or is it all hardwired (a sound only has one specific usage, etc.)?
- Q5. What is the logical justification behind certain sound usages?
- Q6. Were there experimental studies, who were the participants and what were the results?

3.2 Search Strategy and Selection of Primary Studies

The next step is to conduct several searches in search engines such as IEEE Xplore, ACM Digital Library, Springer Link, Science Direct, and popular and comprehensive ones like Google to determine if there are existing papers relating to the subjects of this work. To limit the selection of papers solely to those that touch on the subjects relevant to this discussion, only studies that met both the inclusion and exclusion criteria outlined in Table 3.1 are included. It should be noted that while this document's principal objective is to find a practical and usable solution concerning the Computer Science field, there is also the intention of exploring and discussing a more profound and meaningful theoretical explanation for the usage of sound to propose guidelines towards a more concise understanding of the semiotics of the audible field. For these reasons, papers do not necessarily have to be Computer Science or Software Engineering related in the inclusion criteria.

Table 3.1: Inclusion & Exclusion Criteria.

Inclusion	**Exclusion**
1. Peer-Reviewed Studies	1.Non-peer reviewed studies [1]
2. Study must be written in English or Portuguese	2. Publications not in English or Portuguese
3. Study must be accessible electronically	3. Electronically non-accessible

[1]Abstracts, tutorials, editorials, slides, talks, tool demonstrations, posters, panels, keynotes, technical reports

The search strings were devised alongside the research question, with some terms deriving from it. In Table 3.2, the terms and expressions are presented. When these are connected through `AND` operators, they form the final search string. [39]

Table 3.2: Research query building.

Research Question	Which guiding principles should we consider to use sound and voice as modelling notations, applied in the context of UML?
Auditory Notation	(sound OR voice OR blind OR "model to speech")
Modelling Language	(language OR notation OR diagram OR UML OR domain-specific language OR model-driven OR modelling languages OR modelling software OR modelling workbench)
Editor	(editors OR edition tools OR IDEs OR "model editor")

Some included studies during this process were only obtained through a snowball procedure [40]. This methodology refers to the usage of a reference list of a given paper or to its citations, to identify and extract additional relevant studies, labelled as backward and forward snowballing, respectively [41]. A plausible justification for the fact that these articles were not found through the employed search strings could be that this is a knowledge domain where there still is not a consensus in the different nomenclatures and a more rigorous naming convention (e.g., some research refers to the process as *musification*, while others *sonification*).

3.3 Data Extraction and Results

Several primary studies concerning the various topics of interest related to establishing an effective UML auditory notation have been collected and a total of 18 studies were analysed, with only the most relevant being selected. Although some works lean further towards certain pivotal aspects and serve to extract better insights, the totality provides a basis for the genesis of this work. Since one of the main goals of this document is to create guiding principles to construct auditory notations, while also enhancing the user experience of working with UML through the usage of sound and voice commands, the research found was divided into three sections that concretely focus on each of these aspects, making reading easier. It should be noted that while a few of these articles have overlaps regarding the topics mentioned, a given paper was placed in one of these sections depending on its main subject of discussion.

3.3.1 Sound in a Software Context

The following section consolidates the findings of sound in the context of software, associating musical sounds to certain aspects of a program and how it can help with visualisation.

3.3.1.1 Sound As An Aid In Understanding Low-level Program Architecture

Some papers are concentrated on exploring non-speech sound, which can help establish principles, good practices and concrete analogies for certain sound usages. Such is the case with the research proposed by Berman [42], where a tool and associated sound mapping were developed to aid in the understanding of the static structure of Java programs. Berman proposes sound patterns that enable the developer to listen to specific sounds for a quicker inference of meaning, arguing that this could limit the number of visual context changes between windows inside the IDE. Some interesting ideas for the establishment of principles stem from the extracted real-life meaning and explanations used for certain sounds, such as the sound of hammering for a constructor. Other insights come from using altitude as an analogy to represent all of the elements, from the biggest to the smallest (element Class to the method itself). Furthermore, combining several sounds creates composite sounds that give the user more context. For example, Classes are represented by wind sounds and interfaces by bird sounds. We can indicate the specific interface that a given class implements by joining both.

3.3.1.2 Software Musification

By way of a snowball procedure, the research from Mancino and Scanniello [43] was analysed. A prototype was developed for an approach that the authors describe as *musification*. The general idea is to associate musical sounds with gathered software metrics resulting in a melody representing a given codebase. In addition to this description, this proposal differentiates itself from the precedently described research by Berman [42] in the way that it attends to software maintainability, reusability and complexity. It does not touch upon the topic of UML modelling. Nevertheless, it's related to the focus of this work since it seeks to utilise sound as a tool of utmost importance while complementing and enhancing software visualisation, offering some insights into how to devise guidelines for these types of efforts, such as the following paper.

3.3.1.3 Sonification Design Guidelines To Enhance Program Comprehension

This study by Hussein et al. [44], while not specific to UML, details a set of guidelines for the construction of tools that aim to join visualisation and *sonification*, arguing that these two presentation techniques should be combined to achieve maximum benefit for a large population of software developers. In [45], is detailed an approach to integrate sonic cues in the Eclipse IDE using MAX/MSP (a visual IDE designed specifically for audio and music-oriented applications. Through the usage of a plugin called *jitter* (a MAX/MSP package that provides scripts for processing network transactions), the data is then processed in MAX/MSP to render the corresponding sounds. A controlled experiment was conducted to assess the efficacy of sonification as a cognitive aid that assists program comprehension. Ten participants were involved, with nine being computer science undergraduate students from different universities and one graduate student. They answered questions about an unfamiliar codebase assisted first by visualisation and then by sonification of the program's information, or vice versa, with the order of showing being altered for each new participant. Results suggest that software sonification could be as effective as visualisation if used in the proper context and with the correct program information data. Fundamental guidelines in applying *sonification* include: increasing visual perception speed and accuracy; presenting multiple information pieces simultaneously; summarising information, and interchanging between visualisation and *sonification* to improve effectiveness and improve accessibility (revisited in Chapter 5.2).

3.3.1.4 There's More To Sound Than Meets the Ear: Sound in Interactive Environments

Kenwright [5] states that a sound is an under-appreciated tool in interactive environments that can synergistically connect multiple senses leading to a heightened state of engagement and focus. While this article is focused on virtual environments for serious games, some of its main ideas regarding sound design are universal and can be taken into consideration in the context of software programs. The key contributions of Kenwright's research [5], include the selective review and discussion on various audio design aspects in the context of interactive environments, identifying many important gaps and areas for future exploration that are often overlooked in sound and interactive environments and the detailing on cross-disciplinary sound discoveries which would have an impact on immersive sound design. Kenwright also states that sound, games, and interactive environments go hand-in-hand with learning and that "certain auditory patterns and sounds have an instinctual impact (soft, slow noises versus high-pitched shrieks). When these sounds are complemented with other material (auditory dialogue or feedback sounds), they offer an improved learning process (and experience)". Some examples of how sound design succeeds and fails are also listed and discussed (revisited in Chapter 5.2).

3.3.1.5 Towards a Conceptual Framework To Integrate Designerly And Scientific Sound Design Methods

This work by Hug and Misdariis [46] contributes to the integration of scientific and designerly methods for commercial sound design, in particular for interactive artefacts. A conceptual framework that aims to improve communication between sound and interaction designers, developed using both scientific and designerly methods is presented and evaluated with expert reviews. This framework integrates several relevant aspects of design to support the decision-making process, in particular in an early phase of development and during design evaluations. The authors state that sound can play an important role in the design of interactive commodities such as smartphones or wearables, given their small size, but also because sound is a powerful medium for conveying complex processes and information, even if they are in the background of our attention. This research is further discussed in Chapter 5.2.

3.3.2 UML Related Solutions For The Visually Impaired

This section compiles the existing research we have found on previously proposed UML solutions for visually impaired people.

3.3.2.1 UML Diagrams For Blind Programmers

Focusing on UML solutions, the experimentation developed by Coburn and Owen [47] started with the design of a tactile and physical solution to help a blind student finish a course, but soon matured into an auditory system called *Audible Browser*. While still a somewhat primitive tool, it only reads class and state diagrams explicitly and doesn't allow for their editing - presents them using brief, distinct, non-verbal sounds combined with speech. The current version uses simple tones for each node in the diagram. The usage of the pitch as the placement in the diagram gets higher (vertical placement) and stereo placement to convey horizontal placement can also be used to heighten the visual notation. This paper will be discussed more extensively in Chapter 4.2.

3.3.2.2 Teaching UML Database Modelling To Visually Impaired Students

This paper from Brookshire proposes a tactile solution [48]. Therefore, it is not entirely adequate for the intents and purposes of this work. However, this research should be briefly mentioned as it proposes ideas to create a sort of tactile syntax for UML Class Diagrams. For this research, Brookshire started by focusing on the critical elements of class diagrams; the rectangles indicate the classes, lines indicate relationships and symbols designate the cardinality of the relationships between the classes. The attributes of the classes correspond to the fields in the database tables. They are composed of text and can be entered separately in a word processor or text editor. Brookshire's implementation used cards to represent the rectangles for the classes and thin plastic strips instead of strings to represent the lines connecting the classes, stating that these are easier to manipulate. In contrast, strings would require some physical anchoring system on each end to remain straight. Flat thumbtacks were used to fasten the strips to the mounting surface where the diagram was mounted, and pushpins represented cardinality. No pin represented zero, one pin represented one, and two represented many. For example, a one-to-many relationship had one pin on one side of the strip and two pins on the other. A visually impaired student tested this method, and the quality of the diagram they created was similar to the quality of the ones made by the sighted students [48]. Seeing that the main focus of this work is to create an auditory notation for UML, it may be possible to learn from the reasoning and analogies used by this author to bridge the gap between his solution and the visual class diagrams' syntax. This research also inspires us to the possible use of different sounds that allude to distinct tactile sensations for various elements of the diagrams, simulating the feeling of touch through sound, particularly considering that these two senses are even more intertwined for visually impaired people.

3.3.2.3 On The Inclusion Of Blind People in UML e-Learning Activities

Luque et al. [49] present the main challenges for the inclusion of blind participants in interactive e-learning activities involving UML, stating that the difficulty concerning the use of tools to support these activities arises mainly from the lack of physical interaction between participants, considering their spatial distance, as well as the limited availability of haptic devices. Despite having a greater focus on the collaboration between sighted and visually impaired individuals, this paper analyses tools that support accessible representations of UML diagrams and evaluate them regarding the fulfilment of the proposed requirements. According to the authors, the results indicate the lack of solutions that enable the inclusion of blind people without being affected by 'transactional distance', defining this as the "psychological and communication space to be crossed during interactions between instructors and learners, which increases by the occurrence of spatial or temporal distance". To contribute with technology capable of fulfilling the requirements proposed in this paper and fill the existing gaps by other researchers working in this field, the authors present some details on the prototype development of a web-based tool called Inclusive UML, which was developed and would be made available to the public as free software. As of this research, no releases have been made [49].

3.3.3 Speech Recognition - Software Modelling Using Voice Commands

In this section are gathered the studies associated with Speech Recognition, to enable the creation and substantiation of a syntax that makes sense for UML diagrams whilst learning from the obstacles faced by these authors.

3.3.3.1 Using Voice Commands for UML Modelling Support on Interactive Whiteboards: Insights and Experiences

The main goal of Jolak et al. [50] was to find out which features of software design environments are prioritised to be supported by voice interaction, as well as understand whether the support of these features could enhance the usability and efficiency of software design environments and be of benefit for software design processes. To achieve this, they designed and evaluated a software design environment called OctoUML, which supports touch, mouse, keyboard and voice. Two user studies were also conducted with a population sample of thirty subjects to evaluate and compare the usability and efficiency of two versions of OctoUML. One version has speech recognition (OctoUML-V), and the other does not (OctoUML-Lite). The authors concluded that "the general perception regarding the usability is positive", with the voice interaction modality being most preferred for text input, stating that it is "easier, faster and more comfortable to use voice instead of typing" [50]. The subjects preferred OctoUML-V over the other version, as it "enhanced the efficiency of the software design process by reducing the required time for naming the elements of the software design diagram" [50]. Utilising voice makes sense when drawing UML diagrams. Using this alternative instead of typing (for example, in naming classes and packages in Class Diagrams) is less time-consuming and justified for people with disabilities. OctoUML currently only supports class diagrams.

3.3.3.2 A Voice-activated Syntax-directed Editor For Manually Disabled Programmers

This paper by Hubbell et al. [10] approaches the problem not from the perspective of blind people but through manually disabled people, focusing on designing and implementing a Speech User Interface that allows manually disabled people to more easily perform the task of programming as an input alternative to the keyboard and mouse. As such, it is not specific to UML or aimed at creating sound cues or an auditory notation, but could help formulate guidelines and patterns to apply voice commands when designing a syntax-directed editor for UML. The tool developed for this effect is VASDE (Voice-Activated Syntax-Directed Editor), whose purpose is to be efficient, effective and intuitive to use after minimal initial training. It is explicitly bound to the java language. Java Speech API (JSAPI) provides the interface between VASDE and the underlying speech recognition engine by providing a means for recognising speech and triggering required actions. Lastly, the IBM ViaVoice engine was selected for use with VASDE. Regarding its experimental study, the participants were a small group chosen from the ranks of university students and professional programmers, including one evaluator with severe manual disability. The evaluation process began with training the speech recognition engine for each subject. The evaluator was then provided with a tutorial that introduced the major functions of VASDE. After this tutorial was completed, each evaluator was given a sequence of individual tasks that involved transcribing and editing a Java program in the VASDE application. Afterwards, each evaluator responded to a questionnaire that was used to provide feedback on VASDE. The authors state that from evaluation comments and observations, the most glaring source of frustration was the lack of recognition accuracy by the speech recognition engine, despite approximately one hour of voice training. This resulted in proposing three ways in which the accuracy might be increased. The first and second approaches involve the training of the engine, while the third suggestion was to alter the actual speech commands themselves. Longer commands provide more context for the speech recognition engine and greater accuracy. However, longer commands are likely to be harder to be remembered by users [10].

3.3.3.3 EULER - Mathematical Editing By Voice Input For People With Visual Impairment

As with UML, the visual nature of mathematics makes it challenging for those with vision impairments to learn and develop their skills. The authors of this paper [9], while focusing on people with visual impairment and voice input, directed their efforts at mathematical editing. Rivas-Pérez et al. state that "representation of mathematical expressions through voice recognition is an alternative for the development of accessible mathematics". This under-development tool called EULER, allows ease of use, offering visually impaired users the option of creating verbally constructed mathematical expressions, emphasising a machine learning-based solution that uses a learning algorithm for more accurate input of mathematical symbols and expressions. For this, they used TensorFlow, an open-source library for machine learning and artificial intelligence developed by Google, and the voices of 145 volunteers for training. Experimental studies were also conducted for the evaluation of voice input, interpretation and representation. Three people participated, two of them with low vision and one with blindness [9]. The results indicated that the interpretation/parsing of specific numbers was not always correct and that it is necessary to improve the accuracy. In the future, the authors plan to integrate more composite symbols that correspond to advanced levels in secondary and university education, such as systems of equations, integrals and propositional logic, and include support for multiple languages, since only Spanish is currently supported. The authors also deem it necessary to expand the number of participants, especially those with blindness or low vision. Certain aspects of this research could prove helpful in learning the shortcomings of their process to develop a more effective voice input for UML.

3.3.3.4 Voice-Driven Modeling: Software Modeling Using Automated Speech Recognition

This research [51] proposes a Voice-Driven Modelling (VDM) approach to Model-Driven Engineering. This still in development VDM framework consists of 3 phases: Speech Processing, Natural Language Processing and Context-Specific Modelling. The authors support its suitability through the analogy of a traditional compiler, stating that "the context-specific modelling phase is akin to the target code generation phase, in which the intermediate representations produce output in a target language". It is worth mentioning that there are other tools for Voice-Driven Software Engineering such as Vocola, SPEED and VoiceCode, with the authors drawing inspiration from all the designs and benefits of these techniques. The authors state that their immediate goal is to create an end-to-end solution for Simulink modelling through verbal input in English, outlining their current early stages of research and plans. Although they are working with Simulink as a baseline, they intend to create an approach that is ultimately portable to a variety of tools, affirming that if VDM proves to be an efficient and effective method of modelling, there is potential for integration into existing modelling tools such as the Eclipse Modelling Framework, rather than functioning as a standalone tool sitting on top of other technologies. Black, Rapos and Stephan believe that VDM and its associated research have the potential to benefit not only modellers with disabilities and non-modellers who are domain experts but also software modellers in general, further claiming that work on VDM has the potential to influence the design and evolution of future modelling language design to better support voice-driven approaches.

3.3.3.5 Improving The Usability of Voice User Interfaces: A New Set Of Ergonomic Criteria

The research by Nowacki et al. [37] proposes a set of usability criteria tailored to VUIs that incorporate the advice from 26 design guidelines from both academic and professional sources while comparing them to Scapin & Bastien's criteria. This resulted in a list of 8 criteria and 18 sub-criteria. According to the authors, this work shows that one of the main challenges that designers of VUIs will face consists in the interpretation of the interface as a fictitious character, accessed through a new medium and with more flexibility. Users might feel uneasy about how to utilise this new medium until new expectations and standards emerge as VUIs become more common. These guidelines are revisited in Chapter 5.2.

3.4 Final Observations

Considering that works that completely intersect with the goals of this document were not found, it was necessary to broaden the scope of our research. a total of 18 studies were analysed, with only the most relevant being selected. Even though the found research was divided into three sections, all of these were investigated with the goal of answering the research question, along with the questions that helped perform a quality assessment of the selected papers. While a few of these articles have overlaps regarding the mentioned topics, a specific study was placed in one of these sections depending on its main subject of discussion.

Some of the studies found are slightly more focused on aspects other than the usage of sound in UML, but they all offer some insight and interesting ideas that can be applied to this work. For instance, multiple studies on tactile systems were uncovered, and despite not being our focus, some ideas overlap with the usage of sound, as is the case with the research by Brookshire [48]. Several articles also provided design guidelines and frameworks for a more scientific sound design method. These are presented in Chapter 5.2. Summarising these guidelines facilitates their examination and further discussion while supporting some of the decisions made during the implementation phase.

4

Related Work

This chapter will focus on more deeply analysing the most relevant works that have addressed, in some form, the topics of this document. The insights provided here will serve as one of the pillars upon which Chapter 5 and subsequent work will be based. In section 4.1, the various principles toward a scientific basis for the construction of visual notations in software engineering that author Daniel Moody proposes [8] are presented. In section 4.2, other works and their concepts are discussed.

4.1 The "Physics" of Visual Notations

The work presented in [8] focuses on addressing and raising awareness about the importance of specific visual representation issues that, historically, software engineering researchers and notation designers have ignored or undervalued. One possible explanation for this situation, offered by Moody, is that researchers see visual notations as being informal and that therefore, serious analysis can only occur at the level of their semantics. However, the author states that this is a misconception, as visual languages are no less formal than textual ones, believing that visual representation decisions profoundly affect the usability and effectiveness of SE notations, equal to (if not greater than) the decisions about semantics. Moody first explains how communication consists of two complementary processes: encoding and decoding. In visual notation, a diagram creator (sender) encodes the information (message) in the form of a diagram (signal), and the diagram user (receiver) decodes this signal. The diagram is encoded using a visual notation (code), and the medium (channel) is the physical form in which the diagram is presented (e.g., paper, whiteboard, computer screen). Noise represents random variation in the signal which can interfere with communication [8]. This work establishes the foundations for the science of visual notation design, by defining a visual theory for how visual notations communicate and, based on this, a set of principles for designing cognitively effective visual notations.

The first principle is **Semiotic Clarity**, which states that there must be a one-to-one correspondence between symbols and their referent concepts. Natural languages are not notational systems as they contain synonyms and homonyms, but many artificial languages such as musical notation are. One or more anomalies can occur when there is not a one-to-one correspondence between constructs and symbols. These are: Symbol redundancy, which happens when multiple graphical symbols can be used to represent the same semantic construct; Symbol overload, which occurs when the same graphical symbol can represent two different constructs; Symbol excess, when graphical symbols do not correspond to any semantic construct and symbol deficit, which occurs when there are semantic constructs that are not represented by any graphical symbol.

The second principle is **Perceptual Discriminability**, where different symbols should be distinguishable from one another. Accurate discrimination between symbols is a prerequisite for an accurate interpretation of diagrams.

Visual representations whose appearance suggests their meaning concerns the third principle, called **Semantic Transparency**. While Perceptual Discriminiability requires that symbols are different from each other, this principle requires that they provide cues to their meaning. As Moody establishes, this concept formalises informal notions of "intuitiveness" that are often used when discussing visual notations. Semantic Transparency is not a binary state but a spectrum. On one side of the spectrum, the symbol is semantically immediate if a novice reader could infer its meaning from its appearance alone (e.g., a stick figure to represent a person). On the other side of the spectrum, the symbol is semantically perverse if a novice reader is likely to infer a different (or even opposite) meaning from its appearance. A symbol with a purely arbitrary relationship between its appearance and meaning is defined as semantically opaque. There are varying degrees of semantic translucency between semantic immediacy and opacity, where symbols provide a suggestion to their meaning but require some initial explanation. Icons are symbols that perceptually resemble the concepts they represent, reflecting a basic distinction in semiotics between **symbolic** and **iconic** signs.

The fourth principle is **Complexity Management**, which calls for the inclusion of explicit mechanisms for dealing with "diagrammatic complexity", measured by the number of elements on a diagram. This complexity significantly affects cognitive effectiveness as the amount of information a single diagram can effectively convey is limited by human perceptual and cognitive abilities. There must be mechanisms for modularisation and hierarchical structuring to adequately represent complex situations.

The fifth principle, called **Cognitive Integration**, states that explicit mechanisms should be included to support information integration from different diagrams. For these representations with multiple diagrams to be cognitively effective, there must be support for Conceptual Integration and Perceptual Integration. Conceptual integration is a summary diagram which provides a view of the system as a whole. By using a technique called contextualisation in the context of diagrams, contextual information on each diagram can be included, showing its relationships to elements on other diagrams. For this, Moody suggests including all directly related elements from other diagrams (its "immediate neighbourhood") as foreign elements. Perceptual integration is about navigating a set of diagrams the same way one would navigate around a city or a website. It follows the four stages of wayfinding: orientation, route choice, route monitoring and destination recognition. Moody suggests clear labelling of diagrams to support orientation and destination recognition, level numbering to support orientation by showing the user where they are in the system, navigational cues on diagrams to support route choice and a navigational map showing all diagrams and the navigation paths between them.

The sixth principle is **Visual Expressiveness**, which is about using visual variables' full range and capacities. These visual variables should be chosen based on the nature of the information to be conveyed. Different variables have properties that make them suitable for encoding different types of information. These variables are horizontal and vertical position (x and y), size, brightness, colour, texture, shape and orientation.

The seventh principle is **Dual Coding**, which is described as using text to complement graphics. Using text and graphics together to convey information is more effective than using either on their own, with textual encoding being the most effective when used in a supporting role, aiding interpretation by providing textual cues to the meaning of symbols when they are not semantically transparent.

Graphic Economy is the eighth principle and refers to the number of different graphical symbols that should be cognitively manageable. There are three main strategies for dealing with excessive graphic complexity, which are: reducing semantic complexity by way of simplifying the semantics of a notation; introducing symbol deficit by finding the right balance between graphical, textual and off-diagram encoding that maximises computational offloading; and finally, increasing visual expressiveness. By using multiple visual variables to differentiate between symbols, the ability to distinguish between many categories is expanded.

Finally, the ninth and last principle is **Cognitive Fit**, which states that different dialects for different tasks and audiences should be used, with the differences between experts and novices suggesting the need for at least two different visual dialects.

These principles can conflict with one another, so they can be used to make certain trade-offs. When these principles support each other, they can be used to exploit these synergies. Figure I.1 (located in the Annex I) describes these relations, where + indicates a positive effect, - indicates a negative effect and ± indicates a positive or negative effect depending on the situation [8].

4.2 Other Relevant Works

Additional works that are relevant to the topics of discussion are examined in the following sections.

4.2.1 Audible Browser - UML Diagrams For Blind Programmers

In Chapter 3.3.2, the research by Coburn and Owen [47] was briefly mentioned. As previously stated, this paper proposes an Audible Browser for presenting visual UML diagrams to a blind user in an audible manner. It automatically interprets the diagrams, avoiding the need for an intermediate user. The authors state that the Audible Browser reads XMI 2.1 files, specifically class and state diagrams. The tool interprets and presents the nodes of a diagram in a tab group, a set of common items that can be listened to by using the up and down arrow keys. The tab groups for a given class are the name, attributes, operations, and associations.

As the authors explain, the **name tab** group initially speaks the class name and any inheritance or nesting, with the arrow keys allowing the user to traverse through a list that includes any other class this class is nested in, classes nested in this class, classes this class inherits and classes that inherit this class, allowing the user to navigate to the associated class by pressing the return key. The attributes in the **class attributes** tab are narrated in a simillar manner to how someone might read them aloud in a typical UML class diagram. For example, the attribute *x: int=0* is spoken as: "x colon int initially zero". The arrow keys allow the user to hear the letters one by one, preventing ambiguity if the speech generation is unclear. The **operations tab** allows users to examine the operations of the class. The function *func(x:int, y:int) : string* is read as "func parameters x colon int y colon int returns string". As before, the arrow keys allow the user to listen to each letter separately. Finally, the **associations tab group** lists associations, aggregations and compositions. In each case, the roles and multiplicities are read with the association. For example: "association diagram 1 to state 0 dot dot asterisk"

Coburn and Owen stress that the physical structure of the diagrams is important, as one can infer relationships due to groupings and proximity. Audible Browser presents a diagram using earcons in combination with the speech mentioned above, to convey the structure of a diagram. The tool uses simple tones for each node in the diagram, which the authors refer to as stars. A collection of these stars is a constellation, and a UML diagram is conveyed as a sequential presentation of stars. The left-right stereo placement of stars represents the X position in the diagram, and the Y position is represented by pitch, assigned to the nearest musical semitone. Nodes on the page are presented with a higher pitch, while the node's name can also be optionally spoken when the constellation is presented.

Although this tool does not allow for the editing of UML diagrams, and associations and inheritance situations still have problems that need to be addressed, it provides a basis for certain interesting ideas, such as the usage of pitch and stereo panning to convey vertical and horizontal placement, respectively, using speech to add an extra layer of information, while allowing the users to interact and navigate by using the arrow keys.

4.2.2 ModelByVoice Tool

The *ModelByVoice* tool is an editor that allows the automatic generation of a model editor from an annotated metamodel with the respective syntactic sound elements. The generated editor allows modelling activities to be carried out exclusively by using sound and voice, enabling better analysis and design of sound diagrams. Figure 4.1 illustrates the tool's architecture with its technological specification. It is divided into three levels and assumes two users: the language engineer and the user who intends to perform the modelling activity. The first and second levels are the language engineer's responsibility, and the third level belongs to the user [11].

The first level contains the editor that allows the language engineer to define the modelling language and the concrete sound language syntax. This concrete sound syntax is defined through the association of sound to elements of the language and settings configuration. By specifying both the concrete and abstract syntax, generating the Main class in the platform *ModelByVoice II* is possible.

The second level aims to integrate the generated class in the *ModelByVoice II* platform, in addition to verifying the existence of sounds whose *path* was not nominally specified, that is, sounds that were not the result of the definition of characteristics.

Finally, the third level corresponds to the final product and allows the end user to perform modelling activities through sound and voice.

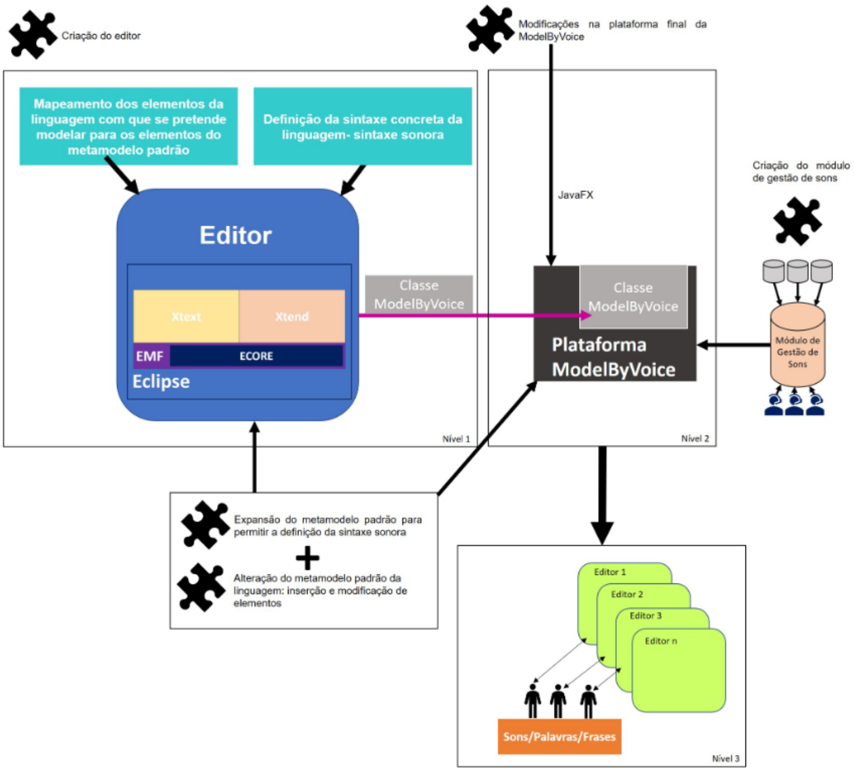


Figure 4.1: ModelByVoice II Architecture Overview. Taken from [11]

5

Principles For The Construction Of Auditory Notations & Useful Guidelines

This chapter contains the elaboration of the proposals that will serve as the foundation for the subsequent work, gathered and formulated through the insights explored and discussed in the previous chapters of this document. Section 5.1 presents an adaptation and subsequent discussion of the principles proposed by Moody (discussed in section 4.1), for the construction of an auditory notation. Section 5.2 gathers and discusses design guidelines and frameworks found during this essay's research to facilitate the examination by having these all in one place while also corroborating some decisions during the chapters to follow.

5.1 Revisiting The "Physics" Of Notations: Principles For An Auditory Notation

In this section, we transpose the principles proposed by Moody (described in Chapter 4.1) in a manner analogous to sound. Table 5.1 was delineated, featuring for each of these principles, the definition stated by Moody, and a proposal for how it can be applied to sound. As the "translation" of each concept is not always straightforward, the reasoning that led to each correspondence will be discussed later.

In **Semiotic Clarity**, each sound should represent only one concept. This means that each proposed sound should be reserved for only one concept.

Different sounds should be clearly distinguishable for the principle of **Perceptual Discriminability**. In the visual field, shapes play a special role in discriminating between symbols, representing the primary basis on which objects are identified in the real world. Thinking about the characteristics that constitute a sound, the timbre (or "tone colour") can be considered as its "shape", as it is a form of distinguishing different objects. Sounds should have distinct and unmistakable timbres to be recognisable. For example, a sound of an arrow being shot can be used for an association in a UML Class diagram, while a baby crying can be used to describe a dependency.

Table 5.1: The 9 principles proposed by Moody with our auditory equivalent proposal.

Principle	Definition	Auditory Proposal
Semiotic Clarity	One-to-one correspondence between symbols and their referent concepts.	Each sound should represent only one concept. meaning that each of the proposed sounds should only be played for only one concept.
Perceptual Discriminability	Different symbols should be clearly distinguishable from one another.	Different sounds should be clearly distinguishable from one another. For example, using different, unmistakable timbres for distinct sounds.
Semantic Transparency	Visual representations whose appearance suggests their meaning.	Auditory representations should suggest their meaning. For example, the sound of a baby crying to symbolise a dependency.
Complexity Management	Include explicit mechanisms to deal with the complexity of a diagram.	Noise of a container/box/door to signify "composition", supporting the concept of hierarchy. Usage of reverberation could be explored, a sound with more echo should signify a further away diagram.
Cognitive Integration	Include explicit mechanisms to support integration of information from different diagrams	Using auditory motifs that refer to a specific diagram as a way of contextualising it. Also using TTS to orientate the user.
Visual (Auditory) Expressiveness	Use full range and capacities of visual variables such as size, brightness, colour, texture, shape and orientation.	In this case, Auditory Expressiveness. Use the full range and capacities of auditory variables such as timbre, pitch, loudness, duration, reverberation, etc.
Dual Coding	Use text to complement graphics.	Certain sounds can be accompanied and complemented by a Text-to-speech voice.
Graphic Economy	Number of different graphical symbols should be cognitively manageable.	Finding the right balance between distinguishable sounds while being cognitively manageable.
Cognitive Fit	Use different visual dialects for different tasks and audiences.	Creating diagrams through the usage of voice can have both simplified commands and more complex ones for experienced users. Certain auditory cues conveying additional information can be omitted for less experienced users.

In **Semantic Transparency**, the auditory representations should suggest their meaning. The successful exploration of this principle at the auditory level could potentially be more complex compared to its visual counterpart, namely when finding outstandingly intuitive sounds for "abstract" concepts. This could result in semantically opaque sounds, with their meaning being purely arbitrary. Ideally, the sounds should be semantically immediate. Still, if they are located somewhere in the area of semantic translucency (as mentioned above, the example of the baby crying to symbolise dependency), it would already be a great starting point for this research.

For the **Complexity Management** principle, explicit mechanisms that deal with the complexity of a diagram should be included. For example, the noise of a container/box/-door can be used to signify "composition" and support the concept of a hierarchy. Reverberation could also be explored: a sound with more echo denoting a diagram that is further away.

Cognitive integration must include explicit mechanisms to support the integration of information from different diagrams. By using sound motifs to summarise the concepts of a diagram, conceptual integration can be included. Combining sound motifs and text-to-speech voice synthesis can help contextualise directly related elements from other diagrams as foreign elements. Perceptual integration could be established by using text-to-speech, allowing for clear diagram labelling and easier wayfinding on the users' behalf.

The principle of **Visual Expressiveness** can be thought of as "Auditory Expressiveness", where the full range and capacities of auditory variables are used. Krygier defines these variables as location, loudness, pitch, register, timbre, duration, rate of change, order and attack/decay [52], each one is explained in figure I.2 (located in the Annex I). Besides these variables, reverberation can be useful, as discussed in the principle of Complexity Management or even panning the sound to the left or right channel.

Dual Coding can be directly translated to an auditory approach. As text can be used to complement graphics in visual notation, certain sounds can be accompanied by a text-to-speech voice, reinforcing meaning as an additional cue to a given concept.

The principle of **Graphic Economy** becomes "Auditory Economy". Just as graphical symbols should be cognitively manageable, so should the auditory ones, as these might be harder to assimilate in a first approach. One solution to this problem might be to increase auditory expressiveness by using multiple auditory variables (described in the principle of visual expressiveness) to differentiate between symbols. The right balance between distinguishable sounds while being cognitively manageable should be found.

Cognitive Fit suggests the usage of different visual dialects when drawing and communicating certain aspects of a diagram. For an auditory approach, creating diagrams through voice can support simplified and more complex commands for novice and experienced users, respectively. While interpreting a diagram, certain auditory cues conveying additional information can also be omitted for less experienced users.

5.2 Design Guidelines

This section will detail and discuss already established guidelines and frameworks by published works. The goal is to facilitate their examination by documenting them all in one locale, corroborating some future decisions, and assisting with streamlining certain aspects of this document's proposed work.

5.2.1 Guidelines For Sound Design In The Context Of Software

Blauert affirms that product-sound design became an issue during its infancy, with sound being treated mostly as secondary, being sometimes compared to unwanted noise. Sound was considered an issue of minor relevance to a product. Nevertheless, Blauert explains: "A product sound carries information about the product. With sound contributing to a positive meaning, users get attracted, access to the product is simplified, and its use is supported. That, in turn, improves the perceived product quality." - concluding that sound is a crucial component of product quality [4]. One of the tasks of sound design is to understand why certain sounds can be regarded as positive foreground events or "identity cards" of a product and, consequently, to acquire the basic knowledge to design such sounds on purpose systematically. However, Blauert also affirms that although there are often several sound designers involved in product planning and specification, sound design as a task seems to lean further towards the artistic side where intuition is called for, rather than being a task that can be approached in a rational and systematic manner. As a result, Blauert proposes various criteria for product-sound design:

- Motivation for product use - What are the reasons for product use? What does the listener gain when using the product? The product may compensate for deficiencies as a hearing aid does or make work much easier.
- Function of the acoustic component in the product context - Is the acoustic signal a key quality element or just a secondary phenomenon in the background?
- Function of the acoustic component in relation to other product components - Are the sounds adequate considering other modalities, adverse conditions, etc.?
- Meaning associated with product use - e.g., applicability, suitability for use, saturation of demands, meeting of requirements, product performance, functionality, security.
- Dominant quality items of the product's sounds - How is the product's sound judged? e.g. innovative, simple, complex, practical, distinctive, convincing or typical.
- Provoked reactive behaviour of the user as to product quality - For example, the product is reliable, safe, high performance, creates emotional reactions such as disgust, contempt, resentment, dissatisfaction/satisfaction, empathy, etc., or motor activity like the ease of handling, ergonomy and flexibility.
- User specification - Can a typical user be specified? If so, how can the group of users be typified, for example, according to age, education, experience of techniques and expectations of quality?

Blauert asserts that sound should be supportive of the product and that the product itself should answer a question and meet demand, adding that sound design itself did not have quite the same expansion and attention compared to other design areas like visual design. As a consequence, the future of product-sound design is open to more radical design solutions [4].

As referenced in Chapter 3.3.1, Hussein et al. [44] propose a set of guidelines aiming to address the challenges that pertain to combining visualisation and sonification in a single comprehension tool. They are as follows:

- Add sonification to simplify visualisations - For example, a Call Hierarchy View in Eclipse. Eclipse generates a tree visualisation of a call graph upon this request on a given method. However, this provides no information about the depth of the generated call graph. Adding a special visual cue that can show the depth of the tree is likely to clutter the tree visualisation. By contrast, adding a sonic cue representing the tree's depth via a different volume or pitch is relatively straightforward, minimising the required interaction time.
- Increase visual perception speed and accuracy by adding sonification - Auditory cues can enhance visual perception. E.g., multiple concurrent edits of the same

source file complicate the subsequent merging of the changes. The authors argue that a sonic cue representing the number of concurrent edits can be rendered. This could effectively supplement the information already conveyed by a visualisation.

- Add sonification to present multiple information pieces simultaneously - The use of sound can improve comprehension and lower cognitive load when a person has to monitor various information sources updated concurrently. Musical scores convey lots of concurrent information together with their mutual relationships. For example, the metric of the number of lines of code in a method used in this research experiment could have used sonification based on multiple attack beats compressed in a short time interval.
- Use sonification to summarise information - A simple sonification could effectively complement a visualisation by providing a summary for large volumes of information. The authors give the example of using a sonic cue to express a relative length or the cyclomatic complexity metric of a selected source file.
- Interchange visualisation and sonification to improve effectiveness - Although audio may not necessarily convey more information, it covers more ground spatially and, as such, could provide more distinguishable entry points for monitoring. The authors state that a study has observed that sound superseded visuals in terms of the conveyed detail, while at other times, the situation was the opposite. This insight suggests that aural cues could be more effective than their visual counterparts when utilised in the same scenario under certain circumstances.
- Alternate visualisation and sonification to improve accessibility - The importance of accommodating users with disabilities has been widely recognised. Auditory representations of the information could supplement the otherwise inaccessible cues for impaired users.

Kenwright [5] states that audio is often an afterthought in many interactive environment projects, stressing that sound should be designed - like all aspects - intentionally or not. The author gives examples of how sound design fails in the context of serious games. The main ones listed below are adapted into a broader scope that can encapsulate the work being developed in this document.

- Positional audio appears nonexistent - Difficulty to tell where voices are coming from
- Volume mixing is inconsistent and random - Actions have a soft sound, and voice audio is too low
- Actions sound weak in comparison to everything else - Some sounds are barely audible, while subsequent sounds may be too loud
- Actions with no corresponding audio - Something happens, but there is no audio indication
- Sound which contradicts one another in tone and style

In the context of this work, the examples listed above could be extrapolated to having positional audio representing the different positions where elements in a diagram are located. Since these elements can be located more to the left, right, higher or lower, the sound representing these elements should translate this positioning into a 2D plane. In addition, there is still the fact that if there is a text-to-speech reading of the various elements that make up the diagram, the sound of the voice must have a balanced volume, along with the sounds that represent the sound notation.

5.2.2 Scientific Method For Sound Design - A Conceptual Framework

Hug and Misdariis [46] developed this framework of concepts and heuristics to help inform design decisions. The framework is divided into three main components (Typology Of Interactive Commodities; Situational Heuristics, and Narrative Metatopics), subdivided into multiple categories. Located in the Annex I, figure I.3 illustrates the framework's components and their respective categories.

The Typology Of Interactive Commodities was developed along with degrees of abstraction of sound and object. As the authors state, sound is closely related to physical and material processes. It plays a core role in communicating "hidden" qualities of an object, such as its stability or solidity. This component is intended to help orient the sound design strategy used and is divided into the following categories:

- **Authentic Commodities** - Simple, self-contained, fitting with existing sonic identity
- **Extended Commodities** - Sound not necessarily related to object's sonic identity, communicates extension quality
- **Placeholders** - Proxies of a virtual object. Sound defines the virtual object. For example, the Wiimote or Tangible User Interfaces
- **Omnivalent Commodities** - Sound defines the artefact. Defined through software rather than physical configuration

Situational Heuristics concern the situational categories that define the relationship between interactive commodities and their use context. These categories are:

- **Social Situation** - private; public
- **Level Of Intimacy** - objectified (meaning: totally detached from human body); pocketable; wearable; implant
- **Relationship To User and Task** - assistant; tool
- **Type Of Use** - casual; professional

Narrative Metatopics are abstracted themes and attributes associated with narratively significant artefacts and interactions in fictional media, like films or games [46]. These provide a means of navigating a complex semantic space and can be associated with a collection of specific sound design strategies, which, according to the authors, serve to build grounded sonic interaction design hypotheses as a starting point for the design. They are also meant to link qualities of interactive processes with qualities of sonic processes. These are:

- Nature and judgement of artefact;
- Qualities of use;
- Qualities of control;
- Power/Energy and its qualities;
- Energy/power life cycles and dramaturgy;
- Structural states;
- Manifestation of life;
- Gesturality;
- Transformation processes;
- Temporal structure;
- Atmosphere, mood.

5.2.3 Guidelines For Voice User Interfaces

When designing Voice User Interface actions, it is essential that the system clearly states the possible interaction options since there are no visual indications for the user [36]. Nowacki et al. [37] propose a set of usability criteria tailored to VUIs, resulting in a list of 8 criteria and 18 sub-criteria. Although not all are an exact fit for the intentions of this document - since the idea is to propose voice commands that would allow users to edit UML diagrams, and not a highly sophisticated VUI - there are some important insights to take into account. They are as follows:

- Guidance - Guide the users throughout their interaction, helping them know what the system can do and learn how to use it. The guidelines in this criteria are **Prompting** - Users should always know if the system is listening or processing the request. Some linguistic markers can be used to help orient the users, such as "first", "then", "finally" and a question at the end so the user can confirm the end of the conversation or continue it. **Immediate Feedback** - Feedback allows users to feel understood or take corrective actions if needed. Reassures them while making up for the lack of visual cues.
- Workload - Aims to reduce the user's perceptual and cognitive workload and increase dialogue efficiency. The guidelines are **Brevity** - concerned with the individual commands needed to perform an end goal and the responses from the system. Respond concisely without extra words to avoid increasing the time users must spend listening. - and **Information Density** - The peaks of user attention and focus should be managed. One of the strategies is placing the new important information at the end of the sentence.
- Explicit Control - This criterion helps reduce errors and ambiguities but is also key to increasing user acceptance and trust in the system. **Explicit User Action** - The VUI should require explicit confirmation before accomplishing an action. Giving full disclosure on the impact and double-checking is crucial. **User Control** - Users should be allowed to interrupt the system or initiate another command at any moment without waiting for the VUI to finish its sentence or task. For example,

saying "stop" should immediately interrupt the action and confirm that the action has been interrupted). **Pro-active User Confirmation** - If users don't express their full intention in their request, anticipate what they might want to ask or do, and propose it in the answer.

- Adaptability - **Flexibility** - Adapt the experience to users' particular needs, capability or style. For VUIs, this includes understanding accents, speaking styles or changes in voice when a person is sick. **Users' Experience Level** - Offer shortcuts for expert users and help beginners. **Multi-user** - The VUI should recognise the "owner"user from other people who might speak to the system and provide information adapted to the user.
- Error Management - The guidelines are: **Error Protection** - systems should prevent errors from occurring by developing error prevention strategies or presenting users with a confirmation option before they commit to the action. **Quality Of Error Message** - avoid error messages that only say that the system did not understand the user correctly. Instead, add the information or question that would help users explicit their request and prevent the error from happening again, and **Error Correction** - Accept corrections, even in the middle of a command. When possible, propose the closest command to what was not understood and ask for confirmation.
- Consistency - The guidelines are: **Internal Consistency** - The same type of commands should lead to the same type of answers. **External Consistency** - Respect existing external norms and conventions to avoid confusion. Users are already surrounded by systems that make some sounds to mean error.
- Compatibility - **Short & Long-Term System Memory** - The VUI should act according to the context and track the history of information given if users allow it. Some answers must be memorised, at least in the short term, while long-term preferences can also make commands more fluid for the user. **Environment** - The VUI should be designed based on all potential contexts of use and provide different options and behaviours to adapt to different environments. The sound level can be automatically adjusted using sensors of noise levels.
- Personality - A VUI will be interpreted as a character, no matter how neutral its design is. **Identity** - Thinking of a personality for the VUI is essential to engage the user and avoid the feeling of frustration that comes from "talking to a machine". **behaviour** - The degree of politeness, usage of humour, and deciding the emotions (or lack thereof) shown in different situations are all crucial elements to gaining the trust and empathy of users. **Language** - The commands and responses should be as natural as a regular conversation with a human. VUIs should be trained to understand the basic rules of conversation and grammar.

6

Detecting Patterns In UML Diagrammatic Readings: An Experimental Study

In this chapter, a study was conducted with the main purpose of understanding if there are any emerging patterns in how different people carry out diagrammatic readings of UML diagrams (Specifically, Class and State Machine Diagrams). This is achieved by observing the most common actions among the participants. The planning, execution and results are discussed in the following sections. Furthermore, its conclusions served to elaborate proposals for the construction of tools that perform auditory readings of diagrams, presented at the end of this chapter as well. Consequently, these proposals will be used as a basis for the auditory reading performed by our developed solution.

6.1 Planning

The planning related to this activity is detailed in the sections that follow.

6.1.1 Goals

This study was conducted with the purpose of understanding any emerging patterns regarding how different users carry out a diagrammatic reading of UML Class and State Machine diagrams, by observing the most common actions among the participants. The results of this research hope to help move towards the construction of tools that perform auditory readings of diagrams, while also serving as a basis for our proof of concept tool, *SoundUML*, described in chapter 8.

In addition to the main goal, subjects were also asked a few supplementary questions, where they had to imagine they were instructing another person to perform some basic operations such as read, insert, update and remove, on the diagrams. The goal was to determine if any patterns would be observed as well, serving to detect the most interesting ways in which users interacted with the diagrams, to create proposals for the construction of diagrammatic reading tools, and subsequently, apply it to our developed solution.

6.1.2 Participants

In this research, a total of six participants were chosen, with ages between 20 and 30, consisting of both master and undergraduate students in computer engineering, since for an effective realisation of the study it was necessary to have some previous understanding of the UML modelling language. This way, it was ensured that the surveyed people expressed themselves as fully as possible, sparing no details in how they answered the questions. Their level of knowledge was varied, with a part of them having no contact with the language for some time. Although subjects could answer in their native language (Portuguese) if they so desired, they also needed to understand basic English, as the questionnaire was written entirely in English.

It was also necessary for the participants to have a somewhat decent working internet connection, as the evaluation sessions took place online.

6.1.3 Tasks

At the beginning of each individual online evaluation session, the participants were sent a .zip file containing the main document with the questions and a brief summary of Class and State Machine diagrams to refresh their memories, along with the full-sized diagram images. After extracting the contents of this file and opening the survey document, they were presented with 2 complete diagrams, both for Class and State Machine diagrams, for a total of 4. In each of these, the subjects were asked to analyse and verbally explain what they were observing - as if they were explaining the diagram to someone else who was not seeing it, imagining it as if they were talking to someone on the phone. At any given moment, respondents were able to consult the support pages containing a brief summary of the notations, in order to describe the diagrams as fully as possible.

In addition to the aforementioned exercise, the participants answered a few other questions, where they had to imagine they were instructing another person to perform some tasks on the diagrams. These simple tasks follow the 4 basic operations of create, read, update and delete (CRUD).

Finally, after answering all the questions for both class diagrams and state machine diagrams, the participants were asked if in retrospect, they would change anything in the way they described the diagrams and if they would like to give some additional information they found relevant during the evaluation process.

6.1.4 Defining The Hypothesis

Next, the hypotheses regarding the study in question are presented. The set of hypotheses is related to finding any emerging patterns regarding how different users carry out a diagrammatic reading, by observing the most common actions among the participants. The main objective of this study is to understand if any patterns regarding these diagrammatic readings of UML Class and State Machine diagrams can be found. Consequently, this study seeks to reject the null hypotheses.

The null hypothesis ($\mathbf{H_0}$) states that there are no emerging patterns in the way different users carry out a diagrammatic reading, which means that diagrammatic readings of UML Class and State Diagrams cannot be standardised. The alternative hypothesis ($\mathbf{H_1}$) states that there are emerging patterns, and that the null hypothesis is rejected.

Defining the set of hypotheses:

$\mathbf{H_{0T}}$: There are no emerging patterns regarding how different users carry out diagrammatic readings of UML Class and State Machine diagrams.

$\mathbf{H_{1T}}$: There are emerging patterns regarding how different users carry out diagrammatic readings of UML Class and State Machine diagrams.

6.1.5 Preparing The Evaluation Session

As previously mentioned, the main goal of this study is to understand any emerging patterns regarding how different users carry out a diagrammatic reading of UML Class and State Machine diagrams, while detecting the most interesting ways in which the users interact with the diagrams, to create proposals for the construction of diagrammatic reading tools, and subsequently, apply these observations to our proposed tool.

Initially, it was necessary to decide which types of UML diagrams were going to be selected for these diagrammatic readings and interactions. To accomplish a feasible and structured end result for this experimental study, we had to limit the choice of which diagram types would be used since there are numerous types of UML diagrams. We decided to choose 2 diagram types, each representing one of the 2 categories of UML diagrams (as pictured in figure 2.3), with these being Structure Diagrams, and Behaviour Diagrams. This choice allowed us to cover a significant amount of use cases, without having the study become too extensive or even impossible to be performed. In this manner, the most appropriate ones seemed to be the Class Diagrams and the State Machine Diagrams respectively, since they are among the most prominent and used in this area of expertise [53].

Once the types of diagrams were chosen, it was then necessary to elaborate the document with the questions to be asked, together with the specific diagrams to be shown to the users. We decided to present 2 complete diagrams, both for Class and State Machine diagrams, with one of these being simpler, and the other more elaborate. Thus, it is ensured that in a first exercise, the participants understand what is being asked so they can describe it more concisely. Afterwards, they are presented with the more complex diagram, where the purpose is to have subjects navigate more thoroughly, requiring them to plan a further detailed strategy to carry out the diagrammatic reading.

In addition to these diagrammatic readings, a few more questions were created, where the participants would have to imagine they were instructing another person to perform some tasks on the diagrams. It was decided that these simple tasks should follow the 4 basic operations of create, read, update and delete (CRUD). These questions have the purpose of understanding how the respondents interact with each diagram and whether any patterns are also found in these interactions. Furthermore, it should be noted that in these questions we opted to label the classes, states and their respective elements, by letters (corresponding to the question's paragraph letter) instead of their names, to observe how the users chose to refer to this information in the diagrams as well. We also presented a third class diagram, much smaller in size, with a more specific question, to determine how they would refer to two different types of associations between the same two classes.

As previously mentioned, the participants' level of UML knowledge was varied. In order for the experience to go as smoothly as possible, and for all the inquired to start on equal footing, and be able to express themselves in the best way, support material was also prepared, in addition to the previously mentioned document with the questions to be asked. This support document sought to summarise the subjects of both Class Diagrams and State Machine Diagrams, as well as their notations. These pages were embedded within the experiment document itself, and a decision was made to allow participants to consult them at any time, to make their reading of the diagrams as clear as possible. A yellow background was also placed on these pages to make them easily distinguishable.

Thus, the document is organised as follows: the first page contains an introductory text, as reproduced in figure 6.1. This brief text frames the study in the context of this work while its goals are explained, together with information specifying that by agreeing to participate, the subjects are granting their consent to record the audio of the experiment, in order to be transcribed into text and analysed. The rest of the document is organised into 2 sections. Right after the introduction on the first page, we have the 1st section, regarding the Class diagrams. The 2nd section is related to State Machine diagrams. At the beginning of each of these sections, on the pages with a yellow background, the aforementioned support material with a brief summary of the notation is presented, followed by the exercises related to the respective diagram type of that section.

After choosing the diagrams, creating the document with the questions and the diagrams to be shown to the participants and the support material, everything was placed in a .zip file, simplifying the process of sending the participants all the necessary material for the experience. This file also included the image of each diagram in its full resolution, to allow the participants to navigate them without any restrictions.

Finally, a pilot test was conducted with a colleague. This initial study took around 100 minutes, so it was necessary to make some changes to the experiment to shorten it, as with such an extended duration, it was going to be particularly difficult to get participants or even results that could be concisely discussed. With the resulting feedback from this pilot experiment, a few changes were made to details in the document along with the support material, and some of the questions were rewritten, so as to make it clearer what is being asked, and questions which were not so relevant to the study were removed as well.

It was then decided to explicitly ask participants to not describe in detail the attributes and methods of each class when performing the diagrammatic reading of the class diagrams. This decision was justified after verifying during the pilot test, that the detailed description of the attributes and methods of each class during the diagrammatic reading, did not add anything to the main objective of this study, as the intention is to understand how users choose to parse the diagrams, by observing the classes they tend to read first and which paths they take afterwards, and not exactly the way they carry out the same repeated task of reading the attributes and methods contained within each class. However, to not completely ignore the reading of class attributes and methods, a question was created in which participants were asked to describe a given class in detail. This decision made even more sense as there were already questions for the create, update and remove operations, with this one being essentially a "read" operation of a given class, thus following the 4 basic operations (CRUD). With these changes, the experience lasted an average of 60 minutes. Although not ideal, it became much more feasible to perform it. It is also worth noting that testing the experiment with someone we knew beforehand was ideal because it allowed for completely honest feedback on what changes needed to be made, namely in terms of what was a reasonable amount of time to ask from the participants, to carry out the experiment.

The introductory text presented to the participants is reproduced in figure 6.1. The complete document that was elaborated for this study can be consulted in Appendix A.

This experiment is being developed within the scope of the thesis "Constructing An Auditory Notation In Software Engineering - Editing UML Models with voice and sound", and aims to detect possible emerging auditory patterns in the way UML models are read.

To this end, **a total of 3 Class diagrams and 2 State diagrams will be presented**. In each of these, **you will be asked to analyze and verbally explain what you are observing** - as if you were explaining the diagram to someone else who is not seeing it (you can imagine it as if you were talking to someone on the phone). You can start by specifying from where you will start. Remember, there isn't a right or wrong answer to how you describe the diagrams, so don't feel pressured. Simply describe them in the best way you think it's possible!

The diagrams will be contained in this document, but in order to facilitate their analysis, the full-sized image of each of them, is also available within the .zip file.

Additionally, after this task, we'll ask you to answer some supplementary questions.

By agreeing to participate, you are granting your consent to record the audio of this experiment, in order to be transcribed into text and analyzed. The results of this study will be public but the data will be anonymized, guaranteeing total confidentiality.

In order to refresh your memory, **a brief summary of Class and State diagram notations is given below**, in the pages with a yellow background. While answering the questions, you can always go back and consult these pages.

Thank you for participating!

Figure 6.1: The document's introductory text, presented to the participants.

6.1.6 Experimental Material

The evaluation sessions took place online, and no restrictions were placed on what software should be used to carry out this study, as it could be done through any software at the preference of the participants. However, all sessions were held through the VoIP platform *Discord*, as it is a popular software, and the subjects already had it installed on their devices. The devices used were left to the choice of each participant as well, however, all of them chose to use their personal computers, as it was more practical. Logically, it was also necessary to use a monitor or some type of visual display, to visualise the files necessary for the realisation of the evaluation, with these being the .zip file containing the main document in a .pdf format, along with the full-sized diagram images.

The subjects also needed to have a reasonable internet connection, as well as speakers or any type of headphones capable of audio playback, so that they could hear the instructions they would be given during the study, along with a microphone, to be able to answer the questions.

On our behalf, a personal computer was used, with the software *discord* installed, a good internet connection, headphones and a microphone. The recordings were made through the *OBS Studio* application, being later transcribed by using *VLC media player* and *Microsoft Excel*.

6.2 Procedure Of The Evaluation Sessions

At the beginning of each individual online evaluation session held through the VoIP platform *Discord*, it was briefly explained how the study would be carried out, and that by agreeing to participate, the subjects were granting their consent to record the audio of the interview, in order for the data to be transcribed into text and analysed, with the results of this study being anonymised, guaranteeing total confidentiality. All sessions were carried in Portuguese, the native language of all the inquired.

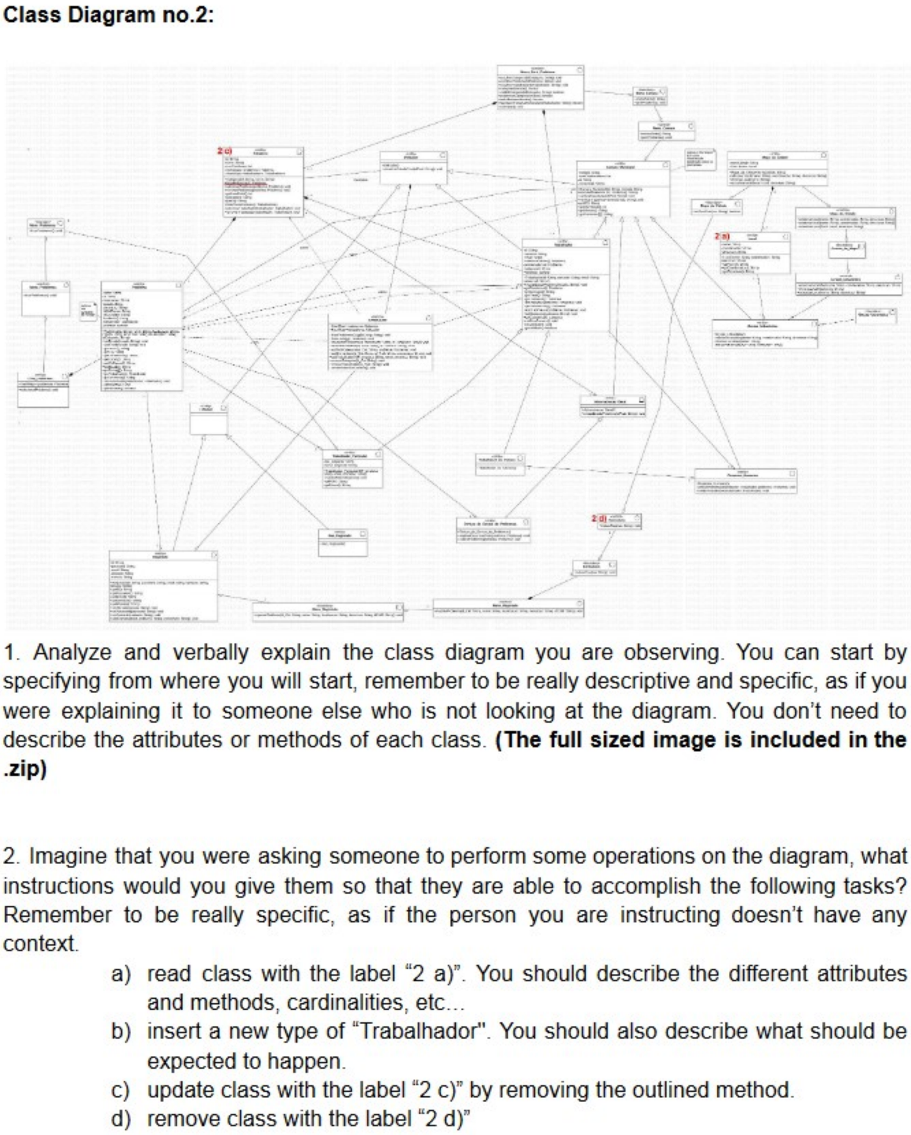
Afterwards, participants were sent a .zip file containing the main document in a .pdf format, along with the full-sized diagram images. In order to refresh their memory, and to assure that all of them were on equal footing when answering the questions, a brief summary of Class and State diagram notations was contrived, and presented on the pages with a yellow background within the main document. While answering the questions, the subjects could always go back and consult these pages. This ensured that all answers would be as complete as possible.

Following the extraction of the contents within the .zip file, participants were given instructions to open the main document. After the introductory text on the first page, the aforementioned summary of the Class diagrams notation was followed by the questions to be asked to the participants. The 1st class diagram was slightly less complex, and the subjects were only asked to describe it. The 2nd class diagram presented a higher level of complexity, forcing the subjects to navigate it more thoroughly, requiring them to plan a more detailed strategy to carry out the diagrammatic reading. Additionally, in this diagram, the questions that follow the CRUD operations were performed, being specifically requested that the participants read, create, update and remove a class, in that order. Finally, a small third class diagram was presented, where instead of having to describe it, respondents were asked a more specific question about how they would remove one of two associations between the same two classes. The goal of this question is to understand how the subjects would refer to a specific association. After these questions related to the Class diagrams were answered, two additional questions were asked, regarding if after the task of describing the diagrams and being more familiar with them, participants would change anything in the manner they explained them, and if they would like to give some additional information they found relevant, concerning the class diagrams.

Moving on to the second section of this study, following the summary of State Machine notation, the questions were presented, along with their respective State machine diagrams. In this section, two diagrams were shown to the subjects. In the first diagram, participants were asked to describe what they were observing, and the questions following the CRUD operations were asked. Participants were asked to read, create, update and remove a state, in that order. In the second diagram, participants were only asked to describe it. Finally, just as in the first section related to Class diagrams, two additional questions were asked, regarding if participants would change anything in the way they explained the state machine diagrams, and if they would like to give some additional information regarding these diagrams.

For the data collection, the sessions were recorded using *OBS Studio*. Regarding the processing of this data, these files were then listened to using *VLC* for audio playback, and then transcribed into a *Microsoft excel* document. A transcription using speech-to-text was attempted at first, but considering that the texts were in Portuguese and contained words from a technical field, the results were not accurate nor satisfactory, so we opted to transcribe the texts manually. Since the sessions were conducted in Portuguese and their duration was somewhat extensive, with an average of around 60 minutes, for the purposes of this document and due to time constraints, it was not an exact transcription regarding what each of the subjects answered, since otherwise, the treatment of results would be a very extensive and unfeasible process. As such, it was necessary to structure the transcriptions and to have some method for the way in which they were carried out, so that they maintained the original meaning while still allowing for a comprehensible reading. This approach is further detailed in Section 6.3.

Figure 6.2 depicts an example of a set of questions within the survey document. These questions correspond to a class diagram, but the same process was followed for state machine diagrams, as the questions are quite similar, with the difference of being applied in the context of state machine diagrams. In the first question, subjects were asked to describe the diagram as if they were explaining it to someone who was not looking at it, while in a second exercise, they had to give some instructions, as if they were asking someone to perform some operations on the diagram. The goal here was to understand how the respondents interacted with the diagram and if any patterns would also be found in these interactions. Since this diagram was rather extensive, the full-sized image was included in the .zip file sent.


Class Diagram no.2:

1. Analyze and verbally explain the class diagram you are observing. You can start by specifying from where you will start, remember to be really descriptive and specific, as if you were explaining it to someone else who is not looking at the diagram. You don't need to describe the attributes or methods of each class. **(The full sized image is included in the .zip)**

2. Imagine that you were asking someone to perform some operations on the diagram, what instructions would you give them so that they are able to accomplish the following tasks? Remember to be really specific, as if the person you are instructing doesn't have any context.
   a) read class with the label "2 a)". You should describe the different attributes and methods, cardinalities, etc...
   b) insert a new type of "Trabalhador". You should also describe what should be expected to happen.
   c) update class with the label "2 c)" by removing the outlined method.
   d) remove class with the label "2 d)"

Figure 6.2: Example of a set of Class diagram questions in the document presented to the participants.

6.3 Results

Each interview was transcribed following the procedure described as follows, translated into English and tabulated. This table can be consulted in a repository at Zenodo[1]. It is available as both an excel and a .csv file.

[1]Detecting Patterns In UML Diagrammatic Readings - An Experimental Study: https://doi.org/10.5281/zenodo.7782463

6.3.1 Transcription Procedure

As previously mentioned, this study was conducted in Portuguese, as this is the native language of the participants and the language in which they felt most comfortable expressing themselves and answering the questions as fully as possible. The duration of this study was somewhat extensive as well, seeing that we sought detailed descriptions for all the questions. Therefore, it is not an exact transcription, since otherwise, the treatment of results would be a very extensive and unfeasible process. As such, it was necessary to develop an approach for how the transcriptions were carried out, to structure them so that they maintained the original meaning of what was answered by the subjects, still allowing for a comprehensible reading, and enabling the extraction of results.

For that purpose, the text was left as similar as possible to the way the participants described the diagrams, only removing small details that were not so relevant to the specific question, with those details being provided in the two additional questions asked at the end of each of the two sections, the former regarding the Class Diagrams and the latter, State Machine Diagrams. The answers given by the inquired were translated from Portuguese to English, and in the class names, we opted to leave them as they are in the diagrams, to make it simpler when consulting these transcriptions along with the diagrams. Finally, some corrections that the subjects wanted to make while describing the diagrams - for example, going back in the diagram to describe something, either because they forgot or were confused - were immediately made in the correct location of the text, as to make the transcriptions more streamlined. If these corrections also added some value to the diagrammatic readings, a note was left.

It should be noted that in general, the main idea of these transcriptions was to understand the logic behind how subjects carried out the diagrammatic readings, with the main goal of verifying if any patterns would emerge. This means that there might be some mistakes in the readings and some lapses by the subjects - namely if an element had already been described, some gaps in the description of multiplicities, not describing some transitions in state machine diagrams, etc. Some of these mistakes were cleaned up to streamline the information, but most of them were kept in the transcriptions as is, as we think that these human "errors" were part of the process of understanding how subjects would approach and describe the diagrams.

6.4 Discussion Of Results

The discussion of results is divided into two subsections, these being the class diagrams section and the state diagrams section. In each, we analyse the results on a question-by-question basis, detailing the most relevant findings. This is followed by a final section with a summary of our observations and conclusions.

6.4.1 Class Diagram Section

In this section, the results regarding the class diagrams questions are presented, along with their subsequent discussion.

6.4.1.1 Class Diagram No.1

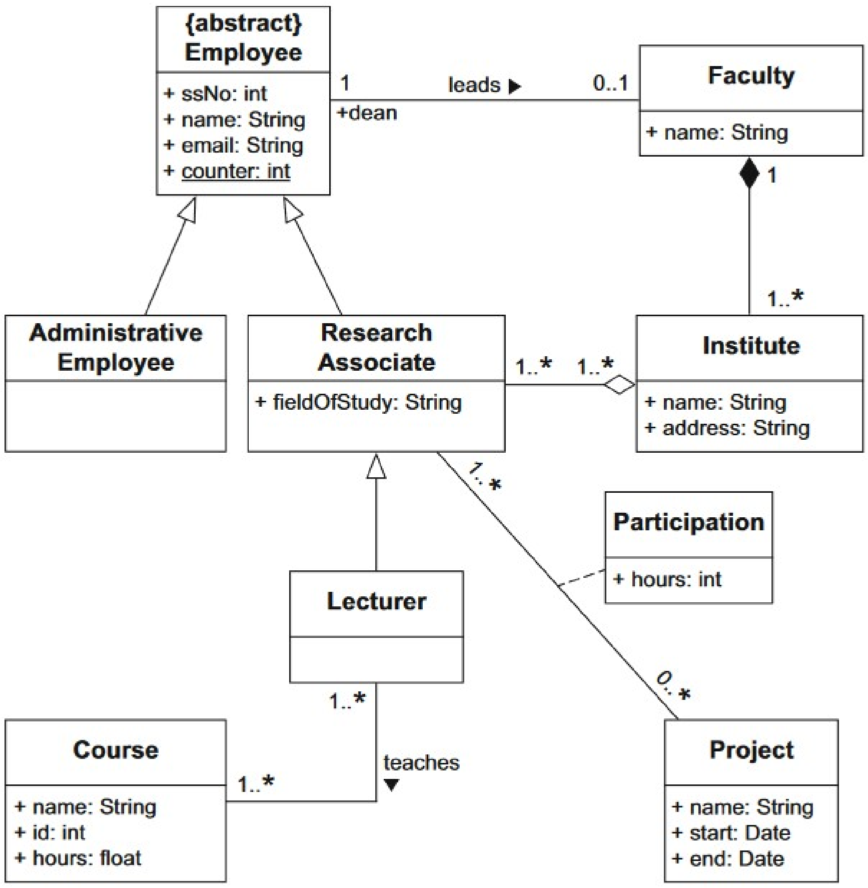


Figure 6.3: Class Diagram from Question 1. Taken from [54].

In **question 1**, the participants were asked to analyse and verbally explain the class diagram located in figure 6.3. Before beginning their diagrammatic readings, 3 of the inquired people mentioned that they would start from top to bottom, and left to right, as visually, it made the most sense to them. All of the subjects, without exception, started their diagrammatic reading by either the abstract class "Employee", or the class "Faculty", with this choice being evenly split. After mentioning the class "Employee", most of the partakers followed by describing the inheritance relationships with classes "Administrative Employee"and "Research Associate", as it seemed to be the logical step to take afterwards. From here, the readings started to slightly diverge. However, from our observation, it seems that although the order in which the subjects got to certain classes differed, they all divided the diagram into chunks, in order to be more easily dissected. These chunks seemed to be more or less the same as they followed the flow of the relationships that a given class has. For example, in the case of "Research Associate", the respondents tended to divide its relationships into two smaller parts, one with the classes "Lecturer" and "Course", and the other with the class "Project" and its class association. As participants described the relationships from a given class, they seemed to prioritise the inheritance relationships. One explanation for this could be the fact that these relationships suggest some kind of hierarchy, giving the subjects a sense of importance to different classes.

After the task, one of the participants mentioned that what they found the most difficult was going from these top classes to the classes underneath, as it seemed natural to choose where to start, since the relationships had a natural flow, but as they progressed, it became more difficult.

From our scrutiny of the answers given to this question, we observe that the subjects intuitively favoured the classes that were visually located at the top of the diagram. Generally, the participants more naturally opted to describe the relationships of a given class as soon as they mentioned that specific class. In describing these relationships, they gave priority to inheritance relationships, something that could possibly be explained by a need to assign different levels of importance and priorities to different classes. The classes visually located at the extremities - the classes "Course" and "Project" - were the ones that participants mostly left for the end of their readings, possibly due to their importance in the diagram as a whole, or even their visual location.

6.4.1.2 Class Diagram No.2

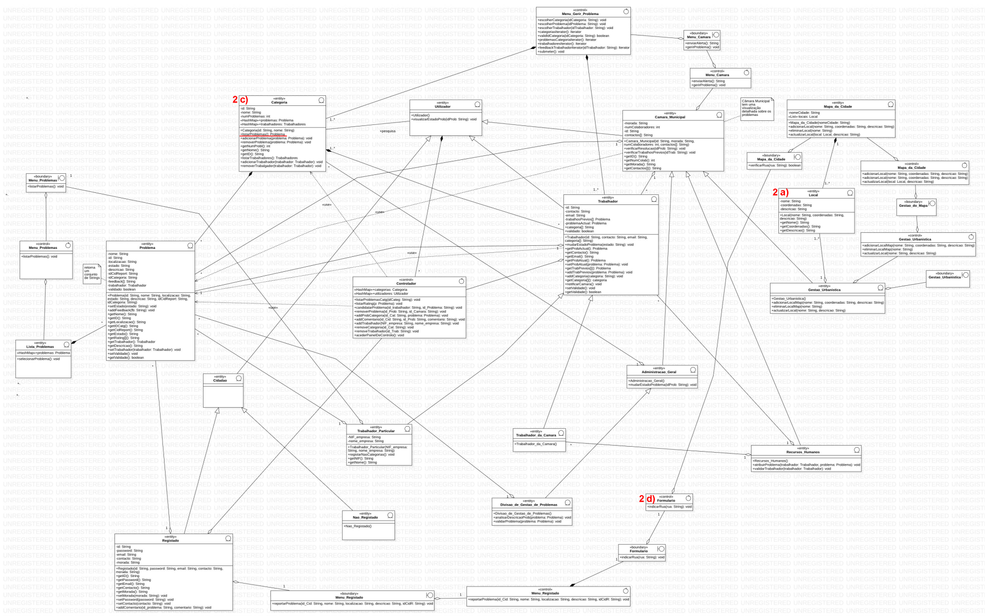


Figure 6.4: Class Diagram from Question 2.

The class diagram illustrated in figure 6.4 was presented to the participants. A full-sized version of this diagram can be found in Appendix A.1. In the first question, they were asked to analyse and verbally explain it. This diagram is intentionally large, in order to encourage the subjects to navigate it further, while potentially increasing the variation in terms of responses, compared to the previous diagram. As this diagram is relatively sizeable, it was chosen to have a set of additional questions that follow the 4 CRUD operations, with the purpose of understanding how the respondents interact with it. In these questions we opted to label the classes, states and their respective elements, by letters (corresponding to the question's paragraph letter) instead of their names, to observe how the users chose to refer to this information in the diagrams as well.

Given that the diagram has the class names in Portuguese, in order to help the reading of this document and give some context to the diagram, we have translated the class names as follows:

```
Menu_Gerir_Problema — Menu_Manage_Problem; Camara_Municipal — City_Hall;
    Categoria — Category; Utilizador — User; Controlador — Controller;
    Menu_Camara — Menu_City_Hall; Trabalhador — Worker; Trabalhador_Particular
    — Worker_Private; Trabalhador_da_Camara — Worker_City_Hall; Problema —
    Problem; Menu_Problemas — Menu_Problems; Lista_Problemas — Problems_List

Cidadao — Citizen; Nao_Registado — Unregistered; Registado — Registered;
    Menu_Registado — Menu_Registered;Formulario — Form

Administracao_Geral — General_Administration; Recursos_Humanos —
    Human_Resources

Gestao_Urbanistica — Urban_Management; Local — Place; Mapa_da_Cidade —
    City_Map; Gestao_do_Mapa — Map_Management
```

In **question 1** of this diagram, we can observe that 3 of the participants started their diagrammatic reading by the class "Camara_Municipal", 2 by the class "Menu_Gerir_Problema" and the remaining, by the class "Utilizador". Their readings continued to one of the already mentioned classes. Specifically, the 2 participants that started their readings by the class "Menu_Gerir_Problema", together with the one that started by "Utilizador", went on to describe the class "Camara_Municipal". The ones that started by specifying the class "Camara_Municipal", coincidentally, followed to the class "Menu_Gerir_Problema". Afterwards, in a general manner and not necessarily in this order, the readings moved on to the classes "Trabalhador", "Categoria" and "Problema". This may be an indication that although the order in which these classes were read differed from person to person, they were identified by the subjects as being the most important classes in the diagram. This could be explained by the number of existing relationships in these classes, namely the inheritance relationships, something that could suggest a need to assign different levels of importance and priority to different classes. Another factor to have into consideration may be their visual location, since subjects generally read the diagram starting from top to bottom.

As with the previous question, participants generally broke the diagram into different chunks, so as to be more easily parsed. The most common breakdown was in 4 parts.

Starting by the topmost part, this includes the classes: "Menu_Gerir_Problema", "Camara_Municipal", "Utilizador", "Trabalhador", "Controlador"and "Categoria"; the left side, which includes: "Problema"; "Cidadao"; Registado", "Menu_Problemas"and "Lista_Problemas"; The lower part, featuring: "Menu_Registado", "Formulario"and the classes that inherit from "Trabalhador"and "Camara_Municipal". Finally, the last chunk was the one located at the right side of the diagram. with the classes: "Mapa_da_Cidade", "Local", "Gestao_do_Mapa"and "Gestao_Urbanistica".

The classes that the participants chose to detail towards the later end of their readings, were those that have fewer relationships and are visually located at the extremities of the diagram, namely the classes "Menu_Registado", "Formulario" and "Lista_Problemas". One participant's approach was to describe the relationships in one direction only, meaning that they only described the outgoing relations from a given class, as it seemed more intuitive to them. Another respondent opted to start their reading by listing all of the inheritance relationships in the diagram, only afterwards detailing the diagram itself. Some of the subjects mentioned that they felt some difficulties performing this task, such as keeping track of what had already been described or not, as the diagram was very extensive. Two participants said that they could have described certain elements of the diagram better, such as the relationships, or the types of classes, respectively. One of the subjects stated that he would probably have started their reading by the class "Problema"or "Camara_Municipal" as these have a great number of relationships, thus, covering a bigger part of the diagram, establishing an outline for the classes of the diagram.

Although the order of the classes started to significantly vary as the participants carried out their diagrammatic readings, we can generally affirm that they attributed greater importance to the classes that had the most relations, as they started their readings from them. As in the previous question, the inquired people seemed to prioritise the inheritance relations when describing the relationships from a given class, and also divided the diagram in chunks, where the classes visually located at the extremities were the ones that respondents mostly left for the end of their readings. This was possibly because they had fewer relationships (and as such, regarded them as less important) or even because of their visual location in the diagram. Analysing the visual aspect, the subjects generally read the diagram starting from top to bottom, breaking it into 4 different chunks, so as to be more easily parsed.

The questions that followed, requested that the subjects imagined they were instructing someone to perform some operations on the diagram. As previously mentioned, these questions follow the 4 CRUD operations, and have the purpose of understanding how the respondents interact with the diagram. These questions aimed to be direct and answered briefly.

In **question 2a)**, participants had to read the class with the label "2a)". All participants started their readings by the name of the class, "Local", described then its variables, "nome", "coordenadas"and "descricao", with all 3 being of type String and private. Afterwards, they mentioned the constructor, stating that it receives the String variables "nome", "coordenadas" and "descricao". Subsequently, all the subjects detailed the methods, stating that there were 3 get methods, which are public, for each of the variables previously described, these are: "getNome", "getCoordenadas" and "getDescricao". Finally, as the last step in this exercise, the respondents specified the relationships of the class "Local", which are a composition and an aggregation. A composition with entity class "Mapa_da_Cidade", where 1 "Mapa_da_Cidade" is composed of 1 or many "Local", and an aggregation with entity class "Gestao_Urbanistica", where 1 "Gestao_Urbanistica" aggregates 1 or more "Local". In summary, we observed that the natural order in which the subjects described this class, was to start by reading its name, followed by its variables, then by its constructor and methods, finishing with the relations that this class possesses, along with its multiplicities.

Afterwards, **question 2b)** saw the participants inserting a new type of "Trabalhador", while describing what should be expected to happen. Initially, 4 of them gave a name of their choosing to this new class. All the subjects without exception mentioned that it would be necessary to create an inheritance relationship with the class "Trabalhador", where "Trabalhador" would be the parent, meaning that this new class would inherit from it. 3 out of the 6 inquired, mentioned that this class would naturally have its constructor. Finally, 4 of them mentioned that this class would probably have some relationships with other classes as well, in the same way that the other classes that inherit from "Trabalhador" do, with 2 of them going even further, stating that these relations could be for example, an aggregation relationship to class "Problema".

In **question 2c)**, it was asked that the subjects updated the class with the label "2 c)" by removing the outlined method. We can observe that all of the subjects referred to this class by its name, "Categoria", with 3 of the participants giving further context of where this class was situated in the diagram, mentioning that this class has a composition relationship with the class "Menu_Gerir_Problema". Coincidentally, the class "Menu_Gerir_Problema" was the class by which 2 of these 3 had started their diagrammatic reading in exercise 2, with the other having started by the class "Camara_Municipal". After this brief contextualisation, all of the subjects removed the outlined method, referring to it by its name, "listarProblemas".

Question 2d) asked subjects to remove the class with the label "2d)". The subjects, without exception, started by referring to this class by its type and its name: control class "Formulario". All of them also mentioned that to remove this class, it would be necessary to delete its relationships as well. Additionally, 4 of the 6 inquired specified these relations, with them being the aggregations with boundary classes "Formulario" and "Mapa_da_Cidade".

6.4.1.3 Class Diagram No.3

In this question, participants were shown the diagram present in figure 6.5, where there are two associations between the same two classes. Subjects were asked to imagine they

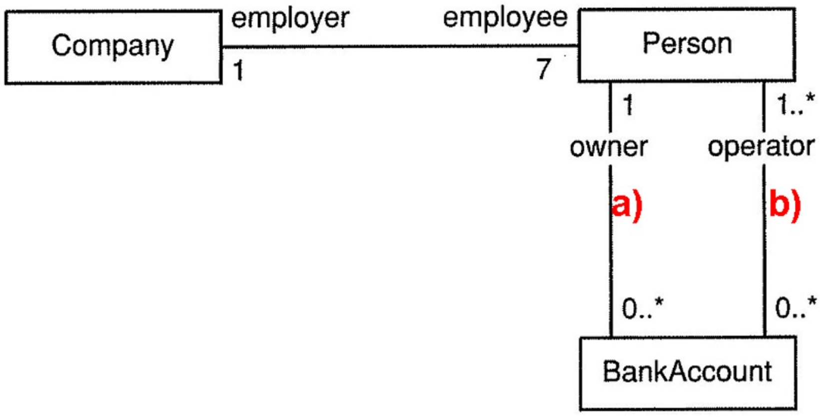


Figure 6.5: Class Diagram from Question 3. Adapted from [54].

were instructing someone to remove the association with the label "a)". This question had the purpose of understanding how they would distinguish this association from the one with the label "b)", so that this differentiation could be implemented as a human-computer interaction, in a future solution. In order to not influence the answers, we decided to give each of these associations a label a) and b).

Subjects mostly referred to the association to be removed by its role, "owner", with most also subsequently describing its multiplicity, suggesting that the roles and multiplicity were the most obvious ways to distinguish between these two associations. Only two of the respondents prioritised the differences between the multiplicities of both associations, instead of the role, only mentioning the role "owner" afterwards. During their answers, four of the participants described that there were two relationships between the two classes. Two of them felt the need to mention that by removing the association with the "owner" role, the other association present in the diagram, with the "operator" role would remain and not be removed.

6.4.2 State Machine Diagram Section

In this section, the results regarding the state machine diagrams questions are presented, along with their subsequent discussion.

6.4.2.1 State Machine Diagram No.1

The state machine diagram illustrated in figure 6.6 was presented to the participants. As was done for the previous section, in the first question, they were asked to analyse and verbally explain it. This was followed by a set of additional questions that reflect the 4 CRUD operations, where the subjects had to imagine they were instructing someone to perform these operations on the diagram.

As already mentioned, in **question 1** the participants were asked to analyse and verbally explain the state diagram they were observing. Naturally, all of them started by describing the initial state, which transitions to the state "Study Program Inactive". 1 of

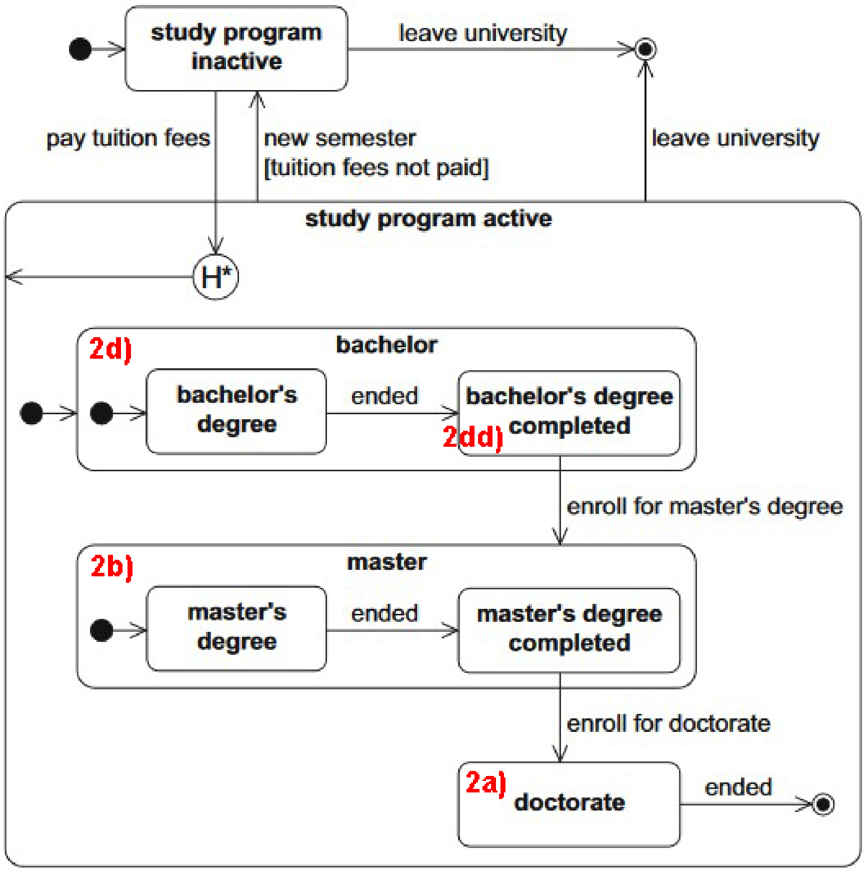


Figure 6.6: State Machine Diagram from Question 1. Adapted from [54].

the respondents, before starting his diagrammatic reading, made a point of mentioning the two states that stood out the most due to their transitions and sub-states, with those being the "Study Program Inactive" and "Study Program Active". From here, 4 of the respondents preferred to describe right away the two outgoing transitions from this state: "leave university" and "pay tuition fees". The remaining 2, opted for an approach where they described the main path, and only at the moment when it was justified - more towards the end of their readings - describing these transitions.

When entering the "Study Program Active", all subjects mentioned the transition to a deep history state within it. 4 of the subjects chose to describe what happens in this deep history state as soon as they mentioned it, while 1 participant only decided to describe it further in their diagrammatic reading. From here, there was not much variation within the "Study Program Active" state, as there was exclusively one sequential path. All participants described the sub-state "Doctorate"towards the end of their readings, together with its transition "ended", to the final state. At the end of their diagrammatic readings, was when some of the subjects decided to mention the outgoing transitions of the state "Study Program Active", together with the transition "leave university", as well as the transitions where it was possible at any time to leave that state or return to a previous one. This corroborates what has already been said above, concerning the approach participants adopted to describe the transitions.

In summary, we can observe that the participants followed a certain order in their readings, something easily explained by the fact that there is a sequential order in this type of diagrams, with initial and final states helping to maintain a certain order. Where we believe there was more variation in the answers given, was in relation to the description of the transitions, where we could identify two approaches: In the first, the subjects preferred to mention right away the possible transitions at a given state, while the others preferred to only mention the transitions after describing the sub-states, that is, the inquired first described what they considered the main path and only afterwards, detailed what they did not include in this flow. The former seemed to be the most preferred option by the participants.

In **question 2a)**, participants had to read the state with the label “2 a)” and describe its context and transitions. All of the subjects started by referring to this state 2a) by its name, "Doctorate", followed by its incoming and outgoing transitions, along with the sub-state that directly precedes it (the state "Master’s Degree Completed" and the subsequent transition "Enrol For Doctorate"). All participants also detailed that the outgoing transition "ended" leads to a final state. 4 of the participants described the bigger context of "Doctorate" in the diagram, detailing the state in which it is inserted, with "Doctorate"being the last sub-state within "Study Program Active". 1 of the participants while describing the outgoing transition "ended", mentioned that besides this possible transition more directly observed, there also were the possible outgoing transitions from the state "Study Program Active", with these being the transitions "leave university", which also leads to a final state, or "new semester", which leads to the state "Study Program Inactive". In summary, we observed that the natural order in which the participants described the state with the label "2 a)" was to start by detailing its name, followed by the state and transition that directly preceded this state "Doctorate" (In essence, providing a "local" context), moving on to describe the possible outgoing transitions from "Doctorate". Additionally, the majority of participants provided a brief depiction of the "general" context in which "Doctorate" is inserted, by detailing that it is inserted in "Study Program Active", along with the description of this state’s outgoing transitions.

Afterwards, **question 2b)** saw the subjects inserting a new type of study program before the state with the label “2b)”, while describing what should be expected to happen. All of the participants understood that this new type of study program would be created after the state "Bachelor", specifically after the sub-state "Bachelor’s Degree Completed", even though this was not specifically stated in the question. They always indicated that there would be a transition from "Bachelor’s Degree Completed" to the new state, and from the new state to the "Master" state, giving them names such as "Enrol For New Study Degree" and "Enrol For Master’s Degree", respectively. In short, they started by giving a name to this new state, and then described the transitions needed to connect this sub-state to both the states that preceded and followed it. 4 out of the 6 subjects further described how this state could be internally specified, mentioning that it would be similar to the pre-existing "Bachelor" and "Master" states, where this new state would have an initial state, which would transition to a sub-state where this new type of study would start, having then a transition "ended"to another sub-state where this new type of study program would be completed. From this last sub-state, there would be a transition labelled "Enrol For Master's Degree", which would connect to the state "Master".

In **question 2c)**, it was asked that the participants removed the state with the label "2 a)". All of them started by identifying the state by its name, "Doctorate". After this, they connected the preceding state to the one that followed: In this case, they connected "Master's Degree Completed" to the final state, removing the state "Doctorate" and its transitions. 2 of the subjects gave some more context to "Doctorate" as well, referring that it is located within the state "Study Program Active", and that it transitions from the state "Master", specifically from the sub-state "Master's degree completed".

Finally, in **question 2d)** subjects were instructed to update "2d)" with a new state before the state with the label "2 dd)". All of the subjects understood and affirmed that this new state would be inserted between the states "Bachelor's Degree" and "Bachelor's Degree Completed", even though this was not specifically stated in the question. From here, all participants detailed that two transitions would be needed, an incoming transition from "Bachelor's Degree" to this new state, and an outgoing transition from this new state to "Bachelor's Degree Completed" naming them accordingly.

6.4.2.2 State Machine Diagram No.2

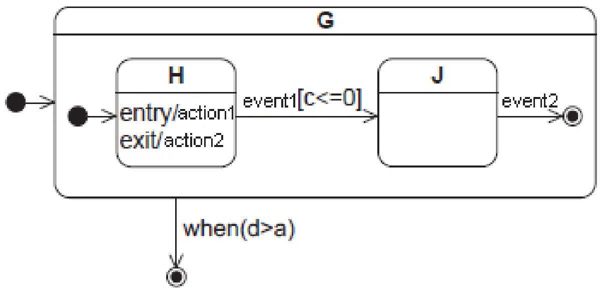


Figure 6.7: State Machine Diagram from Question 2. Adapted from [55].

In the diagram represented in figure 6.7, participants were only asked to analyse and verbally explain the state diagram they were observing. As with the answers to the previous diagram, it was noted that the participants followed a certain order in their readings. The description of the transitions was where the greatest variability was found, where we could identify the two approaches already mentioned. All subjects without exception, when entering state "H", immediately described the activities contained within it. The answers were evenly split in regards to describing that at any time, when d is greater than a, one can transition from state "G" to a final state. Half of the participants chose to mention this as soon as they described the state "G", while the other half opted to refer it at the end of their readings, after detailing everything contained within "G".

6.4.3 Summary

A brief summary of the previous discussion of results, together with our observations is presented. As was the case with the aforementioned discussion, the summary has been divided into two sections, as these insights and conclusions concern two different types of UML diagrams.

6.4.3.1 Summary Of Class Diagram Findings

From our scrutiny of the answers given to the diagrammatic readings of class diagrams, it was possible to identify some patterns. It seems that the subjects assigned different levels of importance to the classes, suggesting that they attributed greater importance to the classes that had the most relations, as they started their readings from them. Generally, when describing the relationships of a given class, the participants more naturally opted to describe these relations as soon as they mentioned that specific class. In both diagrams, when describing the existing relationships, they gave priority to the inheritance relationships, corroborating what was previously stated about the necessity to assign different levels of importance and priority to different classes.

The classes visually located at the extremities were the ones that participants mostly left for the end of their readings. This was possibly due to the fact that these had fewer relationships (and as such, participants regarded them as less important) or even because of their visual location in the diagram. In regards to the visual aspect, the subjects generally read the diagrams starting from top to bottom, breaking them into different chunks, so as to be more easily parsed.

After performing this task, some of the subjects mentioned that they felt some difficulties, such as keeping track of what had already been described or not. One of the participants mentioned that what they found the most difficult was going from the top classes to the classes underneath, as it seemed natural to choose where to start, since the relationships had a natural flow, but as they progressed, it became more difficult.

When describing a given class to its fullest detail, we observed that subjects started by reading its name, followed by its variables, then its constructor and methods, finishing with the relations that the class possesses, along with its multiplicities. When inserting a new class in the diagram, the majority of participants started by specifying a name for this new class, followed by its relationships. In this case, the class was a new type of "Trabalhador", where the inquired understood that an inheritance relationship needed to be created. Afterwards, subjects went on to describe the constructor and possible methods. To update or remove a class, participants without exception, started by referring to the classes by their names, followed by the instruction they wanted to give, in this case, Finally, in the third class diagram that was presented, subjects were asked to imagine they were instructing someone to remove the association with the label “a)”. They mostly referred to the association to be removed by its role, "owner", with most of them also subsequently describing its multiplicity, suggesting that the roles and multiplicity were the most obvious ways to distinguish between the two associations.

6.4.3.2 Summary Of State Machine Diagram Findings

Regarding the state machine diagrams, the readings were a more natural process than the readings of the class diagrams, with the participants following a certain order. This is explained by the fact that these diagrams are somewhat sequential, having initial and final states. As such, the subjects naturally started their readings by the initial state. Where we observed more variation in the given answers, was in regard to the description of the transitions. Here, we could identify two approaches: In the first, the subjects preferred to mention right away the possible transitions at a given state, while in the other, they preferred to only mention the transitions after describing the sub-states, that is, the respondents first described what they considered the main path of the diagram and only afterwards, detailed what they did not include in this flow. The former was to be the most preferred option by the participants. Specifically, in the second state machine diagram that was presented, we corroborated what had been previously detailed, along with the observation that all the respondents described the activities of a particular state as soon as they were describing that particular state.

When describing a given state, the answers given seem to suggest a natural order followed by the participants. We observed that they started by detailing the name of the state - which, in this case, was "Doctorate" - followed by describing the directly preceding state and transition, moving on to describe the possible outgoing transitions from "Doctorate" itself. The majority of participants also described the bigger context of "Doctorate"in the diagram, stating that it is contained within another state, "Study Program Active", while also mentioning that state’s outgoing transitions, and as such, providing a brief depiction of the "general" context in which "Doctorate" is inserted. When inserting a new state in the diagram, subjects started by giving a name to this new state, and then described the transitions needed to connect it to both the states that preceded and followed it. Although it was not explicitly requested, the majority of participants further described how this state could be internally specified. To remove a state, all of them started by identifying the state by its name, followed by connecting the preceding state to the one that followed. Finally, in order to update a state with a new sub-state, the subjects proceeded in the same manner as when inserting a new state in the diagram, by giving a name to this new state, and then detailing the transitions needed to connect it to both the states that preceded and followed it.

6.5 Proposals For The Construction Of Diagrammatic Reading Tools

As previously stated, one of the goals of this study is that the results of this research help move towards the construction of tools that perform auditory readings of diagrams. As such, following the discussion of results, together with the emerging patterns observed, a number of proposals have been prepared. Additionally, some ideas that arose during the discussion of results that could be interesting to be implemented, are discussed as well. As was the case with the previous section, the proposals have been divided into two sections, as these concern two different types of UML diagrams.

Before all else, our proposal would be to use a TTS, thus making it possible to transmit the intended information to the user in an auditory manner.

6.5.1 Proposals For Class Diagrams

Regarding the reading of the class diagrams, the tool should start by presenting the total number of existing classes and other elements in the diagram, so that the user can have a first "naive" impression regarding the size and complexity of the diagram. Based on the previous discussion, it was noted that the participants assigned different levels of importance to the classes, suggesting that they attributed greater importance to classes that had the most relations, while prioritising inheritance relationships. As such, our proposal would be to start by the classes with the most relations, preferably through the ones with inheriting classes. Another idea that could be interesting to implement, emerged based on these remarks. Our heuristic consists in elaborating a ranking system for the different classes in a diagram. In order to do so, the program would iterate all of the elements that constitute a diagram, attributing different levels of importance to each class in a diagram, based on the weight of several factors, such as: the number of relations that a specific class has; the number of outgoing or incoming inheritance relationships; the number of attributes of that class; the number of methods of that class; etc. Afterwards, the diagram would be read based on the results of the calculated ranking. In a first approach, the weight of these factors could be user defined by what they consider more important in a diagram.

The detailing of a specific class, following once more the detected patterns in the answers given by the participants, could be done by starting with the class type and its name, subsequently describing its variables and methods, and finally moving on to its relationships, together with their respective multiplicities. To make the detailing of relationships more digestible, we suggest separating them into two groups, first describing all outgoing relations, followed by all the incoming relations of that given class.

Another idea that surfaced during the analysis of the answers, was the possibility of an alert sound being played when the tool describes a class that has not yet been mentioned during the diagrammatic reading. For example, suppose the tool was detailing the outgoing relationships of class x, and this class has an outgoing composition to class y, if class y has not been mentioned before, a sound would be played, alerting the users to new information regarding the diagram.

Given that during our observations we found that a few participants had other interesting approaches in the way they started their readings, it could be interesting to give users a choice as well. we would still keep the previously detailed approach as the default since that was the most popular among the respondents. Users could be given the choice of for example, having the tool start the diagrammatic reading of all inheritance relations before everything else, start by the class with the least number of relations, start by listing all classes first and only then, the relations between them, etc. We believe that allowing for user choice would help mitigate future frustrations of using a somewhat complex tool (partly due to its auditory nature), as they could configure the settings more to their liking.

In the responses concerning the readings of both the first and second class diagrams, it was also noted that the subjects divided the diagram into chunks. A similar idea that could be implemented in the tool arose based on the breakdown observed in the second diagram, being inspired by clustering algorithms as well, where we could imagine the data points as the classes in a given diagram. One of the criteria for the division of the diagram into chunks could be, for example, the density of classes in a given "region" of the diagram, or by a certain hierarchy. Specifically, in a given class with inheritance relationships to child classes, we could consider these child classes and the classes with which they have relationships, as part of a separate chunk, with a lower priority.

To perform operations on the diagram, firstly, the guidelines for VUIs mentioned in Chapter 5.2.3 should be consulted, in order to ensure successful user interactions. As described earlier, the subjects started by referring to the classes by their names without exception. As such, mimicking this behaviour in the tool, seems to make the most sense. In an implementation, one could start by speaking the command to be performed: "read"; "insert"; "update" or "remove", along with the UML element, such as "class" or a relationship ("aggregation", "composition", etc.), followed by the property that distinguishes this element. In the case of classes, it will logically be their name. Additionally, during this user interaction, the tool could guide the users throughout it, where the criteria for Guidance and Error Management are of great importance. If the user makes a mistake, the tool should propose the closest command. After the insights provided in the third class diagram, along with the feedback given by the subjects, an idea emerged: in the use case of the program not understanding the association that the users want to remove after their input, the program should be able to label each of the associations (or more generally, the relationships) with a number. Thus, the user receives that feedback, and if they desire to remove association 1, they just have to say "remove association 1", if it is association 2, say "remove association 2", and so forth. Consulting once more the guidelines for VUIs mentioned in Chapter 5.2.3, ensures that the criteria of Immediate Feedback and Error Correction are followed. As the program is giving feedback that allows the users to take corrective actions if needed, while proposing the closest command to what was understood, respectively.

6.5.2 Proposals For State Machine Diagrams

When it comes to the reading of state machine diagrams, the approach could be more direct than what was previously discussed for class diagrams, since as already mentioned, these diagrams have a flow and a certain order. As there is an initial state, it is logical to start there. Through the answers given by the participants during their diagrammatic readings, the states in a diagram could also be ranked by importance. The criteria could be, for example, the number of transitions in a given state and the number of sub-states contained within it. The detailing of the most relevant states could be done at the beginning of the reading, before starting to describe the initial state and so forth.

Concerning the reading of transitions by participants, we identified two approaches. As such, two options could be offered in the tool: The first, is to mention the possible transitions in a given state right after reading that state. The second, is to mention the transitions only after the complete description of the sub-states, more concretely, first the main flow is described, and only afterwards, what isn't included in this main flow is described. We think the former approach would be the appropriate choice to be presented to users by default, as it was the most popular among the respondents. Finally, if any type of activity exists, these should be described as soon as the state that contains them is described.

Regarding the detailing of a given state, the participants started by detailing its name, and described the state and transition that directly preceded it. In essence, providing a "local" context, moving on to then describe the possible outgoing transitions from this state. Thus, these steps appear to offer a possible implementation, as copying this behaviour for the tool would make sense. Additionally, in a situation where the state that we are reading is a sub-state, a description of the possible outgoing transitions of this main state can be offered, thus providing, at the same time, a brief depiction of the "general" context in which the state that we want to be read, is inserted.

6.6 Threats To Validity

Some factors can be viewed as threats to the validity of this study. The fact that this experiment is somewhat long in its duration, taking on average, around 50 to 70 minutes (depending on how detailed the participants were with their explanations), may bias the way people answered the questions. For example, towards the end of the experiment, they might have answered the questions more hastily, not being so attentive anymore, as their concentration diminished. Some people also became somewhat nervous by knowing that their voice was being recorded, which in turn may have influenced the way they answered the questions.

Since this study requires some prior knowledge of UML, this made it necessary to find people with knowledge in this area. Thus, the task of finding people to participate became more complicated, limiting beforehand, the number of participants and the variability within this group.

Although every interviewed person was familiar with the UML notation, some did not remember it that well - hence the need to create a brief summary with the notations for Class and State Machine diagrams in the script. This could have potentially influenced how people read the presented diagrams. For example, starting to read the elements in the diagram where they felt more confident, leaving other elements they did not feel so comfortable to the end of their explanation.

The transcriptions were left as similar as possible to the way the participants described the diagrams, however, they are not an exact match and some compromises had to be made. As such, during this process, some bias may have been added and the results may have become somewhat skewed.

As mentioned, the number of participants was limited. Therefore, many of the detected patterns may not be entirely universal or relatable to all users. To address this shortcoming, further studies need to be conducted.

7

A Catalogue Of Sounds For UML Class Diagrams

Following the previous study, a catalogue of sounds was created for the different elements that constitute a UML diagram, specifically for Class diagrams. This sound catalogue is intended to follow a more methodical approach, where the choices for sounds adhere to the auditory principles proposed in Chapter 5.1, along with the study of the semiotics of the audible field from Chapter 2.1.3, and the design guidelines described in Chapter 5.2, such as the criteria for product-sound design proposed by Blauert [4] and the framework of concepts and heuristics developed by Hug and Misdariis for a scientific method for sound design [46].

Through the elaboration of this Catalogue, in a first instance, we hope to achieve greater efficacy regarding the perception of different sounds on behalf of users when compared to the unsuitable choices of sound symbolism from previous efforts, where the choice of sounds was purely arbitrary and without much reasoning. To draw conclusions regarding these statements, an experimental study was prepared where along with this Catalogue that follows the aforementioned good practices, another catalogue was also defined, where sounds were chosen arbitrarily, disregarding said auditory principles and semiotics.

7.1 Defining The Catalogue Of Sounds

To define our Catalogue, we start from the basis of semiotics, which tells us that a sign is composed of the relationship between the signifier, object and signified. The chosen sound cues are our *signifiers* since they are the representation that our signs will have. Looking at the table that maps the semiotics of the audible field, the sounds in our Catalogue all fall within the Aesthetical Regime, being considered "Soundtracks", as these tend to offer a wide range of articulated sound effects. In this manner, our sound cues can be considered auxiliary legisigns that are being replicated to collaborate with signifying processes mainly conveyed through the visual form, such as the case with the visual nature of UML diagrams. Our *objects* are the UML elements in question, which are the concept each of the chosen sounds will represent. Finally, the *signified* is the definition of a given UML concept.

As discussed before, through the definition of the three trichotomies defined by Peirce, the second trichotomy emphasises the different ways in which a signifier can refer to its object. As described by Blauert and mentioned in Chapter 2.1.3, sound cues can be separated into three categories of abstraction, where each of these relates to the aforementioned second trichotomy. Taking this into consideration, a sound cue and its relation to the UML element it represents can be considered an Index, an Icon or a Symbol. Figure 7.1 is based on figure 2.1 and illustrates, in a concise manner, what has been described, denoting the relationships between the different elements that constitute a sign in this specific context. It should also be mentioned that the form in which the signified denotes the object (the third trichotomy) will not be explored in this document, as the definition of a UML concept is already well established within UML documentation.

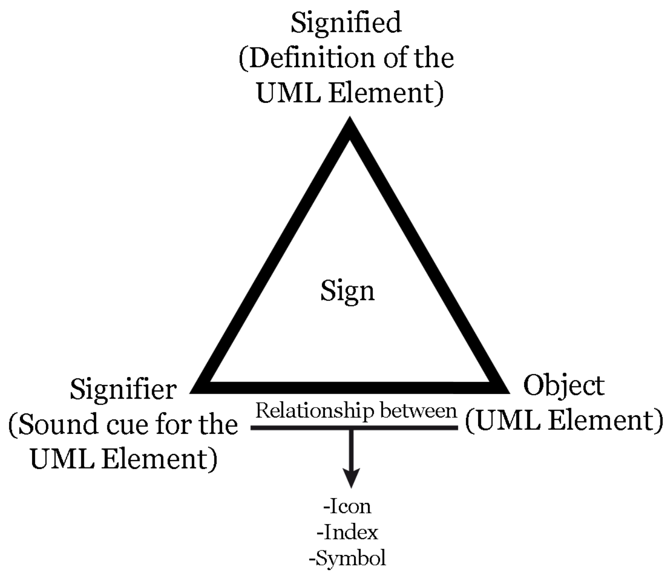


Figure 7.1: Semiotic Triangle For The Catalogue Of Sounds

7.2 Specifying The Catalogue Of Sounds

This section defines our Catalogue of sounds specific to UML class diagrams. As previously mentioned, this Catalogue follows the auditory principles proposed in this document. More precisely, it follows the principles of semiotic clarity and semantic transparency principles, along with perceptual discriminability, auditory expressiveness and economy. As this Catalogue only focuses on defining sounds for the basic elements that form a UML class diagram, the principles of cognitive integration and dual coding will not be applied here, as these concepts imply more complex relationships between the different elements in a UML diagram, along with the need for the usage of a TTS voice. For the case of complexity management and cognitive fit, unfortunately, it is not a simple matter to test these principles in the context of this study. These would have to be experimented through languages other than UML and that would be beyond the scope of this document.

All sounds apply the principle of auditory expressiveness since each one uses the full range and capacities of auditory variables. For example, the pitch is used to represent this relationship between classes in the inheritance sound. The principle of auditory expressiveness is also completely applied, as we use the various characteristics that make up a sound. Additionally, all the sounds created for the experimental study and subsequent use in the developed tool, took into account the guidelines detailed in Chapter 5.2. These sounds all have a maximum of 3 seconds, being considered brief auditory icons (normally called "earcons"), to be more easily recognisable by users. Furthermore, it should be noted that the chosen sounds also try to accommodate the universal human experience to be as understandable as possible, as there may be certain sounds that people from distinct cultures and backgrounds will comprehend in different forms.

The table 7.1 further expands on these concepts by featuring each diagram element, its definition, the defined sound and the reasoning behind the choice.

Table 7.1: Catalogue Of Sounds for UML Class Diagram Elements

Diagram Element	**Definition**	**Sound**	**Explanation**
Class	A class represents an object or a set of objects that share a common structure and behaviour. They're represented by a rectangle that includes rows of the class name, attributes, and operations.	Book opening + positive/-correct sound	Opening a book signifies the opening of something that contains more information, namely regarding attributes and operations. Correct sound is a high-pitched sound, so it sounds "bright". The note becoming higher pitched indicates to the user that something is happening and that this is an important sound, leaving the user more attentive to something that could have more information. The fact that it is a sound that goes higher in pitch is also normally associated with a "positive" sound.

Attribute	Attributes represent the information, data, or properties that belong to instances of a class	Wooden bricks falling + computer notification sound (beeps)	The sound of wooden bricks falling symbolises the building blocks that will be used to build something, in this case, the attributes of a class can be seen as the building blocks of a class. The sound of the notification, through its beeps and sequence, is a sound that points towards something "digital", which represents information and data. The combination of these two sounds points the users to the notion of the attribute.
Operation / Method	Operation is a behavioural feature that may be owned by an interface or class	Keyboard Typing	An operation is linked to the act of programming, so the typing sound symbolises the act of programming the specification of a method
Association	A relationship between two or more classes that involve connections among their instances	Arrow shot	Symbolises a relationship between two or more classes. We can associate the sound of an arrow being shot to a cupid, which shoots arrows. Generally, the cupid symbolises an association/relationship between people. Furthermore, an arrow is also used to represent this concept in UML diagrams visually, so it makes sense to have this equivalence for sound

Inheritance	"is-a" relationship The process of a child or sub-class taking on the functionality of a parent or superclass, also known as generalisation. It's symbolised with a straight connected line with a closed arrowhead pointing towards the superclass.	Sound of a book opening (lower pitched) + Coins falling sound + sound of a book opening (higher pitched)	The sound of the book opening is a motif that points to the sound used for classes. The higher and lower pitched sound is symbolic of the human voice's higher and lower pitched voice, which in turn is related to the sound of a parent and child, a concept related to Inheritance. The usage of the pitch in this sound also respects the principle of auditory expressiveness. Finally, the sound of falling coins tries to make more evident that this is the concept of Inheritance, which is usually associated with wealth and, therefore, money. These three sounds combined, represent the "passing down" of Inheritance (represented by the sound of coins), from the lower-pitched sound for a class, to the higher pitched sound for a class
Realisation / Implementation	Implementation of a specification	Construction noise in the background with a hammering sound standing out	The ambience of construction noise in the background already takes our minds to a place where something is being built. Having the hammer sound stand out makes it a little clearer what is being depicted. As mentioned, these sounds represent something being built/implemented, which is what is intended to be represented with this concept.

Dependency	Directed relationship, a change to one modelling element (the independent element) will affect the other modelling element (the dependent element)	Baby crying	The sound of a baby crying symbolises the infant's dependency on its parents. This sound follows the principle of semantic transparency since the auditory representation suggests the meaning intended for this concept.
Aggregation	An aggregation is a special type of association in which objects are assembled or configured to create a more complex object. An aggregation describes a group of objects and how you interact with them. "has-a" or whole/part relationship (weak, destroying whole does not destroy parts)	Sports crowd	Crowd noise symbolises an aggregation of people, which can be associated to the concept of aggregation, in which objects are assembled together to create a more complex object. This particular sound points the listener to a crowd at a sports game, where this concept can also be associated with a team, which is composed of players. If a team is disbanded, the players live on. This, in turn, meets the concept represented in aggregation, which states that destroying the whole does not eliminate the parts.
Composition	A composition relationship specifies that the part class's lifetime depends on the whole class's lifetime. Strong form of aggregation (parts are destroyed along with the whole)	Sports crowd + fire burning	Just like aggregation, the crowd noise symbolises a gathering of people, which turns the users to the concept of aggregation. Seeing that composition is a strong form of aggregation, it makes sense to have this sound as a motif that reminds the users of this concept. However, the principle of semiotic clarity tells us that each sound should only represent one concept, so to further differentiate these two concepts, the composition has a fire-burning sound. This is a destructive sound, which indicates that in this concept, the parts are destroyed along with the whole.

Class Association	An association class is a class that is part of an association relationship between two other classes.	Arrow sound + Book opening + pages in a book sound	This sound associates the concept of the association relationship and the concept of class. For this matter, it seems appropriate to reuse the already-defined sounds. The arrow sound symbolises the association, joined with the defined sound for the concept of class (book opening) and the new sound of the pages inside the said book. This new sound indicates to the user that this is a new sound. In this way, every sound is distinct for each class diagram element, obeying the principle of perceptual discriminability, but the user is reminded of the concepts already established before (semiotic clarity is respected, seeing that these sounds only represent one concept)
Package	Organise related classes in a diagram. They are symbolised with a large tabbed rectangle shape.	Envelope being opened + zip opening	Follows the same idea as the sound is chosen for the class concept. The envelope being opened ties into the sound used for the concept of a class (book opening) through the usage of paper. An envelope is not only a piece of paper, but it also usually contains written information within. Furthermore, an envelope can symbolise mail (visual icons for this concept usually represent it through a closed envelope) and can be associated with delivering "packages". The sound of the zip of a bag being opened represents something that contains more information within it while also being reminiscent of a mailman's bag.

7.3 Experimental Study

This section presents the experimental procedure regarding the Catalogue of sounds for UML Class Diagrams.

7.3.1 Planning

Before the experiment was put into practice, it was necessary to plan all the details leading up to it. The experimental study has been structured in the phases presented in the following sub-chapters.

7.3.1.1 Goals

As briefly stated before, a study was conducted to qualitatively determine if the construction of a catalogue of sounds following a more methodical approach based on the proposed auditory principles and the semiotics of the audible field is more adequate and effective when compared to a catalogue that disregards said principles and semiotics, where the choice for sounds is purely arbitrary. Additionally, we want to determine how relevant the previously proposed auditory principles are to the users.

7.3.1.2 Participants

To effectively carry out the experimental study, the subjects needed to have a certain level of understanding in the modelling domain, specifically UML class diagrams. Furthermore, it was required that participants had a basic knowledge of English, since the questions were asked in English and it could be necessary to explain their reasoning, if they so desired. They also needed to have a device capable of sound output, to be able to select the most appropriate sound. They were additionally required to have a working and reasonably reliable internet connection, as this evaluation was carried out using an online form, being necessary to download the aforementioned sounds as well. The participants were chosen through convenience sampling, as we contacted the people who were available, with these being computer engineers, and master and undergraduate students in computer engineering.

To better understand the profile of the participants and their level of knowledge, the following questions were asked in the form, thus allowing us to get a more comprehensive picture of who answered the questions:

- Gender
- Are you hard of hearing?
- How do you consider your level of understanding of UML?
- When was your last contact with UML?
- In what context do you use UML?

The results of these questions will be displayed in Chapter 7.3.3.1. In total, this experimental study had thirty one participants.

7.3.1.3 Tasks

Participants had to answer a form where they were presented with the elements of a UML class diagram and their visual representation. For each of these concepts, users were then asked to listen to two sounds: the sound previously defined in the Catalogue that follows good practices and a sound from a baseline Catalogue that does not satisfy the auditory principles. Subsequently, the respondents were asked to select which sound they thought best suited the concept being presented to them, being able to explain their choice by text if they so desired. If the participant didn't think any of the two sounds matched the concept, they could also suggest their own and explain it.

After this task, participants had to answer a few additional questions where they evaluated the relevance of the previously proposed auditory principles.

7.3.1.4 Defining The Hypotheses

Next, the hypotheses regarding the study in question are presented. The first set of hypotheses concerns the use and effectiveness of the catalogues of sounds from the user's point of view. In contrast, the second set refers to the relevance of the auditory principles in concrete. The main objective of this study is to understand if there are differences from the user's point of view regarding the use of a catalogue of sounds that follows the elaborated auditory principles and semiotics of the audible field compared to an arbitrary sound catalogue. Secondarily, it aims to understand the relevance of the proposed auditory principles to the users. Consequently, this study seeks to reject the null hypotheses.

The null hypothesis ($\mathbf{H_0}$) states that there are no differences, both in the usage of a catalogue of sounds that follows the elaborated principles versus an arbitrary sound catalogue and that the proposed auditory principles are not relevant. The alternative hypotheses ($\mathbf{H_1}$) state that there are differences, that the auditory principles are relevant, and that the null hypothesis is rejected.

Defining the first set of hypotheses:

$\mathbf{H_{0CatalogueT}}$: There are no differences in using a catalogue of sounds that follows the elaborated auditory principles and semiotics of the audible field, *versus* an arbitrary catalogue of sounds, from the users' point of view.

$\mathbf{H_{1CatalogueT}}$: There are differences in using a catalogue of sounds that follows the elaborated auditory principles and semiotics of the audible field, *versus* an arbitrary catalogue of sounds, from the users' point of view.

Defining the second set of hypotheses:

$\mathbf{H_{0PrinciplesT}}$: The elaborated auditory principles, together with the semiotics of the audible field, are not relevant from the users' point of view.

$\mathbf{H_{1PrinciplesT}}$: The elaborated auditory principles, together with the semiotics of the audible field, are relevant from the users' point of view.

7.3.1.5 Preparing The Evaluation Sessions

For the evaluation sessions to run as expected, it was necessary to plan what was required. After formulating the sound Catalogue mentioned above, it was also necessary to define a catalogue which did not meet the auditory principles. With these two catalogues specified, it was required to create the sounds themselves, which would subsequently have to be presented to the participants during the experiment. For this purpose, the basic sounds used in this study were taken from the website *zapsplat.com* - which provides sounds for free use - and then edited/truncated with the software *REAPER*, a digital audio workstation, which allows for sound editing, to create the sounds we intended. Figure 7.2 presents a typical view of how the sounds were created in *REAPER*, together with a screenshot of the homepage of *zapsplat.com*. All the sounds created for this study were uploaded and can be consulted in a repository at Zenodo[1].

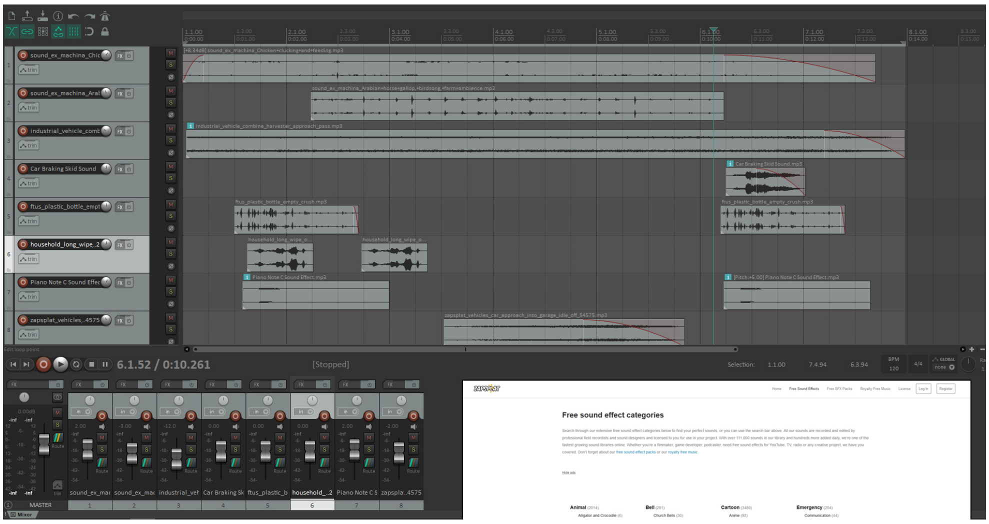


Figure 7.2: Typical view for how the sounds were created in *REAPER*, along with the homepage of the website used to download the free basic sounds.

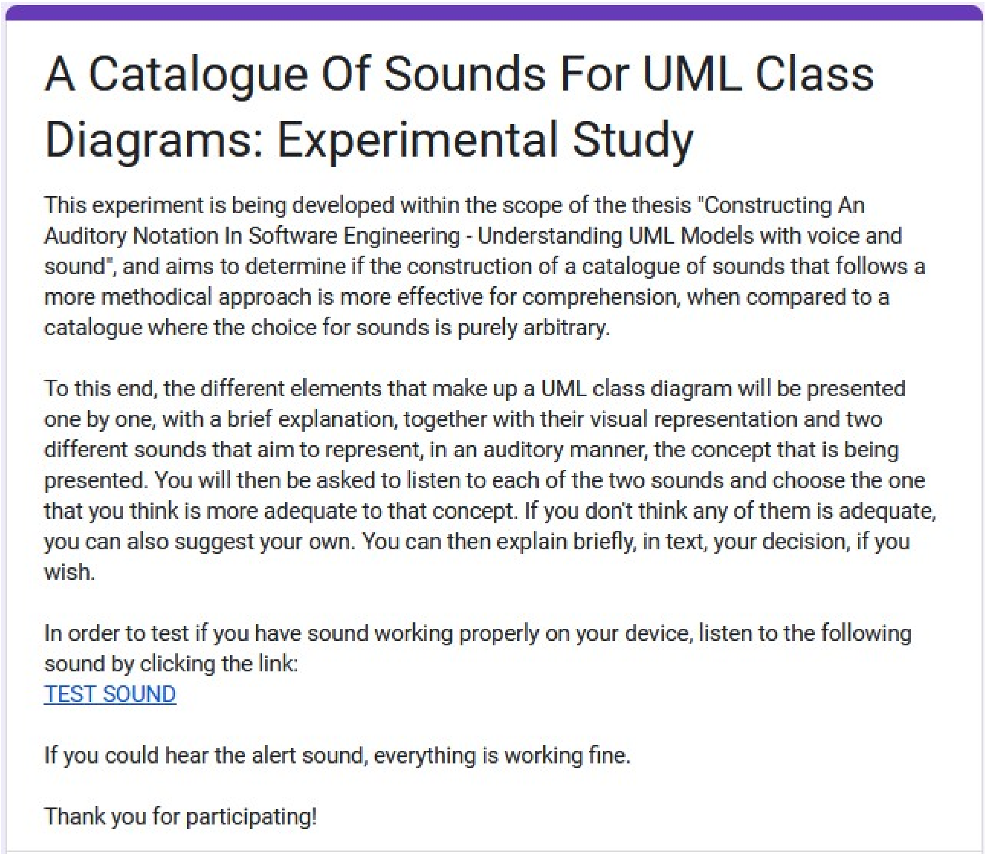
Concurrently, the form was created using Google Forms, where each question focused on only one of the elements that constitute UML Class diagrams, along with its visual representation and a brief definition of the concept. After creating the sounds with the tools mentioned above, each of these questions was added to the form. Since Google Forms does not support audio playback natively, it was necessary to find a workaround so that the sounds would be presented. The sounds were then uploaded to google drive, and the links to the sounds were embedded in the form. It was necessary to make these links accessible by pressing the "Share" button and, in "General Access", selecting the option to "any person with the link " allowing any person with the link to listen to the sound.

Since it was necessary to utilise this workaround and embed the links in the form, the sounds couldn't be presented to the participants in a random manner, so we opted

[1]A Catalogue Of Sounds For UML Class Diagrams - Experimental Study: `https://doi.org/10.5281/zenodo.7766679`

to do something "semi-random". In this way, the sounds for each question were named "Sound A" and "Sound B". The "Sound A" and "Sound B" do not always correspond to the catalogue of good practices or the catalogue of bad practices. This means that, for example, in question 1, the sound corresponding to the catalogue of good practices is represented as "Sound A" and "Sound B" refers to the other catalogue, but in question 2, the sound corresponding to the Catalogue of good practices is represented as "Sound B" and the sound from the other catalogue as "Sound A". Thus, the answer for each question doesn't become predictable because the same letter will not always correspond to the same catalogue, as such, avoiding that participants answer for all questions "Sound A" or "Sound B"as well. It was also decided that the sounds would only be named A or B, without any further description, to give the participants as little context as possible so they could deduce the sounds they were hearing. It was decided that for each of the questions, the sound belonging to the catalogue of good practices would correspond to the following letter: 01-A; 02-B; 03-B; 04-A; 05-B; 06-A; 07-A; 08-B; 09-A; 10-B; 11-A.

After the conception of these questions, a section was created on the form, where the table defined in 5.1 was presented, and participants were asked to define, on a scale of 1 to 5, how important they considered each of the proposed principles to be, along with a question about whether the principles are overall relevant.



A Catalogue Of Sounds For UML Class Diagrams: Experimental Study

This experiment is being developed within the scope of the thesis "Constructing An Auditory Notation In Software Engineering - Understanding UML Models with voice and sound", and aims to determine if the construction of a catalogue of sounds that follows a more methodical approach is more effective for comprehension, when compared to a catalogue where the choice for sounds is purely arbitrary.

To this end, the different elements that make up a UML class diagram will be presented one by one, with a brief explanation, together with their visual representation and two different sounds that aim to represent, in an auditory manner, the concept that is being presented. You will then be asked to listen to each of the two sounds and choose the one that you think is more adequate to that concept. If you don't think any of them is adequate, you can also suggest your own. You can then explain briefly, in text, your decision, if you wish.

In order to test if you have sound working properly on your device, listen to the following sound by clicking the link:
TEST SOUND

If you could hear the alert sound, everything is working fine.

Thank you for participating!

Figure 7.3: The form's introductory text, presented to the participants.

Still, in Google forms, although the answers were anonymous, it was also decided to only allow people to answer when logged in to their email to avoid multiple submissions by the same person.

Finally, after creating both catalogues, the sounds themselves and the form, it was possible to carry out a pilot test with a junior project development team member. This allowed the details of the experiment to be well outlined, in addition to obtaining an estimate of how long it would take the participants to complete the experiment. This pilot experiment took 40 minutes.

The introduction presented to the participants is reproduced in figure 7.3. The complete form elaborated for this experimental study can be consulted in Appendix B.

7.3.1.6 Experimental Material

The evaluation sessions took place online, and no restrictions were placed on what type of device should be used to carry out the evaluation, as it could be done through any device at the preference of the participants. However, subjects needed to have a device capable of producing sound output, such as speakers or any type of headphones, so that they could listen to the sounds. Logically, it was also necessary to use a monitor or some type of visual display, to visualise the form and answer it. Additionally, the subjects needed to have a reasonable internet connection.

On our behalf, a personal computer was used, with the software *REAPER* - a digital audio workstation, which allows for sound editing, to create the sounds we intended - installed, headphones, and a good internet connection to be able to download any basic sounds needed, to create the form using Google Forms and to upload the created sounds.

7.3.1.7 Specifying The Baseline Catalogue With Unsatisfied Auditory Principles

As previously mentioned, it was first necessary to construct this Catalogue of "bad practices", where it was only sought that the chosen sounds violated the auditory principles and semiotics, both previously detailed. This means that apart from these requirements and reasoning, all sounds were arbitrarily decided. We attributed a sound for each concept in UML class diagrams (just like what was done for the previously detailed Catalogue that follows "good practices"), along with an explanation of how that particular sound does not satisfy the proposed auditory principles. Table 7.2 presents this Catalogue, where a sound was defined, along with the list of unsatisfied auditory principles, for each diagram element.

Describing in detail the choices for this Catalogue and how they do not satisfy the proposed auditory principles, it was decided for the elements Class and Attribute, the sound of a car engine. In these two elements, the principle of Semiotic Clarity is violated since the same sound represents different concepts (something that didn't happen in the previous Catalogue, where if the same sound was used, it always represented the same concept). Furthermore, the principle of Semantic Transparency is not satisfied since the

Table 7.2: The Catalogue Of Bad Practices with the Unsatisfied Auditory Principles

Diagram Element	**Sound**	**Unsatisfied Auditory Principles**
Class	Sound of a car engine	Semiotic Clarity (along with Attribute); Semantic Transparency
Attribute	Sound of a car engine	Semiotic Clarity (along with Class); Semantic Transparency
Operation / Method	Sound of running water	Semiotic Clarity (along with association); Perceptual Discriminability (along with Association); Semantic Transparency
Association	(A different) sound of running water	Semiotic Clarity (along with Operation/Method); Perceptual Discriminability (along with Operation/Method); Semantic Transparency
Inheritance	Farm animals + piano notes + window being cleaned + tyres breaking + plastic bottle being crushed + the sound of a car engine	Semiotic Clarity; Perceptual Discriminability; Semantic Transparency; Auditory Economy
Realization / Implementation	Sound of the wind	Semiotic Clarity (Along with Dependency); Semantic Transparency
Dependency	(A different) sound for the wind	Semiotic Clarity (Along with Realization/Implementation); Semantic Transparency
Aggregation	Elephant Sound	Semantic Transparency
Composition	Cartoon running Sound	Semantic Transparency
Class Association	Doorbell Sound	Semantic Transparency
Package	Explosion Sound	Semantic Transparency

sound of a car engine does not suggest the meaning of these concepts. The signifier does not resemble the signified and is therefore considered a Symbol.

Advancing down the Catalogue, for the elements Operation/method and Association, it was decided the sound of running water would be used, with the particularity that in practice, these are represented by two different sounds. This way, the principle of Perceptual Discriminability is not satisfied because two different sounds are very similar and not distinguishable from one another. In the case of the Class and Attribute elements, this principle was not disobeyed because the exact same sound was used for these two concepts instead of two different sounds. Semiotic clarity is also considered unsatisfied because although, in practice, the sounds are different from each other, the same symbol (since the connection between the sound of water - the signifier, and its signified is arbitrary) is being used for two different concepts. Finally, these two elements also do not respect the principle of Semantic Transparency because running water does not suggest the meaning of any of these concepts.

For the concept of Inheritance, the idea was to have a significant quantity of sounds playing simultaneously. As such, the sound is a mixture of several sounds. These are: farm animals; piano notes; window being cleaned; tyres breaking; plastic bottle being crushed, and the sound of a car engine. This association of sounds doesn't satisfy the principle of Auditory Economy, since the number of auditory symbols ceases to be cognitively manageable. This sound also violates the principle of Semantic Transparency by not suggesting the meaning of the concept, along with Perceptual discriminability, as there is a sound of a different car engine here, it is no longer distinguishable from the class and attribute sound; Semiotic clarity is not satisfied as well since, despite the sound of the car being different from the others, it is the same symbol for a different concept. To finalise, it should be noted that this sound is too long. While this is not a violated principle, it goes against the guidelines explored in Chapter 5.2: "use sonification to summarise information".

For the elements Realisation/implementation and dependency, two different sounds of wind were chosen. This means that these do not satisfy the principle of Semantic Transparency since the wind does not suggest the meaning of any of these concepts. The principles of Semiotic Clarity and Perceptual Discriminability were also disobeyed for the same reasons as seen before with the elements Operation and Association, meaning that the same symbol is used for two different concepts, and two distinct sounds are too similar and not distinguishable, respectively.

Lastly, for the elements aggregation, composition, class association and package, sounds that violate the principle of Semantic Transparency were chosen. Furthermore, it should be noted that in addition to the principles that were not followed in this Catalogue, the sounds subsequently created to represent what is detailed here do not follow the guidelines explored in Chapter 5.2, with examples such as: "volume mixing inconsistent and random" and "sound which contradicts one another in tone and style". The junction of cartoon sounds with real life sounds exemplifies the latter. For example, the sound of a car (used to represent a class and attribute) and the cartoon sound used for the concept of composition, or the variety of sounds used for inheritance, makes this catalogue vary in tone and style dramatically.

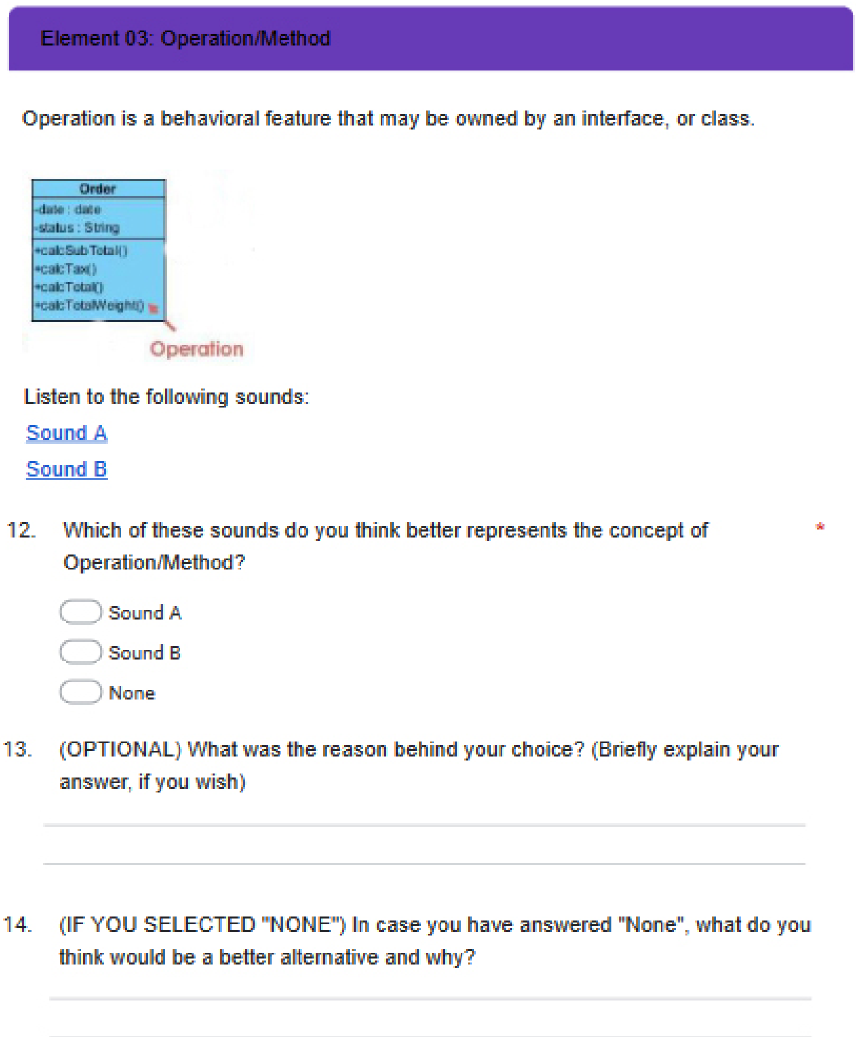
7.3.2 Procedure Of The Evaluation Sessions

The various evaluation sessions took place online. Each session had a single participant, and they were sent the link to the form. Initially, a short explanation is presented, briefly outlining the purposes of the study to give some context to the participants while briefly describing how the experiment will proceed. Since this study requires a device capable of emitting audio, a test sound is also shown so that users can test their setup. Once they hear the sound, the participants are ready to start the experiment and answer the questions.

Firstly, some personal questions were asked, including their level of understanding of

UML and if they were hard of hearing. This allowed us to obtain better profiling of the participants.

Figure 7.4 illustrates how a typical set of questions is presented to the participants. On this specific question, the UML element in focus is accompanied by its visual representation and a brief definition of the concept. Afterwards, the participants click on the hyperlinks for the two sounds, in order to listen to them. Each of these links takes the participants to its respective Google Drive sound file, which allows them to listen to that specific sound, through the embedded audio player. Next, after listening to both sounds, the participants answer what sound they think better represents the UML concept of that given question, where they can explain their answer if they desire as well. If the participants are of the opinion that neither of the sounds is appropriate, they can also recommend a better alternative and explain it.


Element 03: Operation/Method

Operation is a behavioral feature that may be owned by an interface, or class.

Listen to the following sounds:

Sound A

Sound B

12. Which of these sounds do you think better represents the concept of Operation/Method? *

- Sound A
- Sound B
- None

13. (OPTIONAL) What was the reason behind your choice? (Briefly explain your answer, if you wish)

14. (IF YOU SELECTED "NONE") In case you have answered "None", what do you think would be a better alternative and why?

Figure 7.4: Structure of a typical set of questions in the form.

Finally, participants were shown table 5.1 after answering these questions. On a scale of 1 to 5, they were asked to define how important they considered each of the proposed principles to be, along with a question about their opinion regarding the overall relevance of the proposed principles. This allows us to make a better global assessment of how participants answered the questions *versus* how they perceived the relevance of the auditory principles. This experimental study took on average, around 30 minutes.

7.3.3 Results

After the evaluation sessions, we obtained the results presented below. This section is divided into three: Personal Information, The Catalogue Of Sounds and Principles For Constructing Auditory Notations. The first concerns the personal information of the participants. In the second, their choices regarding the concretely proposed sounds are presented. The latter provides the assessment of the subjects about our proposed principles for constructing auditory notations. The individual results, along with the sounds constructed for this experimental study can be consulted in a repository at Zenodo[2].

7.3.3.1 Personal Information

In this subsection, the personal information of the subjects is presented, allowing us to obtain a better profile of the people who answered the questions. 31 participants responded to the form, of which 16.1% were female and 83.9% were male.

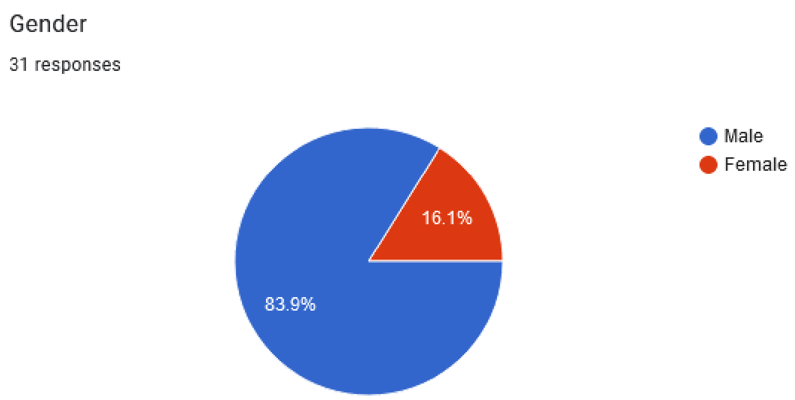


Figure 7.5: Results regarding the participants' gender.

[2]A Catalogue Of Sounds For UML Class Diagrams - Experimental Study: https://doi.org/10.5281/zenodo.7766679

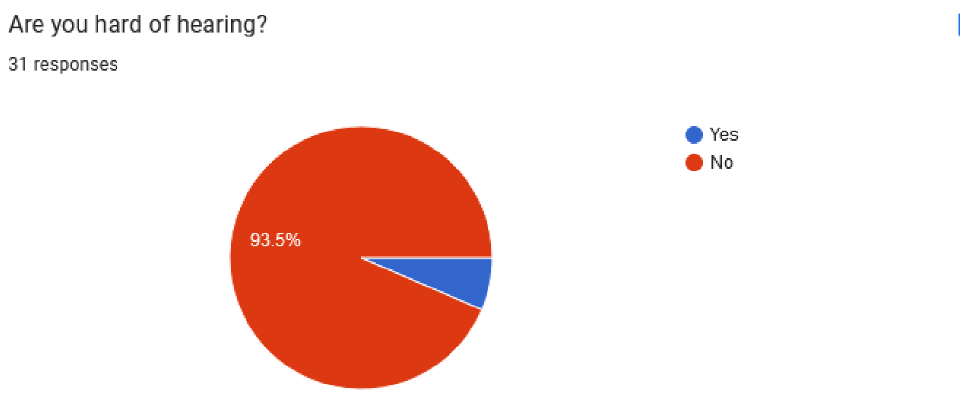


Figure 7.6: Results regarding the participants' hearing condition.

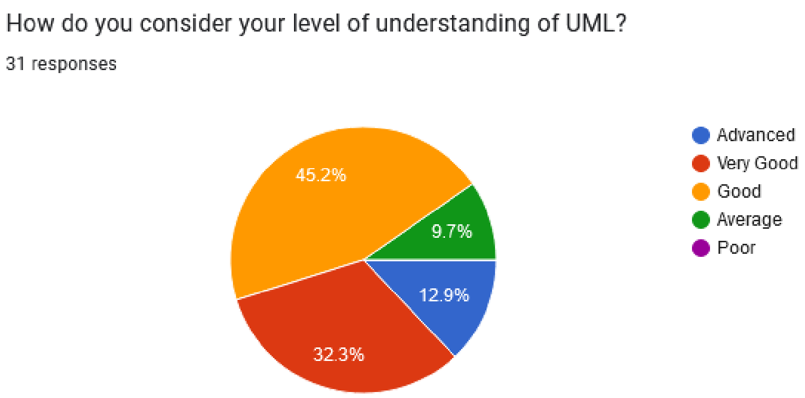


Figure 7.7: Results regarding the participants' level of understanding of UML.

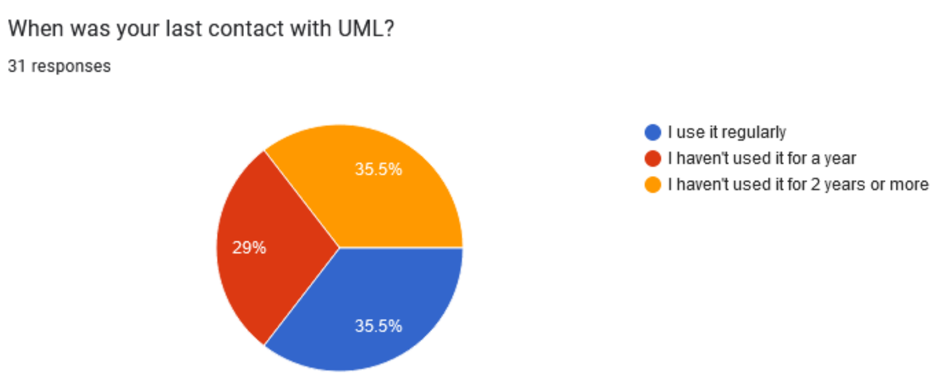


Figure 7.8: Results regarding the participants' last contact with UML.

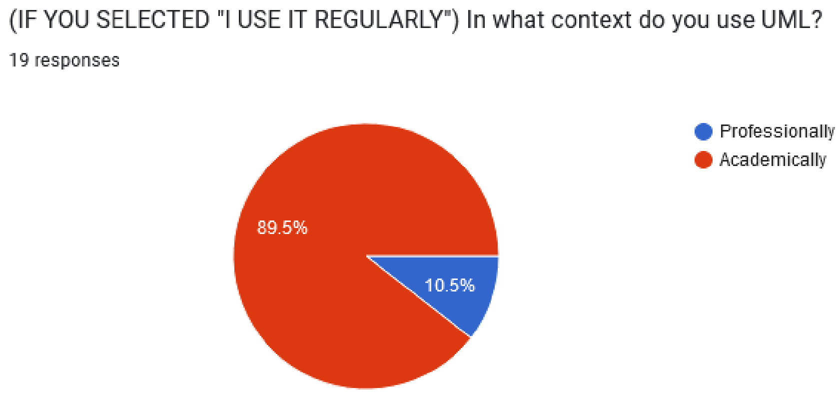


Figure 7.9: Results regarding the participants' context of usage of UML.

7.3.3.2 The Catalogue Of Sounds

The following subsection will present the collected data related to the proposed sounds, as chosen by the participants. The sounds were detailed in the aforementioned Catalogues Of Sounds, in tables 7.1 and 7.2.

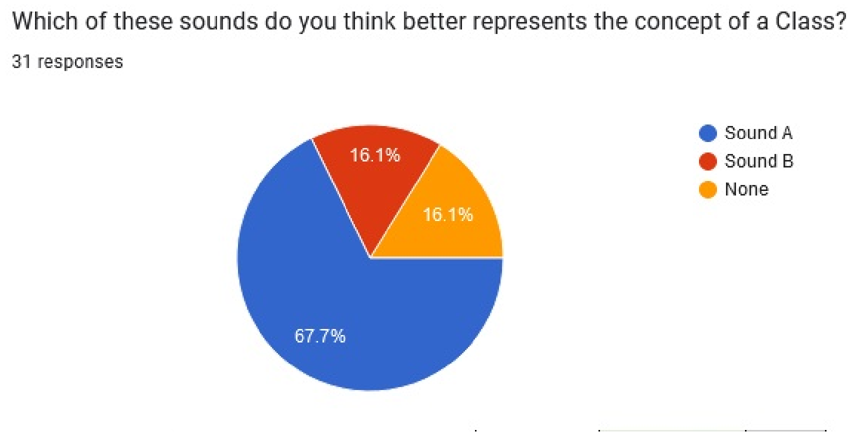


Figure 7.10: Results of the first question.

In the first question, regarding the diagram element Class, the sound constructed through the catalogue of sounds based on the proposed auditory principles and the semiotics of the audible field corresponds to the letter A, which is the sound of a book opening together with a positive/correct notification sound. Sound B is the sound of a car engine, as detailed in the catalogue with unsatisfied auditory principles. 67.7% of participants chose Sound A, while Sound B and None had the same percentage, with 16.1% each.

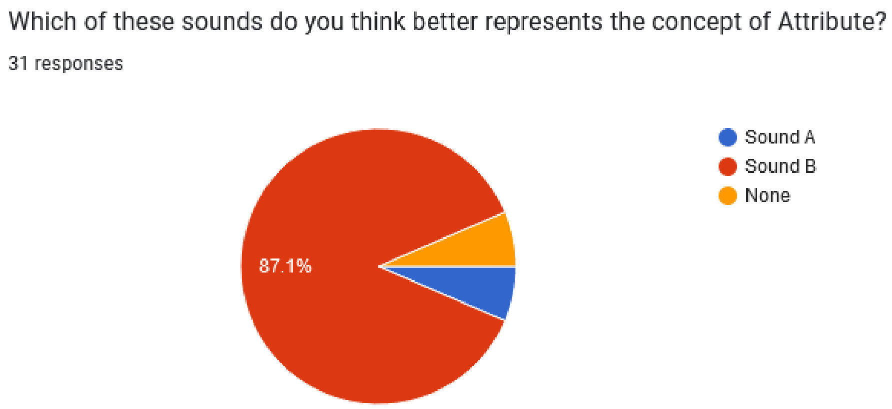


Figure 7.11: Results of the second question.

The second question concerns the diagram element attribute. The sound proposed through the catalogue of good practices corresponds to the sound of falling wooden bricks joined with a computer notification sound, and is represented in Sound B. Option A is the sound of a car engine, as aforementioned in the catalogue with unsatisfied auditory principles. 6.45% of participants chose Sound A, while Sound B had a percentage of 87.10%. The option None obtained 6.45%.

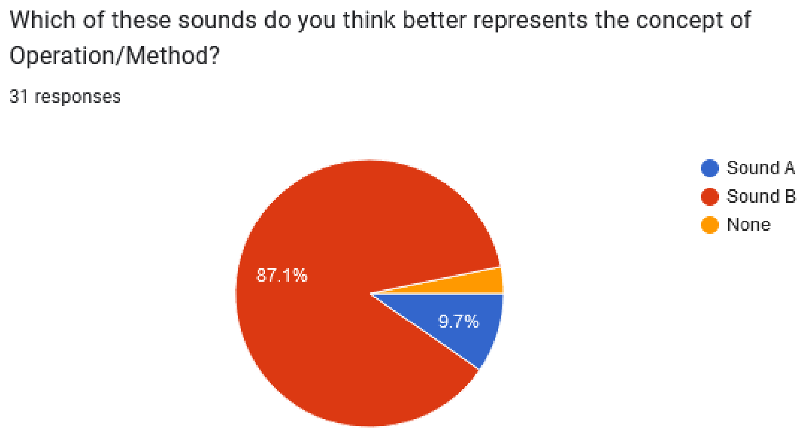


Figure 7.12: Results of the third question.

Operation/Method is the diagram element in focus on the third question. Sound A had a percentage of 9.68% and is the sound of running water, corresponding to the catalogue with unsatisfied principles. Sound B, which is the sound of typing on a keyboard - as described in the catalogue of good practices - obtained 87.10%, and the None option, 3.23%.

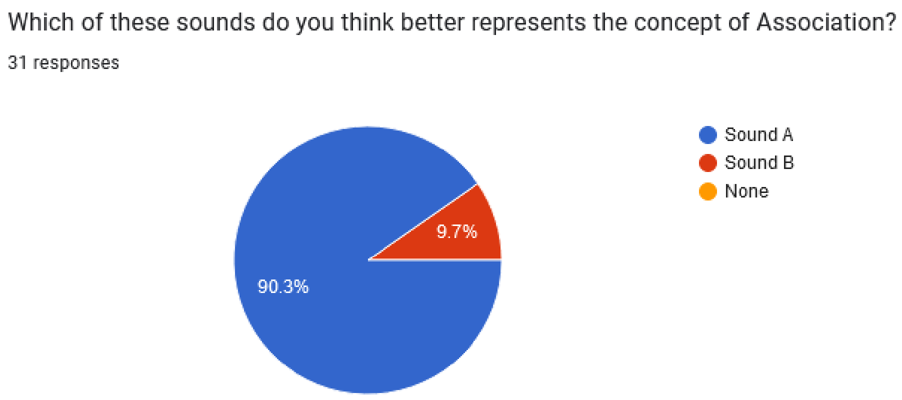


Figure 7.13: Results of the fourth question.

The fourth question is related to the diagram element association. Sound A, with a percentage of 90.32%, corresponds to our proposed catalogue of good practices and is the sound of an arrow being shot. Sound B, with 9.68%, corresponds to a different sound of running water in relation to the previous question, as described in the catalogue of unsatisfied principles. No subjects selected the option None for this question.

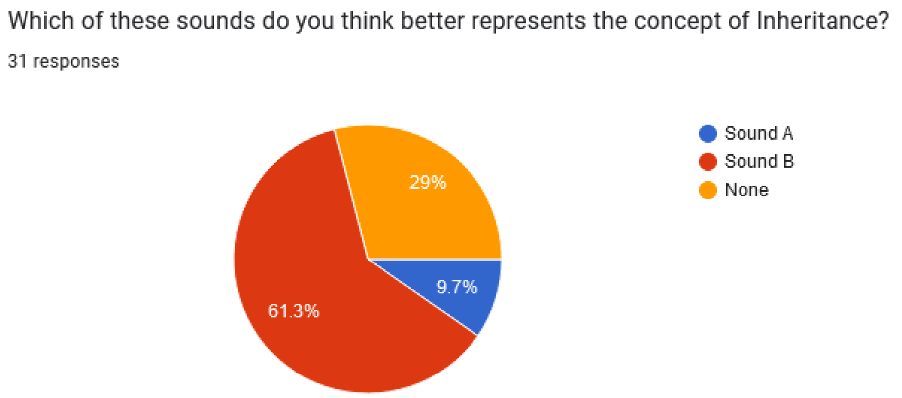


Figure 7.14: Results of the fifth question.

Inheritance is the diagram element of question five. Sound A corresponds to the catalogue of unsatisfied auditory principles and is the junction of farm animals with piano notes, a window being cleaned, tyres breaking, a plastic bottle being crushed and the sound of a car engine. This sound obtained a percentage of 9.68%. Sound B belongs to the catalogue of good practices and is composed of a lower-pitched sound of a book opening, followed by coins falling and a higher pitched sound of a book opening, achieving the result of 61.29%. The None option had a percentage of 29.03%.

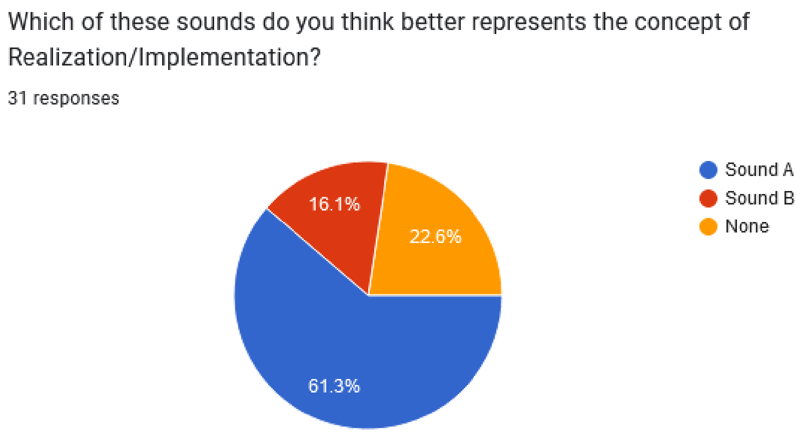


Figure 7.15: Results of the sixth question.

The sixth question concerns the diagram element realisation/implementation. Option A obtained a percentage of 61.29%, and represents the sound proposed in the catalogue of good practices. Specifically, it is the sound of a hammer with construction noise in the background. Option B is the sound of the wind, as detailed in the catalogue with unsatisfied auditory principles, and was chosen by 16.13% of participants. 22.58% was the result for the option None.

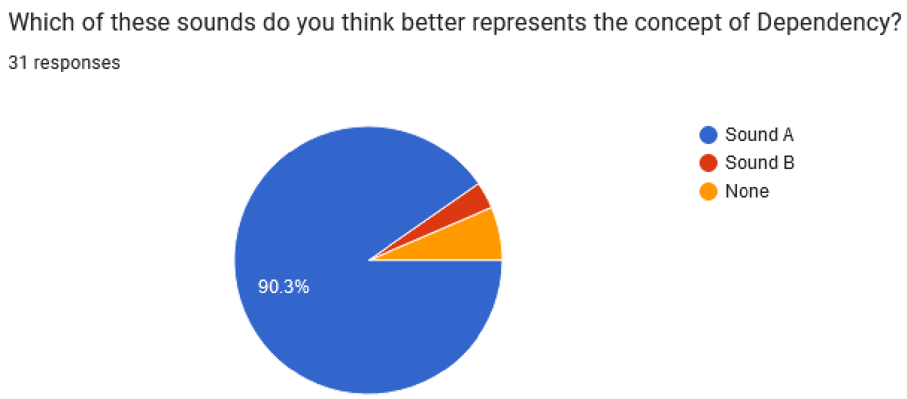


Figure 7.16: Results of the seventh question.

In the seventh question regarding the diagram element dependency, the sound constructed through the catalogue of sounds based on the proposed auditory principles and the semiotics of the audible field corresponds to the letter A, which is the sound of a baby crying. Sound B is a different sound for the wind, as described in the catalogue of unsatisfied auditory principles. 90.32% of participants chose Sound A, while Sound B obtained the figure of 3.23%. Option None was selected by 6.45% of subjects.

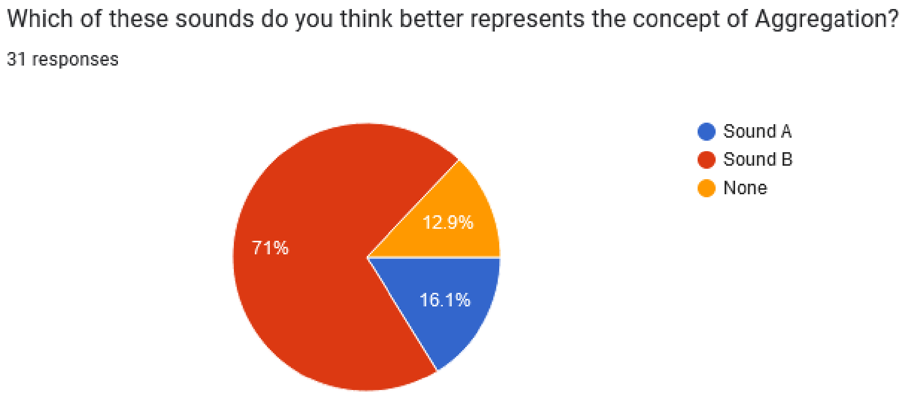


Figure 7.17: Results of the eighth question.

Aggregation is the diagram element in focus on the eighth question. Option A is the sound of an elephant, and corresponds to the catalogue with unsatisfied principles, with a percentage of 16.13%. Option B, which is the sound of a sports crowd - as detailed in the catalogue of good practices - achieved 70.97% and the None option, 12.90%.

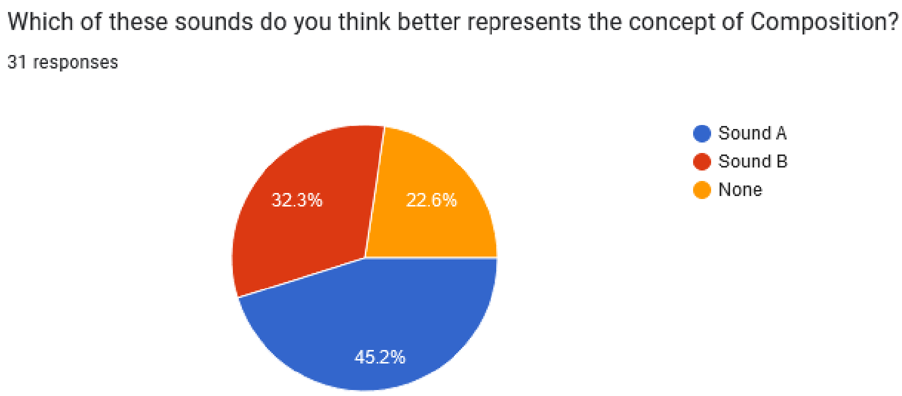


Figure 7.18: Results of the ninth question.

Composition is the diagram element of question nine. Sound A belongs to the catalogue of good practices and is composed of a sports crowd, together with the sound of burning fire, obtaining a result of 45.16%. Sound B corresponds to the catalogue of unsatisfied auditory principles and is the sound of a cartoon running. This sound obtained a percentage of 32.26%. The None option had a percentage of 22.58%.

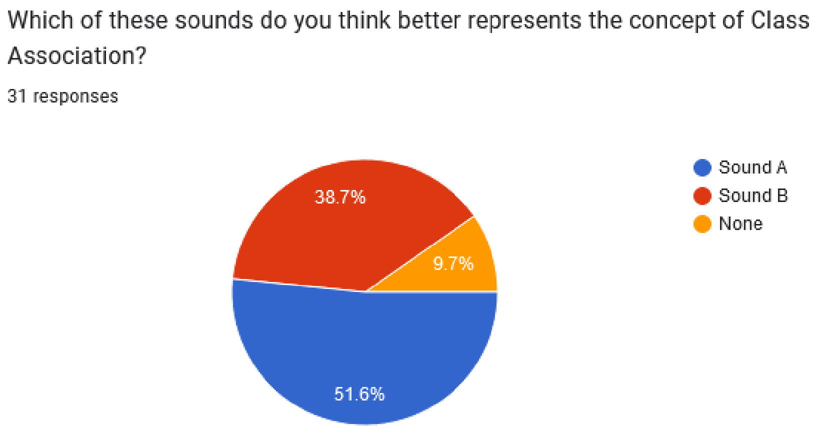


Figure 7.19: Results of the tenth question.

In the tenth question, regarding the diagram element class association, sound A corresponds to the catalogue of unsatisfied auditory principles, and is the sound of a doorbell. The sound constructed through the catalogue of good practices corresponds to the letter B, which is the junction of an arrow being shot with the sound of a book opening, together with its pages being flipped. 51.61% of participants chose Sound A, while Sound B obtained the figure of 38.71%. Option None was selected by 9.68% of subjects.

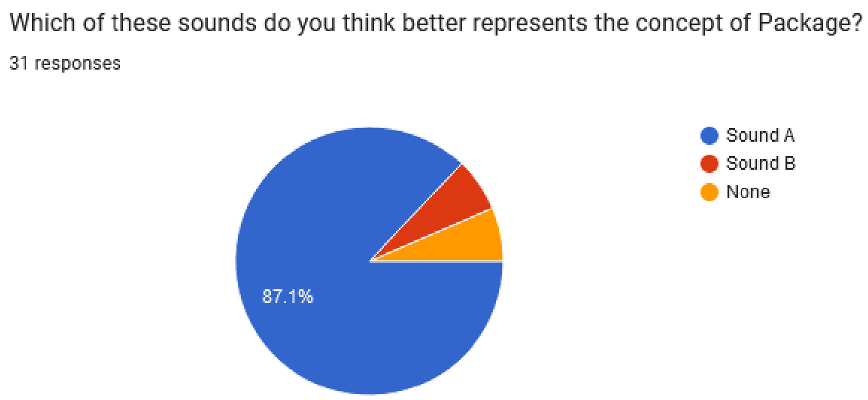


Figure 7.20: Results of the eleventh question.

Finally, the eleventh and final question concerns the element package. Option A achieved a percentage of 87.10%, and represents the sound proposed in the catalogue of good practices. Specifically, it is the sound of an envelope being opened, together with the opening of a zip. Option B is the sound of an explosion, as detailed in the catalogue with unsatisfied auditory principles, and was chosen by 6.45% of participants. 6.45% was also the result for the option None.

To conclude, the results are succinctly presented in table 7.3. This table contains for each of the questions, the number of answers A, B and None, together with their percentages. Highlighted in green, are the percentages with the majority of answers for that specific question. Highlighted in grey, are the answers that correspond to the sounds based on our proposed catalogue of sounds that follows the proposed auditory principles and the semiotics of the audible field.

Table 7.3: Summary of the results regarding the choices for the sounds.

Question	A	A(%)	B	B(%)	None	None (%)	Total Answers
01 - Class	21	67,74%	5	16,13%	5	16,13%	31
02 - Attribute	2	6,45%	27	87,10%	2	6,45%	31
03 - Operation/Method	3	9,68%	27	87,10%	1	3,23%	31
04 - Association	28	90,32%	3	9,68%	0	0,00%	31
05 - Inheritance	3	9,68%	19	61,29%	9	29,03%	31
06 - Realization/Implementation	19	61,29%	5	16,13%	7	22,58%	31
07 - Dependency	28	90,32%	1	3,23%	2	6,45%	31
08 - Aggregation	5	16,13%	22	70,97%	4	12,90%	31
09 - Composition	14	45,16%	10	32,26%	7	22,58%	31
10 - Class Association	16	51,61%	12	38,71%	3	9,68%	31
11 - Package	27	87,10%	2	6,45%	2	6,45%	31

7.3.3.3 Relevance Of The Principles For Constructing Auditory Notations

The results for how participants rated the individual relevance of the proposed principles for constructing auditory notations are presented in figure 7.21.

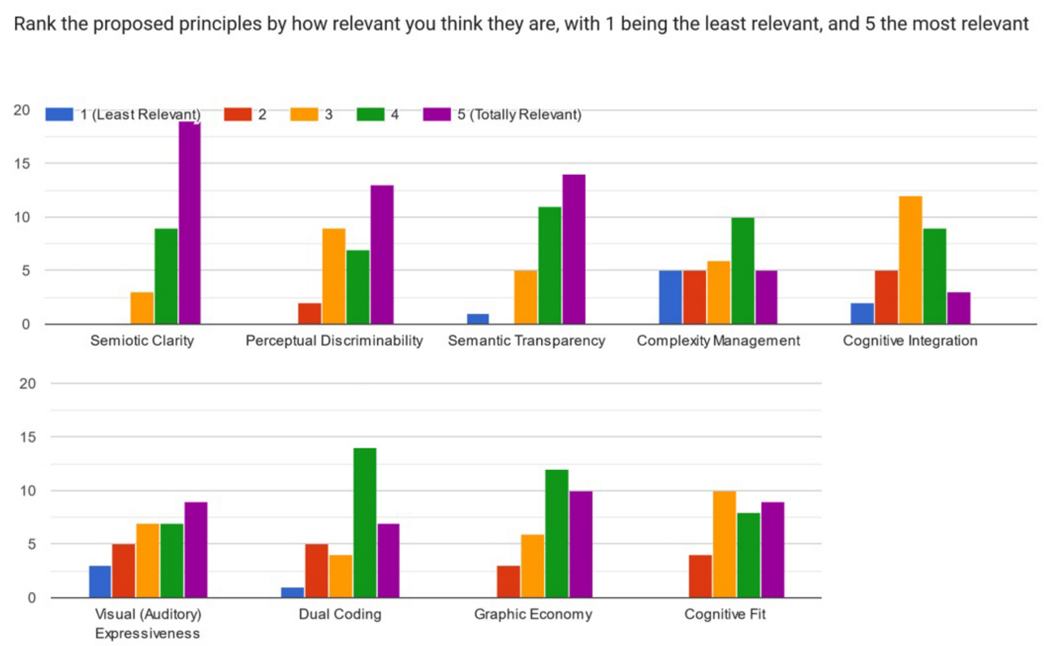


Figure 7.21: Results of the relevance of each proposed principle for Auditory notations, as rated by participants.

Table 7.4 summarises the information related to how the participants scored the relevance of each principle, along with their calculated average score. We can observe that Semiotic Clarity was considered the most important principle, whereas Complexity Management was the least important.

Table 7.4: Average score regarding the relevance of each principle, as rated by the participants.

Principle	1	2	3	4	5	Total Answers	Average Score
Semiotic Clarity	0	0	3	9	19	31	4,51613
Perceptual Discriminability	0	2	9	7	13	31	4,00000
Semantic Transparency	1	0	5	11	14	31	4,19355
Complexity Management	5	5	6	10	5	31	3,16129
Cognitive Integration	2	5	12	9	3	31	3,19355
Visual (Auditory) Expressiveness	3	5	7	7	9	31	3,45161
Dual Coding	1	5	4	14	7	31	3,67742
Graphic Economy	0	3	6	12	10	31	3,93548
Cognitive Fit	0	4	10	8	9	31	3,70968

Results from figure 7.22 show that 51.6% of the subjects agreed that the proposed principles were overall relevant and 41.9% strongly agreed. Only 6.5% had a neutral stance, and no disagreeing opinions were observed.

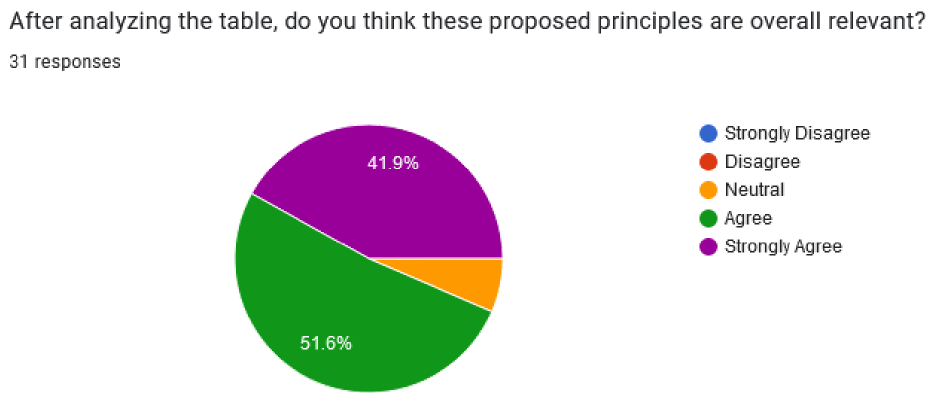


Figure 7.22: Results regarding how participants rated the overall relevance of the principles for constructing auditory notations.

7.3.4 Discussion Of Results

The results of this experimental study are analysed and discussed in the following section, which is further divided into 3 subsections: Personal Information, The Catalogue Of Sounds and the relevance of the principles for constructing auditory notations, respectively.

7.3.4.1 Personal Information

In this subsection, the results concerning the personal information of the subjects are discussed.

Observing the data regarding the personal information of the participants, there was a less expressive representation of females, compared to males. Approximately, in every 6 participants, 5 were male and 1 was female. Concerning their hearing, about 6.5% of all subjects identified themselves as hard of hearing, something which, despite being a small sample, helps to further diversify the profile of the answers obtained, allowing further observations as to the characteristics and perceptibility of the chosen sounds, which might not be so evident. Results also indicate that the vast majority considered themselves as having a good level of understanding of UML, being used regularly by around 35.5% of the respondents. From this population with regular contact with UML, 89.5% uses it in an academic context, while the remaining 10.5% uses it in a professional context. Although this sample of professionals in this field is narrower, it is very useful since we are offered a perspective where there will certainly be a high degree of demand and rigour, not only regarding our proposals for the sounds, but also in terms of the auditory principles, directed towards a usage within a professional context.

7.3.4.2 Catalogue Of Sounds

In this subsection, the observed results related to the catalogues and the choices of sounds made by the participants for the different diagrammatic elements that constitute UML class diagrams, are discussed.

According to the results obtained and summarised in the table 7.3, we can verify that in a total of 11 questions asked, where each corresponds to a different element of the UML class diagrams, the sounds that were proposed according to the catalogue of good practices were the choices with the highest number of votes in 90.91% of the questions, that is, in 10 of them. In 9 of these 10 questions, the result was above 50%, thus obtaining an absolute majority, while in 1, the registered figure was 45.16%. The sounds for the catalogue with unsatisfied auditory principles were the option with the most votes in 9.09% of all questions. This information can be found in table 7.5.

Table 7.5: Option with the most votes in the questions regarding the catalogues of sounds.

	Number Of Questions With The Most Votes	Number Of Questions With The Most Votes (%)
Catalogue Of Good Practices	10	90,91%
Catalogue With Unsatisfied Auditory Principles	1	9,09%

Furthermore, in the total of 11 questions, 7 out of the 11 sounds proposed according to the catalogue of good practices - that is, around 63.64% - obtained a very high rate of preference on behalf of the participants, which we considered to be above 65%. Specifically, these are the sounds for questions 01, 02, 03, 04, 07, 08 and 11, with these sounds obtaining the percentages of 67.74%, 87.10%, 87.10%, 90.32%, 90.32%, 70.97% and 87.10%, respectively, and correspond to the following elements in the diagram: class, attribute, operation/method, association, dependency, aggregation and package. We consider that with a percentage of answers above 65%, it becomes clearer that there is an evident preference towards a particular sound, as the gap in responses in relation to the other possible options - the other catalogue or None - becomes wider. As such, we believe that these results suggest that it is possible to infer with a certain degree of reliability that these sounds fulfil the objective of being more adequate choices to represent in an auditory manner, the diagrammatic elements in question, when compared to the sounds used in the catalogue of unsatisfied principles. Another aspect that further reinforces our belief that these results are indeed positive, was through the explanations given by the participants for their choices. In general, they seemed to intuitively understand the logic behind the creation of the sounds, without needing an explanation. In all of the aforementioned questions, the number of answers where the participants chose the sound that corresponds to the catalogue of unsatisfied auditory principles, had a modest expression, obtaining results below 10%. Only in questions 01 and 08, this catalogue of unsatisfied auditory principles obtained slightly higher values, with 16.13% in both. We believe that these figures are not very substantial and do not seem to imply any fundamental problems with the overall choices of sounds, since they are rather modest values. Additionally, the number of answers to the option "None" was not too expressive and even fewer suggestions and alternatives for sounds were given in regard to these diagrammatic elements.

In the remaining questions, 05, 06, 09 and 10, which respectively correspond to the elements inheritance, realization/implementation, composition and class association, we can observe that the answers were slightly more distributed between the three possible options. In these instances, the results may suggest other underlying factors, such as the proposed sounds not being so semantically clear and intuitive, potentially not fulfilling their purpose as effectively.

Concerning questions 05 and 06, the answers referring to the catalogue of good practices obtained the majority of votes, with 61.29% in both. Despite the less expressive result of the sounds corresponding to the catalogue with unsatisfied auditory principles (9.68% and 16.13%), these were, of all the questions, those which obtained the highest number of "None" answers - above 20% - together with question 09, obtaining the values of 29.03%, 22.58% and 22.58%, respectively. This may suggest that although the sounds from the catalogue of good practice were preferred over the sounds from the catalogue of unsatisfied principles, there may be much more suitable sound choices than those proposed by us. Logically, it was in these questions with the highest number of "None" answers, that we observed a greater number of responses to the question of proposing an alternative by the subjects, obtaining 6.6 and 3 responses respectively.

Observing the alternatives proposed by the participants, regarding question 5, two of the subjects suggested the sound of a baby. However, in this way, the same symbol would be used for two different concepts, which violates the principle of Semiotic clarity. In question 6, one of the participants explained that in his opinion, sound A would work with a little less background noise while another suggested that some other sounds such as hammering nails could be added. Although these two participants chose "None"on this question, their answers were in the sense of contributing to the sound already proposed by us and not by suggesting a new sound, as the question asked. Interestingly, two of the participants proposed the sound of a lamp turning on, with one of them specifying it as a cartoon sound. This proposed alternative was explained through the connection of the concept of realisation with the concept of "having an idea". This sound could be interesting to incorporate and test, as it might be a concept recognisable to most people. Additionally, the sound of an echoing arrow was also suggested by one of the participants. Moving on to question 9, one of the subjects referred that they could not imagine the sounds chosen by us as meaning composition, but at the same time could not find anything that might be a good alternative. Another participant suggested an orchestra sound without further explanation and lastly, the other participant who answered this question explained that they believed the elephant sound would fit this concept, as it gives the idea of a large entity, which they thought is appropriate for composition, while at the same time being a single uniform entity since we cannot delete part of the elephant and everything is unchanged to the remaining parts. This proposal could be incorporated and tested in the future.

Furthermore, it should be noted that it was necessary to overlook certain answers given by the subjects in this question of suggesting alternatives to the sounds, as many of those who selected the "None" option, only explained that they did not find any of the proposed sounds adequate, but did not have any other recommendation. In some cases, if they did, their answers were somewhat derivative of the sounds we proposed, suggesting the same sound for different concepts, which violates the principle of semiotic clarity. Additionally, some of the answers were also just suggestions and small modifications to the sounds already proposed by us. It is worth highlighting that because of the particular nature of this question, the task of obtaining good results proved to be arduous. We believe that this could be explained by the fact that we are asking for something outside the scope of the participants' area of expertise. In other words, we are asking computer engineers to think about sound design and suggest new sounds. Ideally in the future, this question could be further explored alongside people with a musical background and education.

Results show that the least expressive figures for the catalogue of sounds that follows the semiotics and the proposed auditory principles were obtained in questions 9 and 10, with a corresponding value of 45.16% and 38.71%, respectively. In both of these questions, the results of 32.26% and 51.61% were registered for the sounds belonging to the catalogue of unsatisfied principles, being therefore, the sound chosen by the majority of participants in question 10. Regarding the "None"answers in these two questions, question 09 obtained the percentage of 22.58% - having been previously discussed - and question 10 obtained 9.58%, a value that we consider not very expressive. Contrasting what happened in the previously analysed questions, in both of these, the percentages concerning the preference between the sounds for each of the catalogues were very closely matched. These results show that there was not quite as much consensus in the choices, suggesting that what should in theory be the most appropriate sound, was not so evident and intuitive, thus implying at first glance that the sounds chosen by us in these two questions, may not be the most appropriate.

Specifically in question 10, although the results were close, most participants chose the sound used for the unsatisfied principles catalogue, demonstrating that the sound we chose is not at all suitable, since it does not fulfil the objective of being a more adequate choice to represent in an auditory manner, the diagrammatic elements in question, when compared to the sound used in the catalogue of unsatisfied principles. A possible factor that may help justify this choice by the participants, is the fact that the sound itself is shorter and simpler, which may thus be more perceptible. The participants could have attributed a higher importance to these characteristics, therefore concluding that this would be the most appropriate sound to represent a class association. There could also have been some fatigue on behalf of the participants resulting in a lack of attentiveness. Analysing the answers given as justification for this choice, it is not possible to conclude anything substantial, as most of the explanations are directed towards the sound which belongs to the catalogue of good practices, justifying that it appeared to be the most intuitive and logical, with one of the participants highlighting that the arrow reminded them of the concept for association, and the paper unfolding, the concept for class. Most of the people who chose sound A, did not justify their choice, only 1 of the subjects justified it with the fact that it is a more basic sound, further explaining that it is like a neighbour class reaching out to another with the intent of association. This statement helps to corroborate our previous explanation. We believe that the results of this question may be somewhat inconclusive, since as mentioned before, the results between both catalogues were close (51.61% vs 38.71%) and the justifications regarding the sound from the catalogue of good practices were in accordance with our logic and reasoning. We believe that with a sample size bigger than 31 people, this question could be further examined and more conclusions could be drawn.

Overall, we consider that the results obtained were encouraging, since for 90.91% of the sounds, the subjects considered the ones proposed based on the auditory principles and semiotics to be more adequate when compared to the sounds used in the catalogue of unsatisfied principles, thus fulfilling the objective of being more adequate choices to represent in an auditory manner, the diagrammatic elements in question. Despite this, it should be taken into consideration that certain choices of sounds might be even better than those proposed in this study, since even though all the sounds here considered try to accommodate the universal human experience in order to be as understandable as possible, different people have different sensibilities and different life experiences, so there may always be certain sounds that won't be the most adequate for everyone. However, the main objective of this study was to observe whether there were indeed differences when using a more informed method for the construction of sounds for auditory notations in contrast to a naive approach. In that respect, the results here obtained suggest that there are indeed positive and noticeable differences. However, we think that this study would benefit from a larger sample size, so that these questions could be further explored.

7.3.4.3 Relevance Of The Principles For Constructing Auditory Notations

The following subsection discusses the results concerning the questions about the relevance of the principles for constructing auditory notations.

With respect to the relevance of each of the proposed principles, all were generally considered somewhat relevant by the subjects. Each principle was voted on a scale of 1 to 5. From here, we calculated the average of the responses obtained, as observed in the table 7.4, and it was possible to elaborate a ranking of importance. In descending order, they are as follows:

1. Semiotic Clarity (4.516)
2. Semantic Transparency (4.19355)
3. Perceptual Discriminability (4)
4. Graphic (Auditory) Economy (3.93548)
5. Cognitive Fit (3.709)
6. Dual Coding (3.67742)
7. Visual (Auditory) Expressiveness (3.45161)
8. Cognitive Integration (3.19355)
9. Complexity Management (3.16129)

As can be observed, the 3 principles considered most relevant were Semiotic Clarity, semantic transparency and perceptual discriminability. It should be noted that the principles that were considered most relevant were those that had the strongest emphasis across our choices for sounds, as well as being the repeatedly unsatisfied principles in the catalogue of bad practices. These results seem to suggest that there is a correlation between the exposure that participants have had to a particular principle and how relevant they consider it to be. For example, we expected dual coding to be considered as a more relevant principle than the position it was placed in by the participants. We believe that this may be explained by the fact that it was a principle that could not be directly tested in the context of this experimental study, thus participants did not have firsthand contact with it as they did with other principles, which could have affected their perception of its relevance. To corroborate our statement, it is possible to observe that the principles considered less relevant were those which were not feasible to test in the context of this study. This is the case of cognitive integration and complexity management.

Regarding the overall relevance of the proposed auditory principles, these were in general, well received by the participants. As it can be observed, 51.6% of subjects agreed and 41.9% strongly agreed that the principles are overall relevant, resulting in a total of 93.5% positive answers. Only 6.5% of the respondents had a neutral stance and there were no disagreeing opinions recorded. These results seem to support our second hypothesis which states that the proposed auditory principles and semiotics of the audible field are relevant from the user's point of view.

7.3.5 Threats To Validity

In this section, the threats to validity that were identified will be presented.

Even though all the sounds considered for the proposed catalogue that satisfies the auditory principles try to accommodate the universal human experience in order to be as understandable as possible, different people have different sensibilities and different life experiences, so there may always be certain sounds that won't be the most adequate for everyone.

Additionally, although the proposed sounds for the catalogue of unsatisfied auditory principles have a structured justification and were chosen because they do not follow the proposed principles and semiotics, there may exist a certain bias on our behalf concerning that particular selection of sounds. This is due to the fact that there is already the intention from the onset, that these sounds should not make as much sense when compared to those chosen for the catalogue of sounds that satisfies the auditory principles. This may influence the subjects' decision to choose a sound not necessarily because they consider it appropriate, but by excluding the other one that sounds blatantly out of place.

The participants might not have had the right conditions for obtaining ideal results during this study, since the experiment was not performed in person and under our control. The subjects might have been in a noisy environment and more inattentive. This is evidenced by the results of one of the optional personal information questions. More people than those who were supposed to, answered it, as it was meant only for those who had responded "I USE IT REGULARLY"to the prior question.

As the form could not have sounds directly embedded, a workaround had to be arranged. In each question, the subjects had to click on "Sound A" and "Sound B" to open the links that allowed them to hear the sounds in a new tab. This increases the possibility of making a mistake or swapping sounds, and accidentally selecting the less suitable sound.

Concerning the relevance of the proposed principles, due to their very nature, it was not possible to test all of the principles. For example, in the case of dual coding and cognitive integration, it would only be possible to test them in a different context. As such, the fact that the participants did not have direct contact with all principles may bias their perception of the relevance of each one of them.

Lastly, since the participants were chosen through convenience sampling, there may be certain profiles which are not represented.

8

Validation In The Context Of Use: Auditory Diagrammatic Readings

This chapter seeks to validate and discuss the drawn conclusions regarding our proposals for a diagrammatic reading of UML class diagrams, constructed with sound, obtained through the findings detailed in the previous chapters. To this end, we present the construction and implementation process of our developed prototype, a proof of concept named *SoundUML*. The architecture of this tool will be specified, together with the tools used in its implementation, the technological options taken to integrate the voice synthesis mechanisms and audio playback, as well as a guide with the steps needed to set up and utilise the tool. Finally, an experimental study is presented, with the purpose of evaluating our proposals and the tool.

8.1 SoundUML: A Tool For Auditory Readings Of UML Class Diagrams

This section briefly describes our technological prototype, along with its intended purposes.

The tool here presented, which we named **SoundUML**, is a prototype that allows UML class diagrams to be read to the users, through the usage of voice synthesis and sound playback. It is a module developed for Modelio - an open-source UML tool - and has the intended purpose of serving as a testing instrument, in order to understand if our proposals for diagrammatic readings are effective when compared to the common, visual approach. SoundUML was also created to help draw conclusions regarding the validation of the principles for an auditory notation proposed in this document, while testing the feasibility and usability of working with sound and voice synthesis when using UML, more specifically Class diagrams. Additionally, it aims to contribute to the inclusion of visually impaired and blind people in activities of this type.

As a proof of concept, SoundUML seeks to incorporate the proposed auditory principles and design guidelines discussed in chapter 5, along with the drawn conclusions from the experimental studies detailed in chapters 6 and 7. As such, the reading provided by our tool is based on the previous two experimental studies that we conducted during this dissertation. The first study, from chapter 6 was aimed at detecting any emerging patterns regarding how people read UML class and state diagrams. The second, from chapter 7, consisted in creating two catalogues of sounds for the different elements that constitute a UML Class diagram, where one of the catalogues followed the proposed auditory principles and design guidelines discussed in chapter 5, while the other catalogue disregarded those principles. Afterwards, we asked participants to choose which sounds they considered the most suitable. With the results from this study, we hope to achieve greater efficacy regarding the perception of different sounds on behalf of users.

SoundUML was created as an extension to Modelio, which is an open-source UML modelling program. This program was chosen as the foundation, because it is a well-known open-source editor and therefore, we believed that it would have an established community of users and developers, while also being widely adopted in both commercial and academic environments. Modelio was mostly made in Java as well, which due to our increased familiarity with the language, would allow greater agility and speed in the process of implementing a prototype that would serve our testing needs.

The project's source code, along with the latest compiled version, can be consulted and downloaded through its GitHub repository[1].

8.2 SoundUML Architecture

In this section, a general outline of the tool's architecture will be presented.

As previously mentioned, **SoundUML**, is a module developed for Modelio which allows UML class diagrams to be read to the users, through the usage of voice synthesis and sound playback. Modules are, essentially, extensions for Modelio. According to its developers, they can be seen as the equivalent of extensions for an internet browser [56].

Figure 8.1 illustrates the architecture of *SoundUML*. Note that this figure is merely an illustrative general outline, serving to give an insight into what we are discussing.

Succinctly, the SoundUML tool is implemented in three layers, each with a distinct purpose that ensures the functioning of the tool.

In **the first layer**, lies the Modelio API. This API is organised into several services, each specific to a particular aspect of Module development, with the Eclipse Modelling Framework as the foundation for its implementation. This layer allows to perform several operations that serve the tool's implementation, namely obtaining the different elements (MObjects) that constitute a class diagram, as well as their properties and attributes. By having access to these MObjects, we can distinguish which specific element we are dealing with, through the various existing interfaces for the different elements that constitute the UML metamodel. Concretely in terms of implementation, we check to see if each

[1] https://github.com/SuperGuerreiro/soundUML

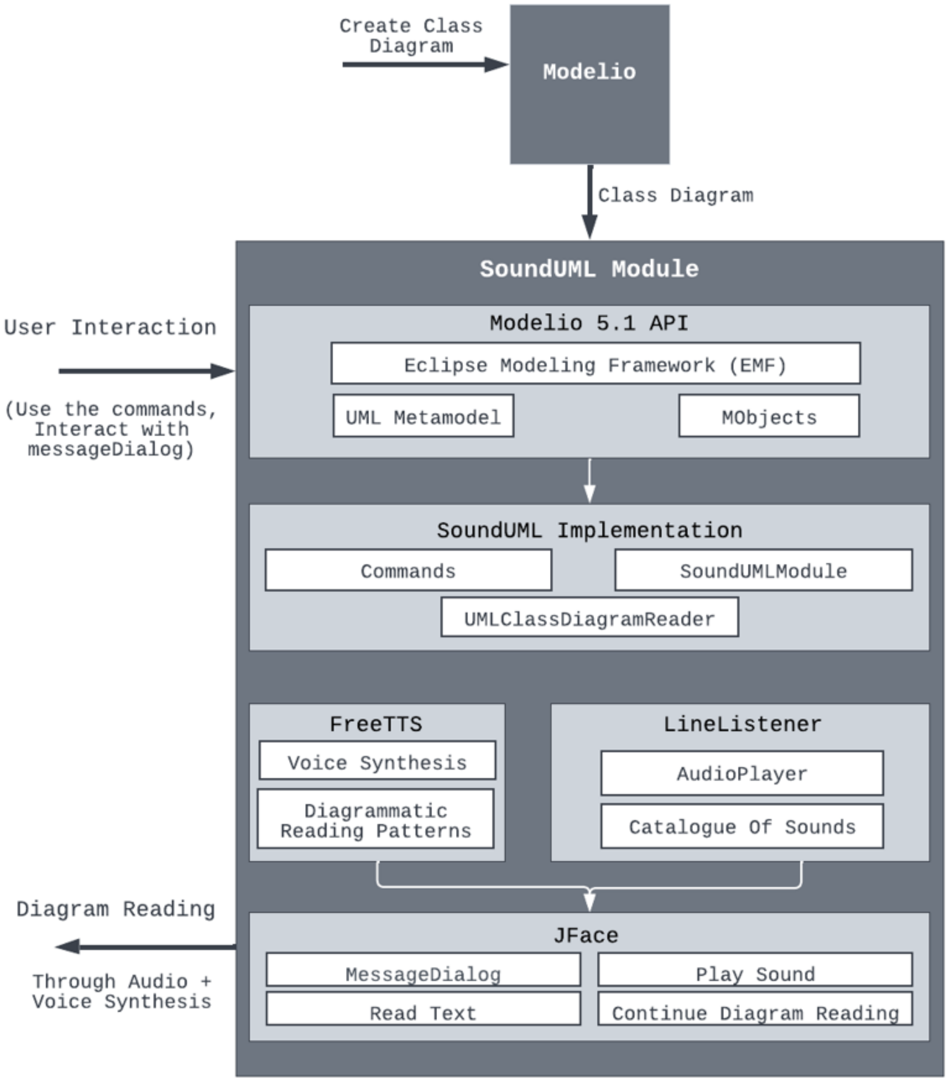


Figure 8.1: General outline of SoundUML's architecture.

element of the diagram, which is a MObject, is an instance of the specified interface that corresponds to that particular element of the UML metamodel.

The second layer is concerned with the implementation itself, responsible not only for the commands and operations that the users will be able to perform, but also for the information that will be displayed and how it will be presented. We highlight the class *UMLClassDiagramReader*, where each element that constitutes a class diagram will be handled, and the information for each existing element in the class diagrams is treated and parsed. In this class, decisions were taken based on the diagrammatic reading patterns detected in the chapter 6, such as the order of priority given to the different types of existing relations, with special attention to the inheritance relations. Certain aspects and details of the implementation of this class will be further detailed in section 8.4. It is also worth mentioning that, due to the modular nature in which this tool was designed and created, it would be possible to extend its functionality to support other types of

UML diagrams in the future. Furthermore, it would also be possible to implement other commands, as well as integrate a module that allows speech recognition, such as the Google speech-to-text API thus enabling the ability to use voice commands.

The remaining tools essential to the functioning of SoundUML are located in the **third layer**. This layer is logically very interconnected with the implementation layer, as the functionalities of these tools are extended and implemented, enabling us to achieve what is intended for SoundUML. Starting with FreeTTS, it allows to send an audio message through voice synthesis, performing the reading of the important details of the class diagrams, according to some of the details and conclusions regarding the diagrammatic reading patterns, which were taken from the study conducted in chapter 6 and implemented in the layer above. LineListener, in turn, was used for the implementation of the AudioPlayer, which allows a sound to be assigned to each element that makes up the UML class diagrams. These sounds are the ones that were thought of and tested in chapter 7, where the ones with the most votes from the participants were selected and integrated into SoundUML. These two tools combined make possible the implementation of the principle of dual coding, since in the same way that text can be used to complement graphics in visual notation, we are joining the reading of important information from the diagram, to the usage of sounds, thus reinforcing the meaning of a given element. Finally, to facilitate user interaction, all the information presented to the users will be displayed through a message dialog, implemented through JFace. It was necessary to extend its functionalities, in order to have multiple buttons and a persistent message that does not disappear unless the user wants to. These newly implemented features allow the user to play the sound for that specific element or hear the TTS read the text as many times as they want. Additionally, some other features were incorporated, which enable the users to go back to a previous step in the diagrammatic reading, continue forward to the next element, or finish the diagrammatic reading at any time, allowing for improved usability, as well as a slightly friendlier user experience.

8.3 Essential Tools For The Development Of SoundUML

In this section, the essential tools for the development of SoundUML are listed and described, along with some of the decisions that led to their usage.

8.3.1 Eclipse Modeling Framework (EMF)

The created solution required the manipulation of UML class diagrams and consequently, the elements that compose them, so that the diagrammatic reading could be performed. For this purpose, we decided to use the software Modelio as the platform for the development of our tool. This program, in turn, has the Eclipse Modeling Framework (EMF) as the foundation of its implementation, using and extending the interfaces and methods that it provides.

EMF is an Eclipse-based modelling framework and code generation facility for building tools and other applications based on a structured data model. EMF unifies three important technologies: Java, XML, and UML. From a model specification described in XMI, EMF provides tools and runtime support to produce a set of Java classes for the model, along with a set of adapter classes that enable viewing and command-based editing of the model, and a basic editor [57].

EMF provides a runtime framework that allows any modelled data to be easily validated, persisted, and edited in a UI. The core EMF framework includes a meta model (Ecore) for describing models and runtime support for the models including change notification, persistence support with default XMI serialization, and a very efficient reflective API for manipulating EMF objects generically [58].

8.3.2 Modelio

Modelio is an open-source UML and BPMN modelling environment developed by Modeliosoft. It supports a wide range of models and diagrams, while providing many services that facilitate the modelling of architectures, such as model assistance and consistency-checking features [59].

This program was chosen as the foundation for our prototype, because it is a well-known open-source editor and therefore, we believed that it would have an established community of users and developers, with active forums, serving to assist us in some of the challenges that could arise during the development of our tool, while also being widely adopted in both commercial and academic environments. Furthermore, Modelio was mostly made in Java, an aspect that weighed on our decision to use it, due to our increased familiarity with the language, as it would allow greater agility and speed in the process of implementing a prototype that served our testing needs.

Modelio offers an API for development purposes, supporting both the creation and usage of Jython scripts to automate some tasks, as well as the development of modules using Java to extend the program's functionalities [59]. In order to achieve a vision as close as possible to what we intended, and to materialise and implement the concepts discussed throughout this document, it was necessary to extend Modelio's functionalities. As such, Jython scripting would not serve our objectives. Instead, we chose to develop a module for Modelio, as it was the option that made the most sense for our purposes.

A module, in turn, according to its developers, can be seen as the equivalent of extensions in a web browser. A module is a set of java, XML and resource files that are compiled into a compressed archive with the .jmdac extension. This file can then be deployed and executed in Modelio [56]. A module can provide additional metamodel material that enriches the Modelio metamodel, additional functions (commands) and additional GUI (views, dialogs and diagrams). Additional commands, views and diagrams are developed in Java code using the Modelio API, which is a documented set of interfaces and classes that provide basic and advanced services required for the development of a module. The

Modelio API is organised into several services, with each being specific to a particular aspect of Module development. In practical terms, these services are combined, in order to implement most of the functionalities of a module. The Modelio API is delivered as a unique jar file, named "modelio.jar" [60]. As previously mentioned, this API uses EMF as the foundation of its implementation, utilising and extending the interfaces and methods that it provides. Some of the aspects concerning the implementation process, will be discussed later on.

8.3.3 Maven

Module development requires the usage of Apache Maven, which is a software project management and comprehension tool. Maven can manage a project's build, libraries and documentation from a central piece of information [61]. Every module, and consequently, our tool, uses a Maven archetype, which is a Maven project templating toolkit. Specifically, modules use the *modelio-module-archetype*, which is essentially an Eclipse Maven project with a "conf" directory containing all the elements that are specific to the Modelio module that should be packaged with it [59].

8.3.4 JFace

The Eclipse project defines JFace as "a UI toolkit that provides helper classes for developing UI features that can be tedious to implement". JFace is window-system-independent in both its API and implementation, and is designed to work with Standard Widget Toolkit (SWT) - an open-source widget toolkit for java - without hiding it. JFace includes the usual UI toolkit components of image and font registries, text, dialog and frameworks, among others [62].

In our prototype, we specifically used the class "MessageDialog" to create a dialog that shows messages to the user. To this end, we extended its functionalities, in order to enable our tool to perform the intended interactions, such as allowing the users to play the different sounds or use the TTS as many times as they want, with the dialog box remaining on the screen, as well as advancing or going back one step in the diagrammatic reading.

We decided to choose JFace to implement the UI, since it is widely known and used. As such, plenty of online resources were found, which allowed for the swift development of the UI component. Furthermore, it was the practice followed by the Modelio developers, both in Modelio itself (which extends jface), and in the available examples regarding module development.

8.3.5 Javax - Clip API and LineListener

Javax means "Java Extension". It is a package in the Java library that contains numerous classes and interfaces that are used for various purposes, such as extending the capabilities of the core Java language.

Clip API is an unbuffered or in-memory sound API for Java. The Clip class offers mechanisms for media playback and is part of the javax.sound.sampled package. Before playing back, the entire audio file is loaded into memory, and the user has total control over the playback, allowing to loop sounds. Additionally, we implemented the LineListener interface in order to receive line events - OPEN, CLOSE, START, and STOP - for the playback [63].

This simple audio player enables our tool to play the sounds created in the catalogue of sounds, for each of the elements that constitute a UML Class Diagram. We chose to implement an audio player in this manner, as the Clip class is useful when reading and playing short sound files, which is exactly the intended use in our tool, as well as being easy to integrate into the Eclipse environment.

8.3.6 FreeTTS

FreeTTS is an open-source, speech synthesis system. It is written entirely in Java, so it can be easily integrated into an eclipse development environment, as was the case with our solution [64]. To synthesize speech, FreeTTS breaks the input text into sets of phonemes and then converts them into audible speech by performing successive operations on the input text. FreeTTS stores the cumulative results of each operation in an utterance structure that holds the complete analysis of the text [65].

The core of FreeTTS is an engine that contains a voice and an output thread. The former thread consists of a set of utterance processors that create, process, and annotate an utterance structure. Associated with the voice is a data set that is used by each of the utterance processors. In turn, the output thread is responsible for two actions: synthesising an utterance into audio data and then directing this data to the appropriate audio playback mechanism [65].

This tool was chosen because, as already mentioned, it is easy to integrate into the Eclipse development environment, which was the case with our solution. It is also open-source, meaning that there are several available resources that could assist us during the implementation process. Other options were explored, including those that are not open-source, but these were not selected, as they did not offer any kind of direct integration with java or a medium-term trial period that allowed for the testing of our solution and subsequent utilisation. Additionally, FreeTTS also interfaces with the MBROLA synthesizer - a speech synthesizer based on the concatenation of diphones [66] - and can use the available MBROLA voices. These voices sound less robotic in relation to the ones used by FreeTTS. Unfortunately, according to its developers and our testing, FreeTTS does not support MBROLA on the Windows platform. As such, the voice used in this prototype is one of the available voices provided by FreeTTS, which is a medium quality, unlimited domain, 16kHz diphone voice, called "kevin16".

8.4 Implementation Of SoundUML

After describing the overall architecture and the features of each layer that composes SoundUML, along with the essential tools for its development, we will specifically talk about certain aspects of its implementation.

8.4.1 Module Development Structure: A General Overview

A Module is defined as a set of XML, java and resource files, before being packaged in a compressed archive with the .jmdac extension. This file can then be deployed in user projects and executed in Modelio. Module creation is composed of two parts: XML declaration and Java coding.

To develop a module, we have to use a Maven archetype, which is a Maven project templating toolkit. Specifically, modules use the *modelio-module-archetype*. Essentially, this archetype is an Eclipse Maven project with a `conf` directory that contains all the files specific to the module, and that should be packaged with it. As such, these files are present in our prototype's project, and have been configured accordingly. The files are:

- `pom.xml`: As this is a Maven project, there is a Project Object Model (POM) file. This XML file contains information about the project and configuration details used by Maven to build the project, in order to compile and package the module into a .jmdac file. Here, we included a dependency to freeTTS' repository.
- `src/main/conf/module.xml`: Inside the project, there is a .xml file that is necessary for the correct functioning of our tool. It contains all the declarations of Modelio module objects such as the module, profiles, stereotypes, tag and note types, commands and module parameters.
- `src/main/conf/module.properties`: This file contains the mandatory resources required by the module. It contains mainly the string names and the descriptions of the module and module elements (commands, annotations, etc).

Every module contains two important classes and their respective relations. These classes are detailed in the UML model diagram presented in figure 8.2. We will not delve into greater detail in regard to these classes, since these are automatically generated during the creation of a new Maven project with the aforementioned module archetype. However, to give some brief context to the purpose of these classes in the overall implementation of the tool, we detail them as follows:

- `SoundUMLModule` class represents the module at runtime. This class is considered the main class of the Module and by convention it is suffixed with the word "Module".

- `SoundUMLLifeCycleHandler` class is in charge of the implementation of the Module Session Services. From its deployment into a Modelio user project to its operational use, a module runs through several states called runtime Phases. The operations that this class defines are automatically called according to the Module's runtime states and events.

The detailed UML model diagram for these classes is presented in figure 8.2.

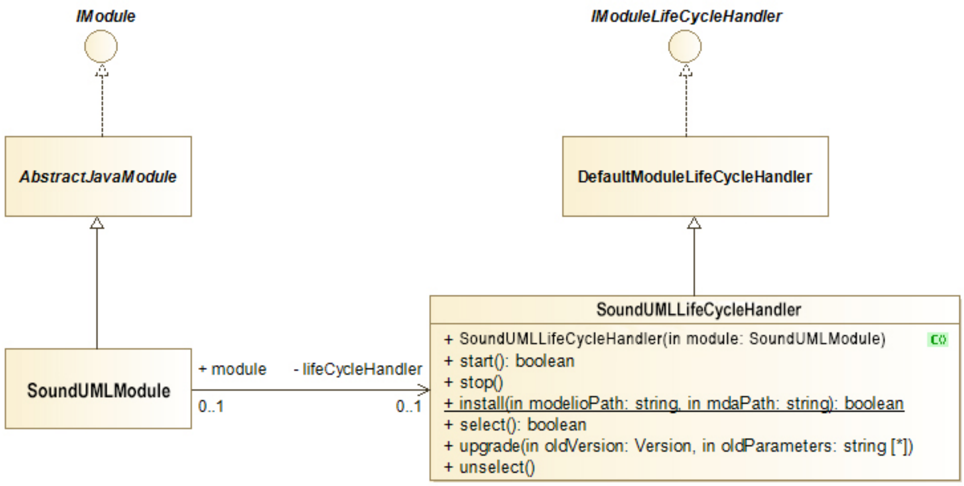


Figure 8.2: The *SoundUMLModule* Life Cycle. Adapted from [67]

8.4.2 Project Overview

This section is concerned with how the project is organised, together with a brief detailing of the most important classes and how they interact and relate to each other, as well as some particular implementation aspects that we consider important to highlight.

Figure 8.3 broadly illustrates, the class diagram with the most important classes that constitute our prototype and that ensure its proper functionality, organised in their respective packages. In order to streamline the explanation of our implementation, certain details are omitted in this figure, namely some of the variables and methods that constitute each of these classes, as well as specific classes generated during the project's creation. The full-sized figure can be found in the Appendix C.1. As already mentioned, the project's source code, along with the latest compiled version, can be consulted and downloaded through its GitHub repository[2].

8.4.2.1 ReadDiagramCommand Class

In the package `org.modelio.soundUML.command` is where the commands that can be executed by the tool's users can be found. Command creation is a two step process, consisting of a declaration in the `module.xml`, along with a Java class that extends the class

[2] `https://github.com/SuperGuerreiro/soundUML`

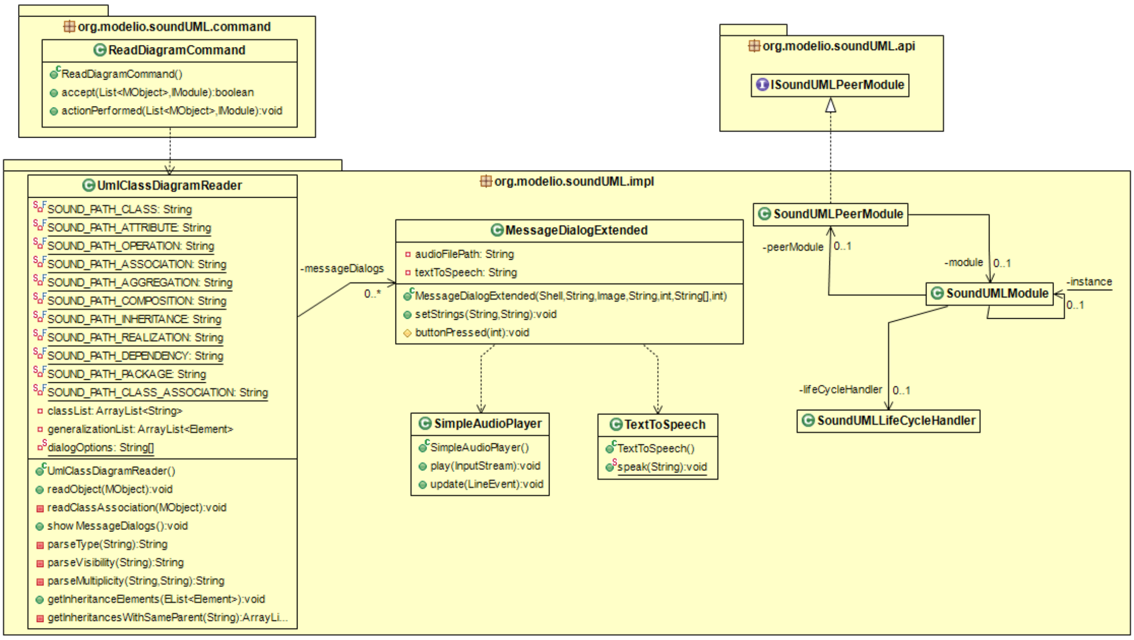


Figure 8.3: Summarised SoundUML's Class Diagram, detailing the most important classes, their respective relations and packages.

`DefaultModuleContextualCommand` and redefines the appropriate operations. Naturally, it is in this package that you can find the class concerning the command that handles the reading of a given UML class diagram. In this class, called `ReadDiagramCommand`, there is the method `actionPerformed()`, which is called when the command is activated by the end-user, meaning that its role is to perform the proper action. Thus, the code of this method is the action itself.

In order to perform the reading of a UML class diagram, firstly, we needed a way to access the elements that compose the class diagram. It should be noted that Modelio works by setting up projects where we can then build diagrams. As such, to create any diagram in Modelio, we need to always create a project, to where the diagram will obligatorily belong. Modules are then installed on each individual project created in Modelio. We can have many different types of diagrams inside the same project. Therefore, it was necessary to first fetch all the diagrams that exist within the given project where we want to perform the UML class diagram reading, through the interface IModelingSession, which represents a project session. The diagrams obtained are of type `AbstractDiagram`, which is the graphical representation of a model. Afterwards, we created the instance of the class that will perform the reading, which is the `UMLClassDiagramReader` class, on which the class `ReadDiagramCommand` depends, as can be seen in figure 8.3.

Next, we iterate the fetched diagrams that can exist inside a Modelio project, where for each, we access the elements that are contained in that diagram, and store them in a list. Then, each of these elements is iterated as well. Here, we are only concerned with the UML elements that belong to the UML Class diagrams. More concretely, we are concerned with the top elements in the hierarchy, which we can assume will always be either classes or packages, as the remaining elements are always child objects of these two. We ensure that this happens through an if condition that checks if the element in question is always the top element in the hierarchy. Finally then, the `readObject()` method of the `UmlClassDiagramReader` class is called, which will read these objects of type Element. After the diagram has been traversed and all the elements are cycled through, the `showMessageDialogs()` method is called. This method displays the message dialogs regarding each of the elements that make up the class diagram that we are reading, one at a time.

We should also mention that in a first version of this method, all objects besides the top ones in the hierarchy, were read directly from the `ReadDiagramCommand` class, but this caused unwanted repetition of elements during the reading. This happened because all the other elements that exist in the diagram (relationships, attributes, methods, etc.) are child objects of those top-level elements (classes and packages). So we opted to implement the reading of those elements only inside the `readObject()` method, while we are handling the reading of classes and packages, ensuring in that manner, the reading order we desire.

After the creation and implementation of this java class, it was necessary to declare the command within the Graphical User Interface (GUI) section of the `module.xml` file. Here there are a number of parameters that have to be filled in. Additionally, it was necessary to edit the `module.properties` file with the text that will be displayed in the tool, for this command.

Furthermore, given the modular nature of our tool's implementation, it would be possible to create additional commands for future functionalities. To do so, it would only be necessary to create within this package, a new class for each command that we wish to add, and follow the steps previously described.

8.4.2.2 UMLClassDiagramReader Class

Moving on to the `UmlClassDiagramReader` class, this is the central class of our prototype. It is located inside the package `org.modelio.soundUML.impl`, along with all the classes related to the implementation itself. Essentially, this class handles and parses each type of element that is part of UML Class Diagrams, according to their respective attributes and unique sound (as defined in chapter 7). In this class, there is a core method, called `readObject()`, where we initially receive the top elements of the diagram (classes and packages), with those objects being of type `Element`. However, as the `MObject` interface is the topmost parent of any model element, it is a superinterface of the `Element` type. Thus, we treat all the diagram elements as MObjects, as this interface provides generic accessors that will enable us to obtain certain specific attributes of a given element.

This method, in a succinct manner, receives as a parameter, a diagram element in the form of an object of type `MObject`, that will be read and handled according to the specific characteristics of that element's type (for instance, a Class, Operation, Generalization, Dependency, etc.). We are able to achieve this, since the various model elements that exist in UML Class diagrams, are represented in Modelio's API via their own interfaces, which in turn, are sub-interfaces of the type MObject. As such, we have several if conditions, where in each one, we check if this MObject is an instance of a type that corresponds to a particular model element.

Afterwards, depending on the if condition that has been satisfied and consequently, knowing the type of element that we are dealing with, we treat it according to the attributes that constitute it, and that are intended to be presented during the reading of that specific element, to the users. For this end, we access the attributes that comprise it, through the methods that are defined for that specific type (listed in the Modelio API), and we define and construct the strings with all the information that will be read by the tool. This string is then passed to a message dialog, which will allow the TTS to read the information regarding this concrete element, along with the file path for the unique sound for that particular model element, through the method `setStrings()`, that belongs to the class `MessageDialogExtended`.

Each message dialog is created right after the parsing of the information concerning a given diagram element. Therefore, these dialogs are unique and relative to each element of the diagram that will be read. Each of them contains the string with the specific information to be read about that particular element, as well as the sound specified for elements of that type. Next, each message dialog is added to a linked list, called `messageDialogs`, which contains all the message dialogs of all the elements in the diagram, being added to this list as the elements of a given diagram are being traversed, and consequently, handled by the `readObject()` method. In this way, they are added to this list, in the order that we intend to show the users. Therefore, it is possible to guarantee the persistence of the position at where the tool is during the reading, allowing users to navigate the diagram backwards and forwards, or end the reading at any time, without any problems.

Regarding the sounds, we defined 11 sounds for 11 different types of modes elements, mirroring what is specified and described in the catalogue of sounds of chapter 7. The sound file associated with each of the elements that make up UML class diagrams, is defined in a set of constant string values that indicate the path for each of the sounds that should play when a specific element of the diagram is being read. For example, the constant string named `SOUND_PATH_ATTRIBUTE` specifies the file path for the sound that will play when the tool is reading an element of the diagram, that is an instance of the type attribute. The same logic is applied to all the other constant values that can be observed in figure 8.3.

It should be noted that these sounds were implemented in such a way that they play and function correctly regardless of the machine where SoundUML is installed. To do this, we had to locate the generic path where the compiled .jmdac project fetches the resources, in order to embed the sounds into the compiled version of the module. The path to the location of the audio files from the project source is: `src\main\resources\org\modelio\soundUML\sounds`. However, in the compiled file, the files are retrieved directly from the resource folder, so we just reference the path from there. As such, `org\modelio\soundUML\sounds\filename.wav`, is the correct path, with "filename.wav" being the name of the file we want the tool to play. The sounds were exported in .wav, since this was one of the file formats supported by the implemented audio player.

The reading implemented in this class, in turn, follows some of the patterns detected in Chapter 6. For example, for any diagram, it is always ensured the reading order of: a class, followed by its attributes and operations, and afterwards, the outgoing relationships of that class. Only after this sequence, do we move on to the next class in the diagram, again following the described order. When reading the operations and attributes, the types are always read. In the operations, the number of parameters, is always stated as well. In the case of the relationships, the roles (if there are any), along with the respective multiplicities of each end of a relationship, are always described. In the cases where we treat an element that is a class - in other words, when the MObject is an instance of the type Class - the child objects of this element (these being its attributes, operations and relations) are immediately iterated. For each of these objects, the method `readObject()` calls itself and deals with each one of these elements as an instance of its respective type, making it a recursive method.

Special emphasis is given to inheritance/generalization relationships as well, since it was detected in the diagrammatic reading study, that participants considered them to be among the most important. The reading of inheritances is then performed, during the reading of both the parent class, and the child class. For this, we have created the methods `getInheritanceElements()` and `getInheritancesWithSameParent()`. In the case of the former, this method receives the list with all the elements of a diagram, iterates through it and adds the elements that are instances of the type `Generalization` to an array list. This list will only contain all the generalizations/inheritances that exist in that specific diagram, and is called `generalizationList`. The latter method, in turn, receives as a parameter, a string that is the unique identifier of a given class, and iterates through the generalizationList, comparing if the unique identifier passed as a parameter is the same as the unique id of the parent of each generalization in this list. In the end, it returns all the generalizations/inheritances where that class is the parent class. If a class is not the parent in any generalization, an empty list is returned. In this manner, it is possible to perform the reading of all the incoming generalization/inheritance relationships of a given class (or in other words, the inheritance relationships where a particular class is the parent).

Once the iteration over all the diagram elements has been completed and the reading order has been arranged, the method `showMessageDialogs()` is called in the class `ReadDiagramCommand`, as already described. This method shows the message dialogs, contained in the aforementioned linked list, to the users. As the message dialogs are already defined and stored in their proper order, we allow user navigation through the iteration of this list. In reality, when users are interacting with the UI and choosing to go forward or backwards in the reading performed by the tool, they are simply navigating through this list.

To adequately parse some of the information, we had to create a few auxiliary methods to handle the strings obtained. They are `parseType()`, `parseVisibility()` and `parseMultiplicity()`.

It should be noted that the modular nature of this tool, would allow its extension to other types of diagrams beyond UML class diagrams, if so desired, in a similar manner to what has been described.

8.4.2.3 MessageDialogExtended Class

The class `MessageDialogExtended` extends the `MessageDialog` class that belongs to the JFace toolkit. Essentially, this class allows the users to interact with the message dialog box. This message dialog allows us to play the sounds, along with the string we want the TTS to read. As already mentioned, a message dialog is created for every element of the diagram that is read by the method `readObject()` from class `UmlClassDiagramReader`. This means that every message dialog is unique.

The method `setStrings()` receives both the *filepath* for the sound relating to each of the element types that constitute a UML Class diagram, along with the string to be read by the TTS.

The method `buttonPressed()` handles the logic that enables the users to play the sound or hear the TTS as many times as they want. For this, it was necessary to override this method, in order for the dialog box to remain on the screen after pressing any button, as the default behaviour would close the dialog box after pressing any of its options once. If the user presses the button that plays the sound, the `SimpleAudioPlayer` class is instantiated, and the file specified in the *filepath* is played. If the user presses the button to make the TTS read the intended message that was passed to this class, it calls the method `speak()` from the `TextToSpeech` class.

8.4.2.4 Additional Classes

The following section will discuss the remaining interface and classes present in figure 8.3.

The `SimpleAudioPlayer` class, as the name suggests, is where we find the implementation of the audio player that enables our tool to play the sounds created in the catalogue of sounds, for each of the elements that constitute a UML Class Diagram, through the usage of both the `Clip` and `LineListener` interfaces from the javax package. The Clip class is useful when reading and playing short sound files, which is exactly the intended use of our tool. The `LineListener` interface, in turn, is able to receive line events (OPEN, CLOSE, START, and STOP) for the audio playback. This class has the method `play()`, which plays a given audio file, received as an `InputStream`. Additionally, we implemented the method `update()` from `LineListener`, which listens to the START and STOP events of the audio line.

The `TextToSpeech` class, in turn, implements the Text-To-Speech functionalities, using freeTTS. This class has the method `speak()`, which receives the text that we want to be read as a String. In this method, we defined certain properties of how the desired text will be read, such as the rate, pitch and volume. We also defined the voice that will perform the reading, with this being one of the available voices provided by FreeTTS. The chosen voice is a medium quality, unlimited domain, 16kHz diphone voice, called "kevin16".

Finall, we have the interface `ISoundUMLPeerModule`, found in the package `org.modelio.soundUML.api`, and the classes `SoundUMLModule`, `SoundUMLPeerModule`, `SoundUMLLifeCycleHandler`, found in the package `org.modelio.SoundUML.impl`. These classes, which are automatically generated during the creation of a new Maven project with the aforementioned module archetype, have already been briefly described. The class `SoundUMLPeerModule` implements Module services.

8.5 Installing SoundUML: A Step-By-Step Guide

The following section will illustrate how to setup and install the developed solution, in a step-by-step guide.

To install and use SoundUML, we need to have Modelio installed on the computer. After opening the program, we are greeted with Modelio's main interface page, where we can select any user project. Before that, as illustrated in step 1 of figure 8.4, on this page, we need to click on the "Configuration" option, present in the program's menu bar. A menu will open, which contains the option "Modules catalog...". We press this option, and the window shown in step 2 will then pop up. Here, we click on the option "Add a module to the catalog...". A file explorer window will open, where we have to select the file "SoundUML_1.0.00.jmdac", which is our compiled tool. After selecting this file, we can confirm in the "Modules catalog" list, that this module has been successfully installed in the program.

Once these steps are completed, we have SoundUML installed in the Modelio program. However, because of the way Modelio works, in order to use it, we need to also add the module to a specific user project. After opening the user project where we want to use SoundUML, as illustrated in step 4 of figure 8.5, we click again on the "Configuration" option present in the program's menu bar and then on the "Modules..." option in the menu that will open. From this, the window "Project configuration" shown in step 5 will pop up, and we press the "Add..." option, located on the right side of this window.

Afterwards, as step 6 shows, a new window will pop up, with the name "Deploy modules", listing all the currently installed modules in Modelio. If all the steps were done correctly up to this point, "SoundUML" should be on this list. We select it and press the option "Deploy in the project" in the bottom right corner. This window closes automatically and back in the "Project configuration" window, "SoundUML module" should

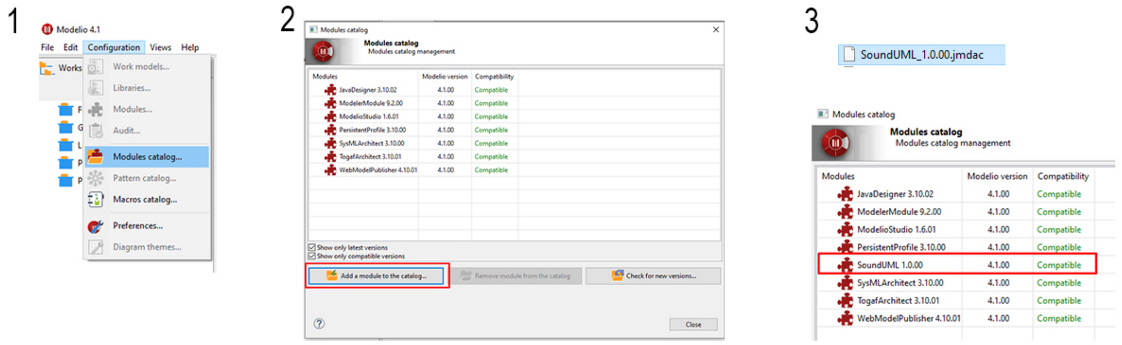


Figure 8.4: SoundUML Install Guide: Steps 1 to 3 - Adding SoundUML to the program Modelio.

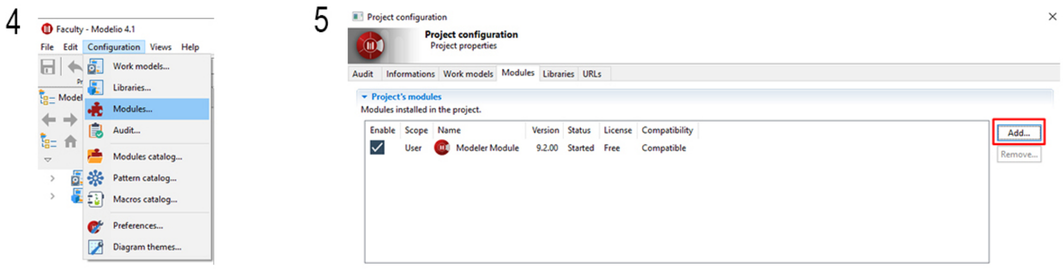


Figure 8.5: SoundUML Install Guide: Steps 4 and 5 - Deploying SoundUML to a specific user project.

appear, as illustrated in step 7. From here, we can close this window, sending us back to Modelio's main window. From here, as shown in step 8, we right-click the folder with the UML icon within the main folder of this specific project. This opens the contextual menu where SoundUML should be located. By hovering the mouse, we can select the "Read Diagram" command.

8.6 Using SoundUML

After installing the tool, this section will detail how to use the developed solution, describing its functionalities.

If the user project where we successfully installed SoundUML contains a class diagram, the tool should start the diagrammatic reading, where a window similar to the one depicted in figure 8.7 should pop up. From here, we are presented with several buttons. The "Play Sound" button allows the user to hear the sound specific to that element of UML class diagrams. As already mentioned, these sounds were implemented according to the results from chapter 7. Next, we have the "Read Message" option which essentially, reads the message that is displayed above with text, through the usage of voice synthesis. These messages are shown according to some of the patterns detected in the study from

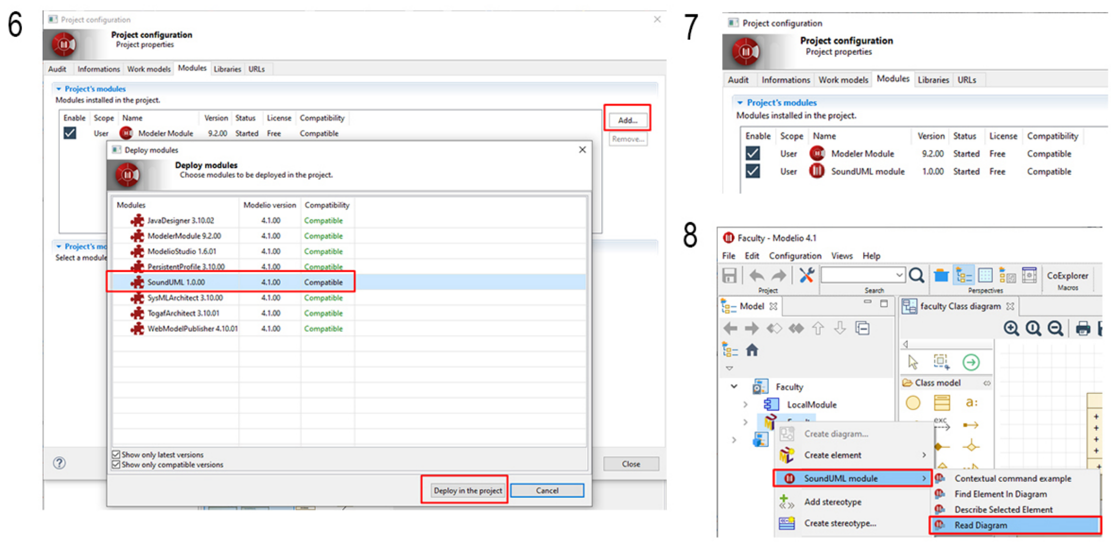


Figure 8.6: SoundUML Install Guide: Steps 6 to 8 - Deploying SoundUML to a specific user project.

chapter 6. Next, the "Reset Buttons" option, allows the users to listen to the previous options as many times as they want. Additionally, we always have the possibility to end the diagrammatic reading at any moment, go back to a previous step in the reading or continue forward to the next element. This window is navigable with the arrow keys or the tab key which highlights the current option, which can be selected with the space or the enter key.

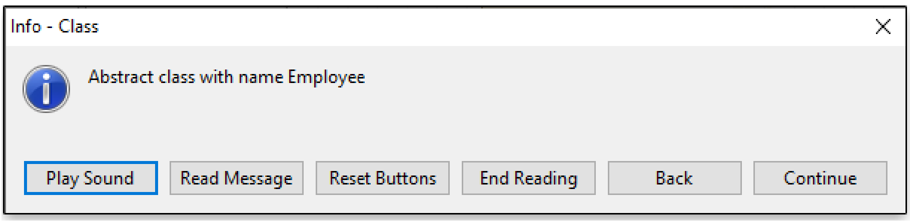


Figure 8.7: Example of a typical SoundUML window performing a diagrammatic reading, showing the reading of an abstract class.

8.7 Experimental Study: Evaluating The Diagrammatic Readings & SoundUML

This section presents the experimental procedure regarding the validation of our proposed diagrammatic reading, through the usage of our proof of concept tool, SoundUML. Furthermore, participants also evaluated SoundUML's usability and features, by answering a questionnaire.

8.7.1 Planning

The planning of this experimental study has been structured in the phases presented in the following sub-chapters.

8.7.1.1 Goals

This experiment has two main objectives, where the goal is to evaluate the tool in a quantitative and qualitative manner, respectively.

With the first main objective, the goal was to understand whether our auditory proposal for diagrammatic readings is effective and usable. In essence, we wanted to determine whether this proposed diagrammatic reading can help in one of the most important tasks of using these diagrams, which is recalling what was observed. Additionally, we compared this approach to the common one (visual) and even to a hybrid solution, in order to assess in a quantitative manner, the accuracy of answers by users. For this purpose, our prototype was used, allowing to perform an auditory reading, which is based on the proposed auditory principles and design guidelines discussed in chapter 5, together with the drawn conclusions from the experimental studies detailed in chapters 6 and 7. Participants experienced 3 different versions of diagram readings, these being the visual version, the auditory version (through our solution) and a hybrid version, which consisted of combining the two. After each diagram, they answered some questions. The results were then compared, as different subjects experienced the same diagrams through different versions.

For the second objective, we intended to qualitatively assess the usage of SoundUML itself. To this end, participants evaluated the tool's usability and features. In turn, their feedback allows us to draw certain conclusions regarding the quality of the tool, its accessibility in the sense of enabling visually impaired people to carry out certain UML modelling tasks, and determine if this contribution could complement and enrich the existing visual notation, in a manner that benefits a large and diverse population of software engineers.

8.7.1.2 Participants

Just like in the previous experimental study, it was necessary for the subjects to have some level of understanding of the modelling domain, specifically UML class diagrams. Furthermore, it was required that participants had a basic knowledge of English, since the questions were asked in English and there were some optional open-ended questions regarding feedback. The study was conducted in person and the participants were chosen through convenience sampling, as we recruited the people who were available, with these being doctorate, master and undergraduate students in computer engineering.

To better understand the profile of the participants and their level of knowledge, the following questions were asked in the form, thus allowing us to get a more comprehensive picture of who answered the questions:

- Gender
- Are you hard of hearing?
- How do you consider your level of understanding of UML?
- When was your last contact with UML?
- In what context do you use UML?

In total, this experimental study had 11 participants.

8.7.1.3 Tasks

Firstly, participants were asked to read the provided document containing a brief introductory text, along with the tasks to be carried out. Additionally, in order to refresh their knowledge of UML, a brief summary of class diagram notations was given on the pages with a yellow background, similar to the one from chapter 6. The subjects could consult this summary at any moment during the evaluation sessions. This document can be viewed in Appendix C.3.

Following the reading of the aforementioned document, participants were asked to fill in the form regarding personal information, intended to briefly ascertain the participants' level of understanding of the UML language.

Next, a total of 3 Class diagrams were presented through Modelio, each in a unique manner: visual, auditory (our tool, SoundUML), and a combination of both. The order in which the diagrams were presented, as well as the unique manner varied from subject to subject. Participants had 5 minutes to analyse a given diagram and afterwards, they were asked to answer a form with a few simple, multiple-choice questions regarding that specific diagram. As the order of the diagrams changed from person to person, they had to first select which one of the versions of that specific diagram they experienced (visual, auditory or hybrid). After the questions for that specific diagram were submitted, another diagram was shown in a different manner.

After the subjects answered the questions for each of the 3 diagrams, they were asked to use our developed tool, SoundUML, a little further, and test its functionalities to the fullest, followed by answering the form regarding the tool's usability and feedback.

8.7.1.4 Defining The Hypothesis

The hypotheses regarding the study in question are presented. The first set of hypotheses concerns the use and effectiveness of our proposed auditory diagrammatic reading from the user's point of view, in one of the most important tasks of using these diagrams, which is recalling what was observed.

The second set is centred on usability aspects and the quality of user interaction in our constructed prototype, SoundUML. Concretely, this set of hypotheses is concerned with ascertaining whether the participants consider that the tool provides any added value, or not, in regards to the topics of accessibility, in the sense of enabling visually impaired people to carry out certain UML modelling tasks, while complementing and enriching the existing visual notation, in a manner that benefits a large and diverse population of software engineers.

Consequently, this study seeks to reject the null hypotheses.

Defining the first set of hypotheses:

$\mathbf{H_{0DiagrammaticReadingT}}$: Our proposed auditory diagrammatic reading is not effective or usable, from the user's point of view, in one of the most important tasks of using UML class diagrams, which is recalling what was observed.

$\mathbf{H_{1DiagrammaticReadingT}}$: Our proposed auditory diagrammatic reading is effective and usable, from the user's point of view, in one of the most important tasks of using UML class diagrams, which is recalling what was observed.

Defining a second set of hypotheses:

$\mathbf{H_{0SoundUMLT}}$: The participants consider that our constructed tool, SoundUML, does not provide any added value, in regards to the topics of accessibility, in the sense of enabling visually impaired people to carry out certain UML modelling tasks, and enriching the existing visual notation.

$\mathbf{H_{1SoundUMLT}}$: The participants consider that our constructed tool, SoundUML, provides added value, in regards to the topics of accessibility, in the sense of enabling visually impaired people to carry out certain UML modelling tasks, and enriching the existing visual notation.

8.7.1.5 Preparing The Evaluation Sessions

To conduct this study, firstly it was necessary to install Modelio, along with our tool, and ensure that these were properly working on the computer that would be used for the experiment.

Since we intended to compare the reading effectiveness between 3 different versions of UML class diagrams (visual, auditory and hybrid), 3 class diagrams were selected for this experiment. This decision was taken because if only 1 diagram was used, participants would answer questions from the same diagram more than once and could logically, obtain better results in successive attempts, since there would be a greater familiarity from repeated viewings of the same diagram, even if those viewings would occur through different versions of the same diagram.

We then sought to choose diagrams that had some variety in terms of the elements present in each, and that had a similar degree of complexity between them. This was due to the fact that it was decided that along with the order of the diagrams themselves, the versions that different participants would observe, would vary as well. Thus, it was crucial that the experience was comparable and fair in all cases. For example, it would not make sense for certain participants to view a simpler diagram but use our tool on another more complex one, or vice versa. Additionally, it was intended that the selected diagrams would have a certain degree of complexity, but not be excessively elaborate, as we wanted the diagrams to be manageable for the participants' memory, given the type of questions that would be asked, and that the duration of the experiment would not become excessively long.

Once the diagrams were chosen, it was necessary to adapt and create them in the Modelio program, therefore making it possible for our solution to perform the diagrammatic reading, given that SoundUML is a module for Modelio. The visual versions of the diagrams, as they were shown to the participants through Modelio, can be observed in the Appendix C.2.

We also planned the type of questions to be answered by the participants, both concerning the diagrams and the comparison between the different versions, as well as our solution's usability and general feedback. Regarding the questions about the diagrams, these were thought out to cover all the existing elements in the UML class diagrams, and that at the same time, they made sense for each of the diagrams. We always tried to have reasonable judgement about the level of specificity of what we asked and that the questions were reasonable in terms of difficulty, since we would be asking the participants to make use of their memory and recall certain details from the observed diagrams. These multiple-choice questions were created with 3 options each. In the creation of the incorrect options, we tried to ensure that they made sense and were consistent with that specific diagram, in order to minimise the possibility of the participants arriving at the right answer through the process of eliminating the incorrect options that did not make sense in the context of the diagram. Finally, the usability questions were carried out through the System Usability Scale (SUS), which is a 10-item questionnaire. With the general feedback, we sought to more precisely understand the opinion of the participants regarding certain qualities of the tool, such as its accessibility, aspects to improve, etc.

A document was then prepared to be shown to the participants, containing a brief introductory text, along with the tasks to be carried out. In this document, some support material was included, similar to the one created for the diagrammatic reading study of chapter 6, in order to refresh the subjects' memory regarding the notation of UML class diagrams. We created the forms in Google Forms, with the questions to be asked for each diagram, together with the usability and general feedback questions aimed at the tool. As it was decided that the order in which the diagrams would be shown to the participants would differ, separate forms had to be created, so that it would be possible to swap the order in which the forms would be shown to the participants. Thus, a total of 4 forms were created, 1 for each diagram, and another for the questions related to the usability and general feedback of the tool, as well as the participants' personal information. In each of the 3 forms, corresponding to the 3 chosen diagrams, there was a first question related to how the participant that is answering the questions, experienced that particular diagram. This allowed to separate the answers given by the subjects, according to how a particular person experienced each of the diagrams, thus allowing for a direct comparison between the 3 different reading versions. For example, if person A experienced the visual version of diagram 1 and person B, with the same diagram, experienced the sound version (with our tool), they each answered the same questions, but the answers were separated according to the version experienced.

All the forms that participants answered, can be consulted in Appendix C.

8.7.1.6 Experimental Material

The evaluation sessions took place in person, in a quiet room, allowing the participants to fully concentrate and listen to the sounds, in the case of the auditory and hybrid diagrammatic readings. In order to adequately perform this study, it was necessary to have a computer with an internet connection to submit the forms with the answers given by the participants. Modelio was also required to be installed on the computer, together with our tool, SoundUML, and the diagrams that were going to be shown to the participants. The forms that the participants were going to answer, along with the document prepared for them, were also necessary to conduct the experimental sessions. Additionally, a device capable of producing sound output was needed, so that the sounds could be heard. In these sessions, we used the computer speakers. Finally, a stopwatch was also needed to keep track of how much time the participants had left to observe each diagram.

8.7.2 Procedure Of The Evaluation Sessions

This experimental study took place in person, on the campus of NOVA School of Science and Technology, where a classroom was reserved for this purpose, in the Department of Computer Science. Here, we began by setting up the necessary material for the experiment: a computer with an internet connection, Modelio and SoundUML with the diagrams ready to be shown in the program, and a stopwatch. On the computer, the document prepared for the participants and forms were also opened and left ready, together with Modelio.

As previously mentioned, this study followed a "pseudo-random" order, so that different combinations and versions of the diagrams were evaluated, helping to prevent skewed results. Taking into consideration that there is a possibility of subjects learning throughout the experiment, it would not make sense to always start the evaluations with the same diagram and the same version. Therefore, the diagrams were shown in different orders and in different versions to different participants, so that all cases could be tested in the same way. This is known as counterbalancing, which refers to testing different subjects in different orders, serving to guard against "order effects", a phenomenon where participants' responses are affected by the order to which they were exposed.

In this manner, the assignment and order of execution of the diagrammatic readings, along with the intended versions for a specific participant, were ensured by us at the moment of carrying out the experiment. There are 3 diagrams: Faculty (A), Library (B) and GUI (C), and 3 distinct ways of observing each of the diagrams: Visual (1), Sound (2) and Hybrid (3). Thus, each diagram can be depicted as follows: A1, A2, A3; B1, B2, B3; C1, C2, C3. However, it was decided that the hybrid version would always be shown last, so the visual versions of the diagrams would be shown first and the sound versions second, or vice versa. This produces 12 possible combinations (3x2x2) for the order in which the diagrams and their versions were presented, which are as presented in table 8.1.

Table 8.1: Table with the 12 possible combinations of the order in which the diagrams were presented.

1st	2nd	3rd
(A1) Faculty Visual	(B2) Library Sound	(C3) GUI Hybrid
(A1) Faculty Visual	(C2) GUI Sound	(B3) Library Hybrid
(B1) Library Visual	(A2) Faculty Sound	(C3) GUI Hybrid
(B1) Library Visual	(C2) GUI Sound	(A3) Faculty Hybrid
(C1) GUI Visual	(A2) Faculty Sound	(B3) Library Hybrid
(C1) GUI Visual	(B2) Library Sound	(A3) Faculty Sound
(A2) Faculty Sound	(B1) Library Visual	(C3) GUI Hybrid
(A2) Faculty Sound	(C1) GUI Visual	(B3) Library Hybrid
(B2) Library Sound	(A1) Faculty Visual	(C3) GUI Hybrid
(B2) Library Sound	(C1) GUI Visual	(A3) Faculty Hybrid
(C2) GUI Sound	(A1) Faculty Visual	(B3) Library Hybrid
(C2) GUI Sound	(B1) Library Visual	(A3) Faculty Hybrid

At the beginning of each session, held with only 1 participant at a time, one of these combinations was assigned to that specific participant. Next, the experiment was explained to the subjects while they were reading the created document for the experiment, which contained a brief introductory text, along with the tasks to be carried out, and the support material regarding UML class notations. The subjects were given a few minutes to read it. Afterwards, we answered any questions they asked. The participants could consult this document at any time during the experiment.

Following this, respondents were asked to answer the form with the questions concerning their personal information, which allows us to obtain a better profile of those who participated in the study. Then, 3 diagrams were shown, one by one, according to the combination assigned to that particular participant. Depending on the attributed version of a diagram, the manner in which it was presented, would differ. In the case of the visual version, the diagram was shown through Modelio's interface. In the case of the auditory reading, carried out strictly through our tool, we minimised Modelio's window, and only kept the navigation window of SoundUML on the screen. In the hybrid version, both the interfaces of Modelio and our tool were displayed. For each diagram, the participants were given 5 minutes to perform the reading. Afterwards, they were asked to answer the questions on the form relating to the specific diagram they had just experienced. During this task, they were not allowed to perform the diagrammatic reading again, relying only on their ability to recall the diagram, by retrieving details from their memory. These steps were repeated for all 3 diagrams.

Finally, after the questions concerning the 3 diagrams were answered and the combination assigned to that particular subject was satisfied, we conducted a brief showcase of our tool. During this demonstration, we detailed some of its features, while giving a few more minutes for the subjects to use the tool and to further explore the diagrams, with emphasis on the diagram that they had observed only through its visual version (as they did not have the opportunity to use the tool for that specific diagram). Subsequently, the respondents were asked to answer the form with the questions related to the tool's usability and general feedback.

Each experimental session was conducted with one person at a time, taking between 30 and 45 minutes, depending on how quickly participants responded.

8.7.3 Results

After the evaluation sessions, we obtained the results presented below. This section is divided into Personal Information and the results for each of the three diagrams. Afterwards, we present the results of the evaluation of SoundUML, concerning its usability, general feedback and accessibility.

8.7.3.1 Personal Information

In this subsection, the personal information of the subjects is presented, allowing us to obtain a better profile of the people who answered the questions. 11 participants responded to the form, of which 18.2% were female and 81.8% were male.

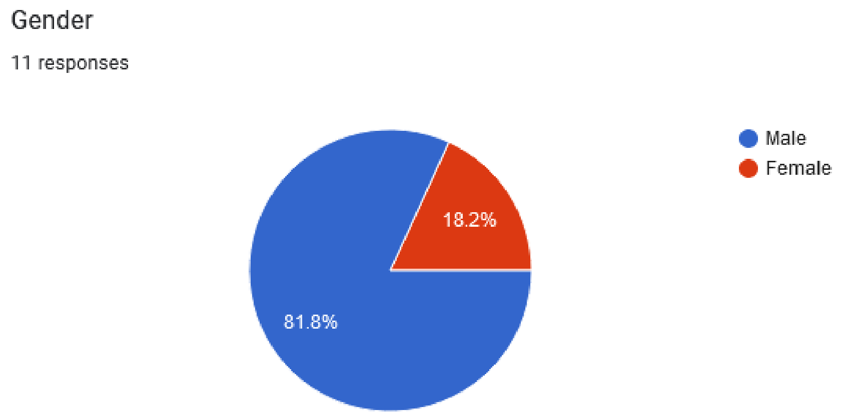


Figure 8.8: Results regarding the participants' gender.

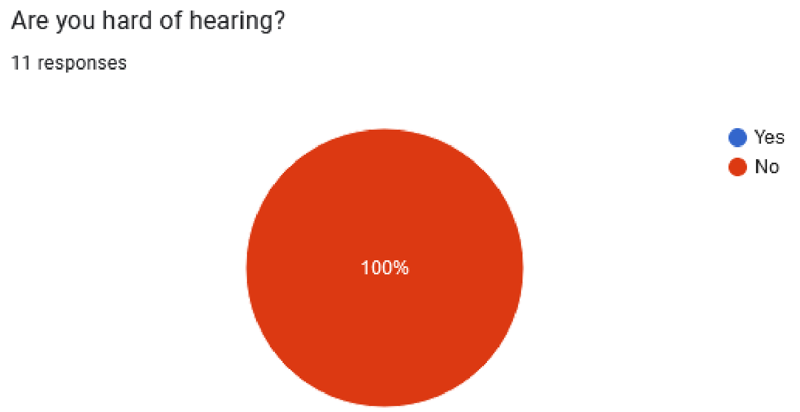


Figure 8.9: Results regarding the participants' hearing condition.

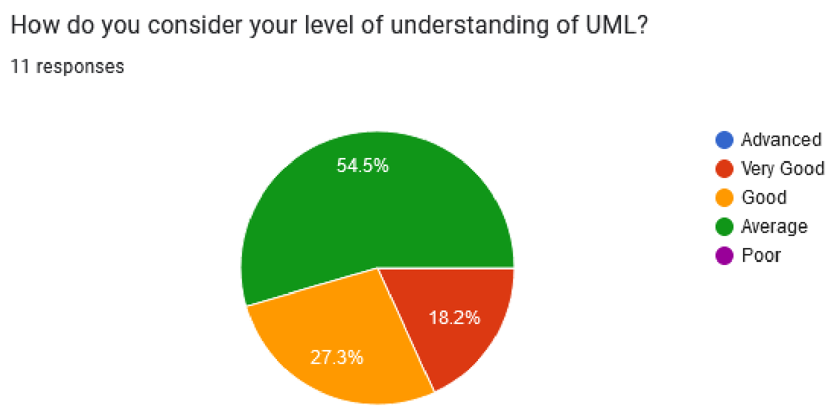


Figure 8.10: Results regarding the participants' level of understanding of UML.

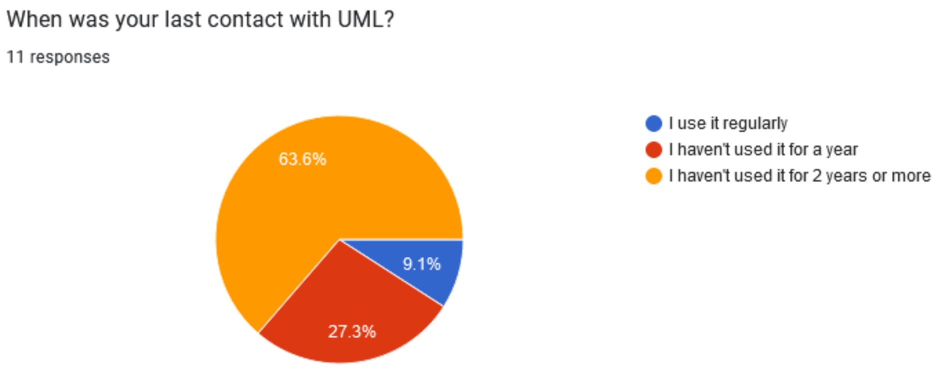


Figure 8.11: Results regarding the participants' last contact with UML.

8.7.3.2 Diagram A: Faculty

The following section presents the results of the questions regarding the diagram Faculty. The visual version of this diagram, as it was shown to the participants, can be consulted in the Appendix C.2.

Figure 8.12 illustrates how the different participants experienced each version of this diagram.

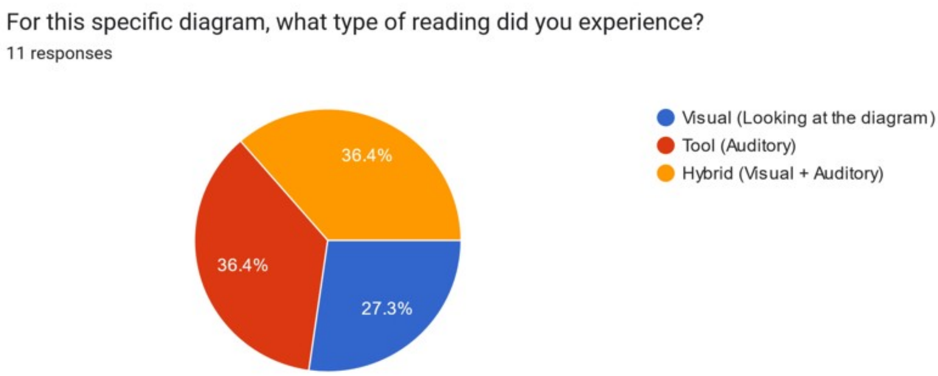


Figure 8.12: Breakdown of how the participants experienced each version of the Faculty diagram.

Figure 8.13 shows the results of **Question 1: How many inheritance relationships exist in this diagram?**

This question was aimed at the Inheritance element of UML class Diagrams, and had as possible answers "2", "3" and "4", with the right answer being option "3". Table 8.2 contains the number of answers for each version, in each of the possible options, with the correct answer highlighted in yellow.

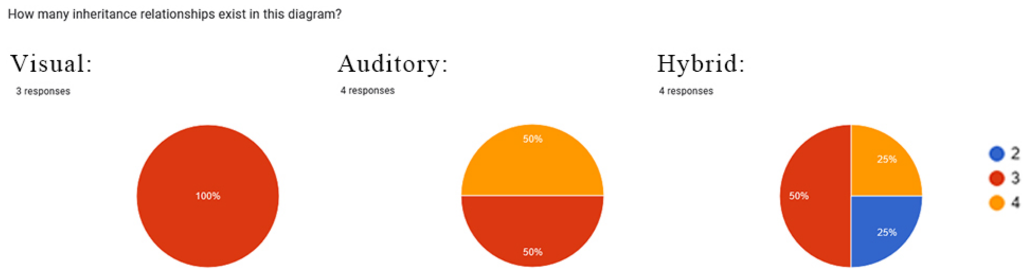


Figure 8.13: Results of Question 1 From The Faculty Diagram, for each of the versions

Table 8.2: Table with the results of Question 1 From The Faculty Diagram

Faculty Diagram Question 1	Visual	Auditory	Hybrid
2	0	0	1
3	3	2	2
4	0	2	1
Total Number Of Answers	3	4	4

In figure 8.14 the results of **Question 02: What is the type of relationship between classes Faculty and Institute?** are represented.

This question was aimed at the Composition element of UML class Diagrams, and had as possible answers "Association", "Inheritance" and "Composition", with the right answer being the option "Composition". Table 8.3 contains the number of answers for each version, in each of the possible options, with the correct answer highlighted in yellow.

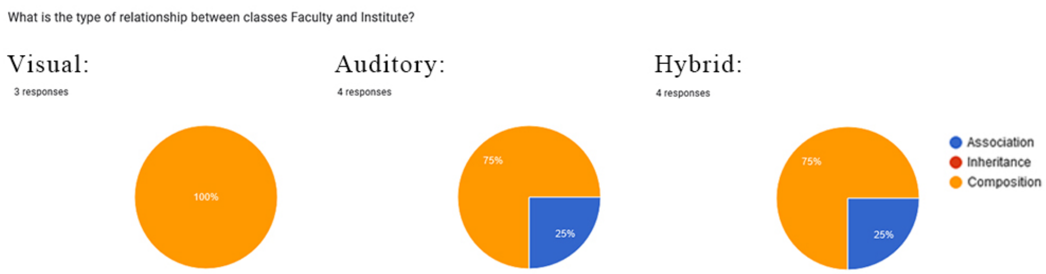


Figure 8.14: Results of Question 2 From The Faculty Diagram, for each of the versions

Table 8.3: Table with the results of Question 2 From The Faculty Diagram

Faculty Diagram Question 2	Visual	Auditory	Hybrid
Association	0	1	1
Inheritance	0	0	0
Composition	3	3	3
Total Number Of Answers	3	4	4

Figure 8.15 depicts the results of **Question 03: What was the name of the only association class in the diagram?**

This question was aimed at the Composition element of UML class Diagrams, and had as possible answers "Participation", "Course" and "Project", with the right answer being the option "Participation". Table 8.4 contains the number of answers for each version, in each of the possible options, with the correct answer highlighted in yellow.

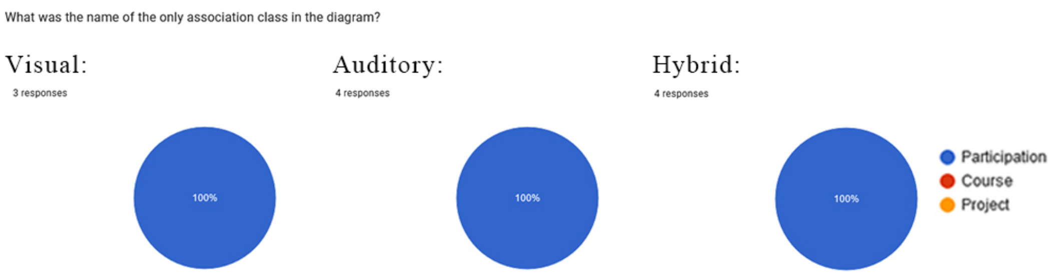


Figure 8.15: Results of Question 3 From The Faculty Diagram, for each of the versions

Table 8.4: Table with the results of Question 3 From The Faculty Diagram

Faculty Diagram Question 3	Visual	Auditory	Hybrid
Participation	3	4	4
Course	0	0	0
Project	0	0	0
Total Number Of Answers	3	4	4

8.7.3.3 Diagram B: Library

The following section presents the results of the questions regarding the diagram Library. The visual version of this diagram, as it was shown to the participants, can be consulted in the Appendix C.2.

Figure 8.16 represents how the different participants experienced each version of this diagram.

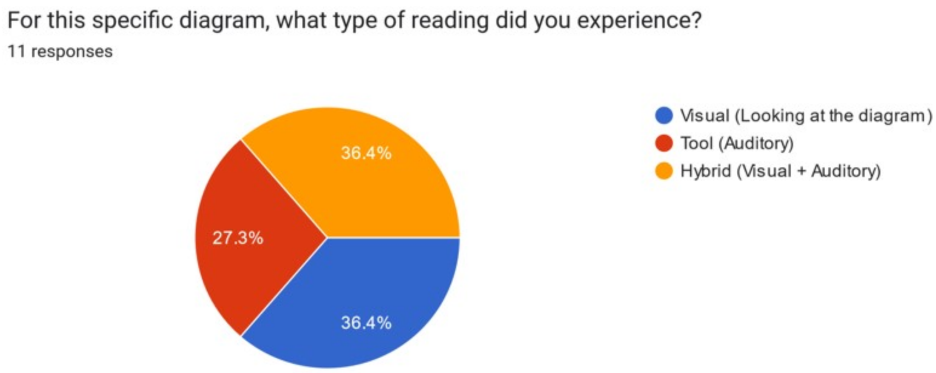


Figure 8.16: Breakdown of how the participants experienced each version of the Library diagram.

Figure 8.17 illustrates the results of **Question 01: How many association relationships exist between classes Book Item and Account?**

This question was aimed at the Association element of UML class Diagrams, and had as possible answers "1", "2" and "3", with the right answer being the option "2". Table 8.5 contains the number of answers for each version, in each of the possible options, with the correct answer highlighted in yellow.

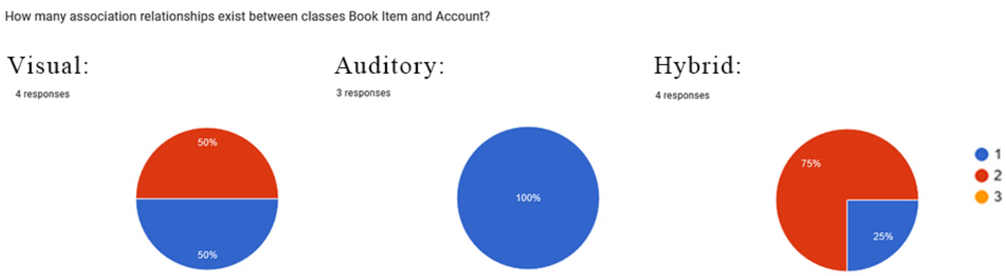


Figure 8.17: Results of Question 1 From The Library Diagram, for each of the versions

Table 8.5: Table with the results of Question 1 From The Library Diagram

Library Diagram Question 1	Visual	Auditory	Hybrid
1	2	3	1
2	2	0	3
3	0	0	0
Total Number Of Answers	4	3	4

Figure 8.18 depicts the results of **Question 02: What class implements the interfaces Manage and Search?**

This question was aimed at the Implementation element of UML class Diagrams, and had as possible answers "Catalog", "Account" and "Patron", with the right answer being the option "Catalog". Table 8.6 contains the number of answers for each version, in each of the possible options, with the correct answer highlighted in yellow.

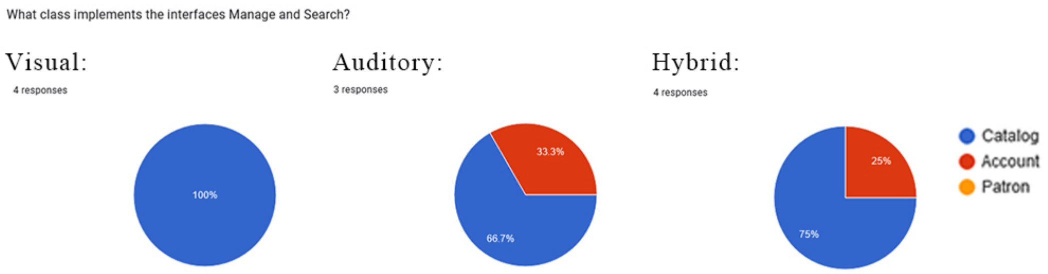


Figure 8.18: Results of Question 2 From The Library Diagram, for each of the versions

Table 8.6: Table with the results of Question 2 From The Library Diagram

Library Diagram Question 2	**Visual**	**Auditory**	**Hybrid**
Catalog	4	2	3
Account	0	1	1
Patron	0	0	0
Total Number Of Answers	4	3	4

Figure 8.19 illustrates the results of **Question 03: What two classes have a dependency to the interfaces in the diagram? (The interfaces are Manage and Search)**

This question was aimed at the Dependency element of UML class Diagrams, and had as possible answers "Author and Account", "Librarian and Patron" and "Book Item and Patron", with the right answer being the option "Librarian and Patron". Table 8.7 contains the number of answers for each version, in each of the possible options, with the correct answer highlighted in yellow.

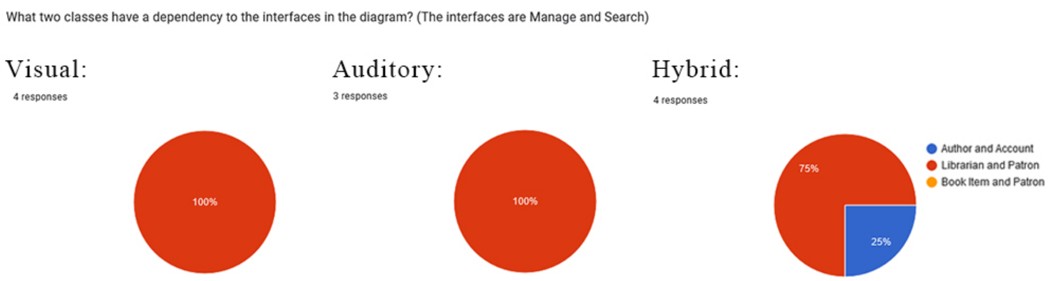


Figure 8.19: Results of Question 3 From The Library Diagram, for each of the versions

Table 8.7: Table with the results of Question 3 From The Library Diagram

Library Diagram Question 3	Visual	Auditory	Hybrid
Author and Account	0	0	1
Librarian and Patron	4	3	3
Book Item and Patron	0	0	0
Total Number Of Answers	4	3	4

Figure 8.20 delineates the results of **Question 04: Which of these is NOT an attribute from class Book?**

This question was aimed at the Attribute element of UML class Diagrams, and had as possible answers "ISBN", "releaseDate" and "publisher", with the right answer being the option "releaseDate". Table 8.8 contains the number of answers for each version, in each of the possible options, with the correct answer highlighted in yellow.

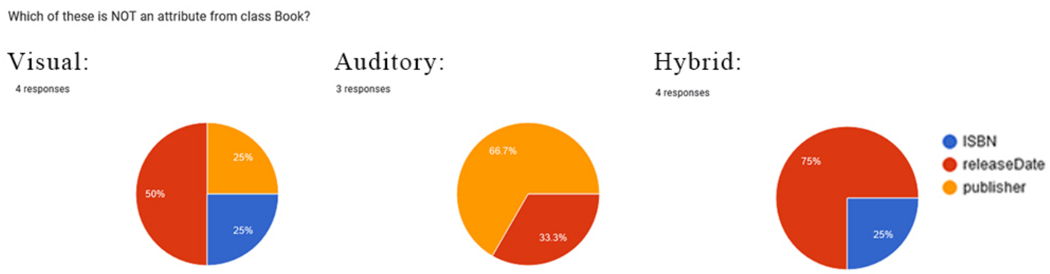


Figure 8.20: Results of Question 4 From The Library Diagram, for each of the versions

Table 8.8: Table with the results of Question 4 From The Library Diagram

Library Diagram Question 4	Visual	Auditory	Hybrid
ISBN	1	0	1
releaseDate	2	1	3
publisher	1	2	0
Total Number Of Answers	4	3	4

8.7.3.4 Diagram C: GUI

This following section presents the results of the questions regarding the diagram GUI. The visual version of this diagram, as it was shown to the participants, can be consulted in the Appendix C.2.

Figure 8.21 illustrates how the different participants experienced each version of this diagram.

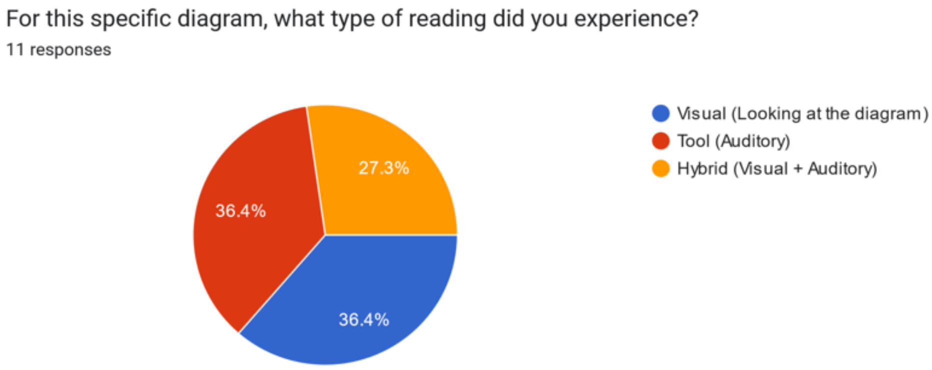


Figure 8.21: Breakdown of how the participants experienced each version of the GUI diagram.

Figure 8.22 shows the results of **Question 01: What is the type of relationship between classes Shape and Window?**

This question was aimed at the Aggregation element of UML class Diagrams, and had as possible answers "Aggregation", "Dependency" and "Composition", with the right answer being the option "Aggregation". Table 8.9 contains the number of answers for each version, in each of the possible options, with the correct answer highlighted in yellow.

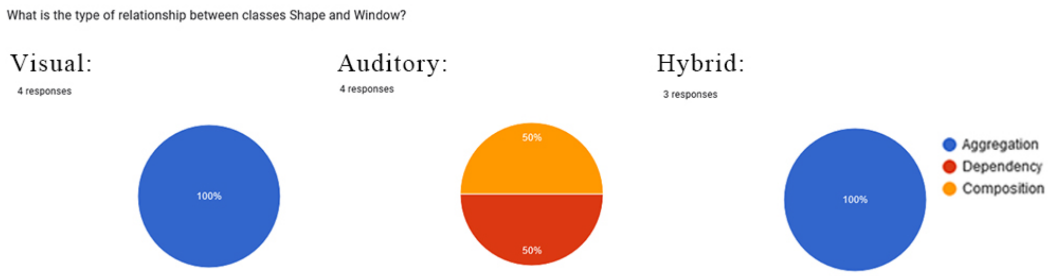


Figure 8.22: Results of Question 1 From The GUI Diagram, for each of the versions

Table 8.9: Table with the results of Question 1 From The GUI Diagram

GUI Diagram Question 1	Visual	Auditory	Hybrid
Aggregation	4	0	3
Dependency	0	2	0
Composition	0	2	0
Total Number Of Answers	4	4	3

Figure 8.23 depicts the results of **Question 02: Which of these classes does NOT exist in the diagram?**

This question was aimed at the Class element of UML class Diagrams, and had as possible answers "Event", "DrawingContext" and "Triangle", with the right answer being the option "Triangle". Table 8.10 contains the number of answers for each version, in each of the possible options, with the correct answer highlighted in yellow.

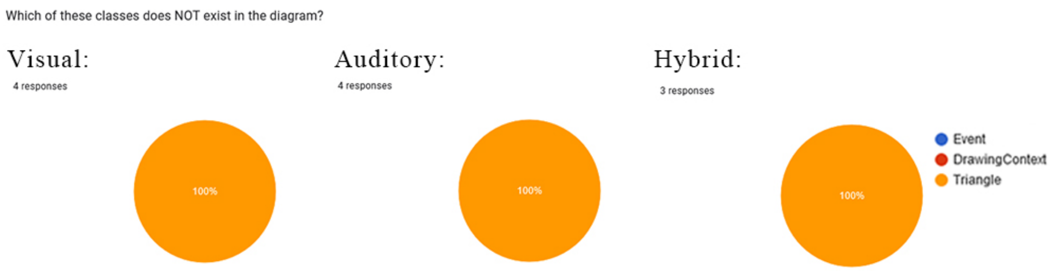


Figure 8.23: Results of Question 2 From The GUI Diagram, for each of the versions

Table 8.10: Table with the results of Question 2 From The GUI Diagram

GUI Diagram Question 2	Visual	Auditory	Hybrid
Event	0	0	0
DrawingContext	0	0	0
Triangle	4	4	3
Total Number Of Answers	4	4	3

Figure 8.24 represents the results of **Question 03: Which of these IS the name of an operation in class Circle?**

This question was aimed at the Operation element of UML class Diagrams, and had as possible answers "calcArc", "setPoint" and "setCenter", with the right answer being the option "setCenter". Table 8.11 contains the number of answers for each version, in each of the possible options, with the correct answer highlighted in yellow.

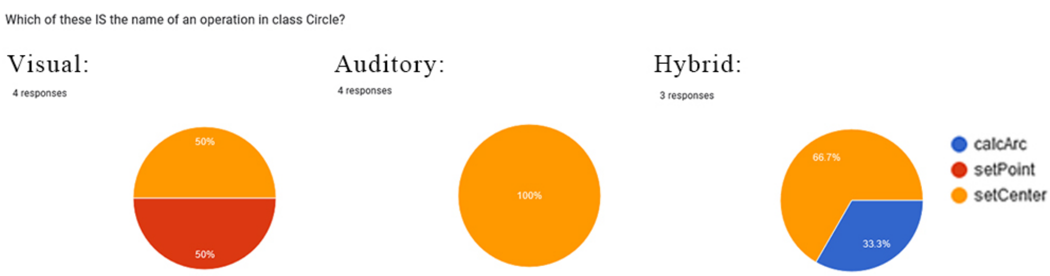


Figure 8.24: Results of Question 3 From The GUI Diagram, for each of the versions

Table 8.11: Table with the results of Question 3 From The GUI Diagram

GUI Diagram Question 3	Visual	Auditory	Hybrid
calcArc	0	0	1
setPoint	2	0	0
setCenter	2	4	2
Total Number Of Answers	4	4	3

8.7.4 Results On The Evaluation Of SoundUML

This section is concerned with the results related to the evaluation of our tool, being divided into three: Usability, where we used SUS; General Feedback, with the results of the overall assessment regarding the tool and the diagrammatic readings it performed; Accessibility, concerning the participants' opinion on how the tool could contribute to the inclusion of visually impaired people, in UML modelling activities.

8.7.4.1 Usability (System Usability Scale)

The following section presents the data resulting from participants' answers to the 10 questions concerning the System Usability Scale, giving an overview of users' evaluations of usability in the SoundUML tool. In the following graphs, the number 1 corresponds to "strongly disagree"and number 5 to "strongly agree".

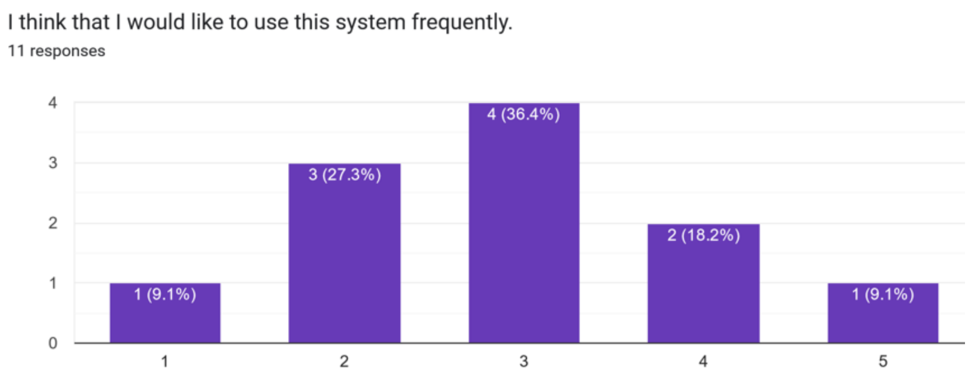


Figure 8.25: Results of the first SUS question concerning the *SoundUML* tool.

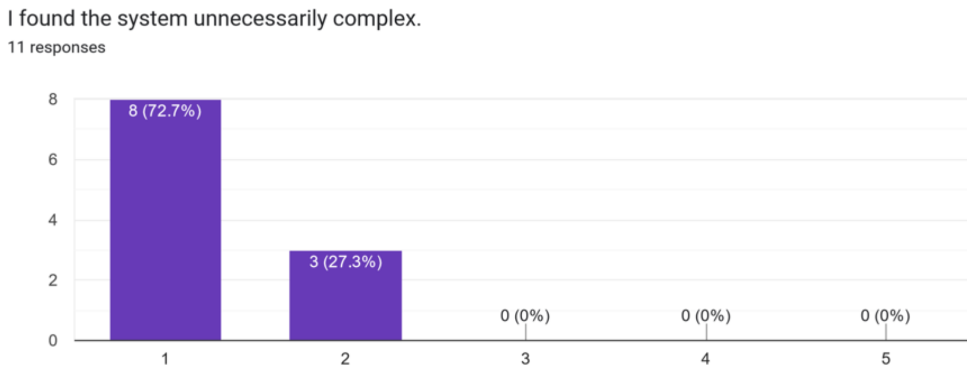


Figure 8.26: Results of the second SUS question concerning the *SoundUML* tool.

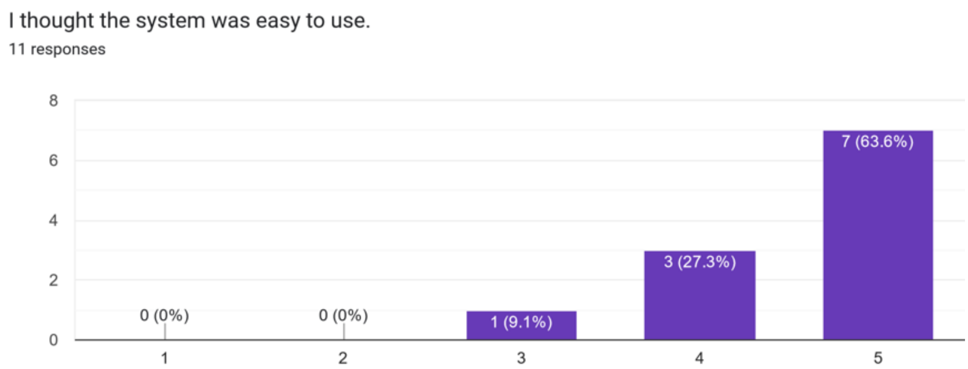


Figure 8.27: Results of the third SUS question concerning the *SoundUML* tool.

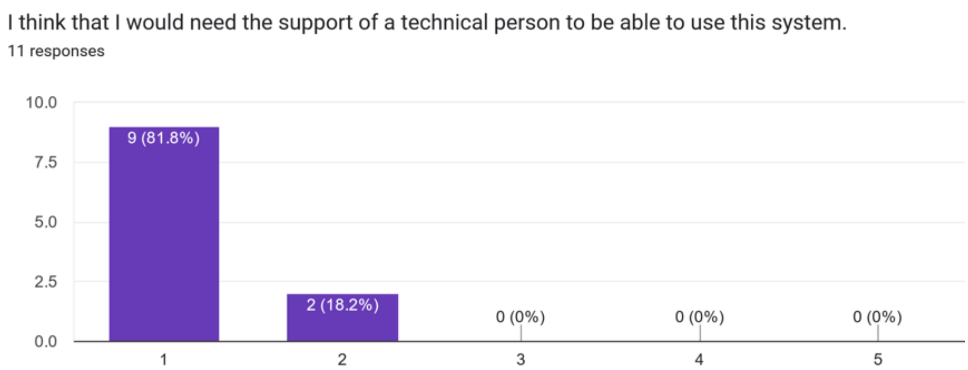


Figure 8.28: Results of the fourth SUS question concerning the *SoundUML* tool.

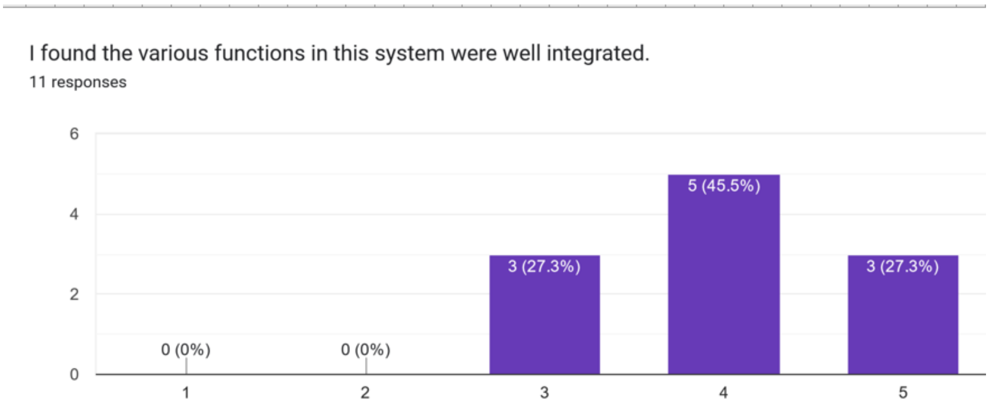


Figure 8.29: Results of the fifth SUS question concerning the *SoundUML* tool.

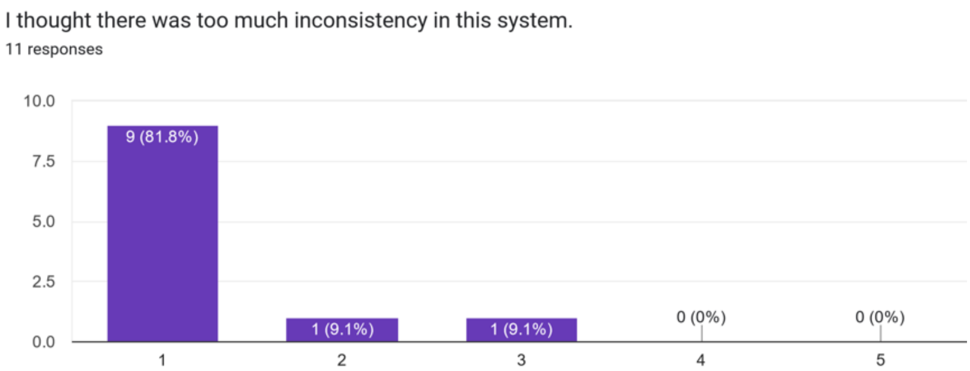


Figure 8.30: Results of the sixth SUS question concerning the *SoundUML* tool.

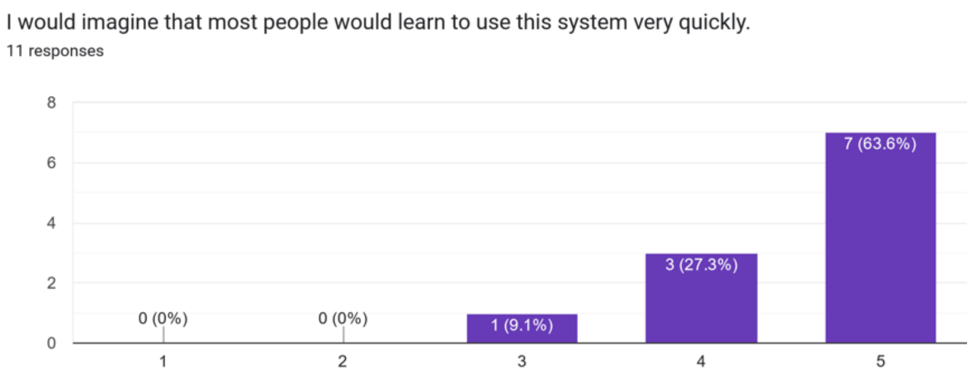


Figure 8.31: Results of the seventh SUS question concerning the *SoundUML* tool.

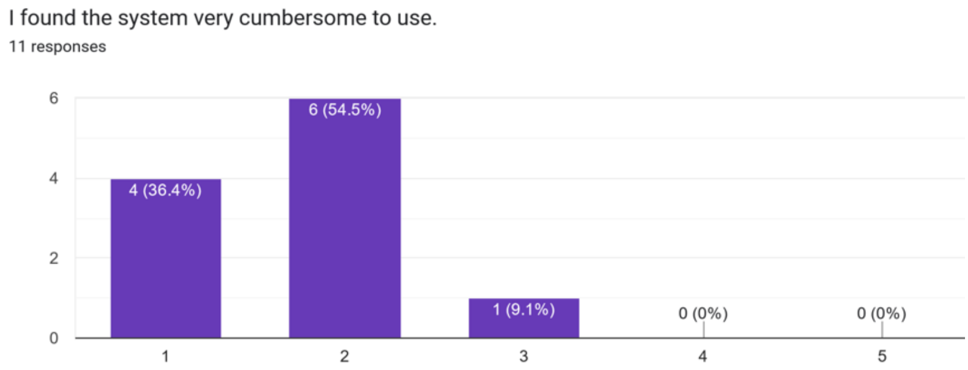


Figure 8.32: Results of the eighth SUS question concerning the *SoundUML* tool.

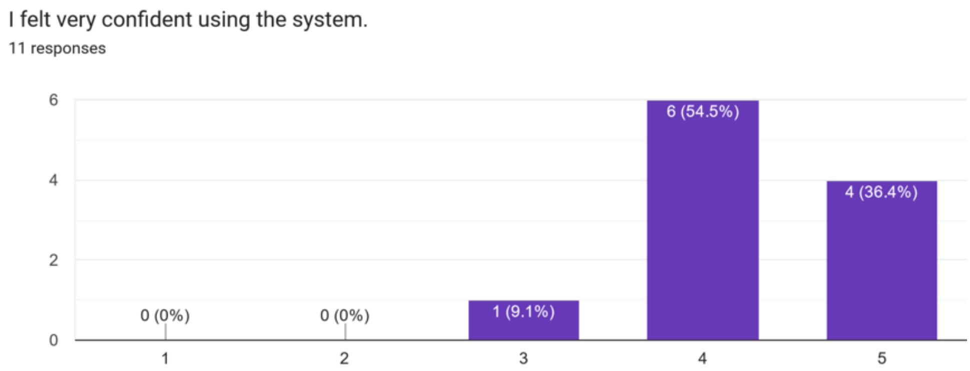


Figure 8.33: Results of the ninth SUS question concerning the *SoundUML* tool.

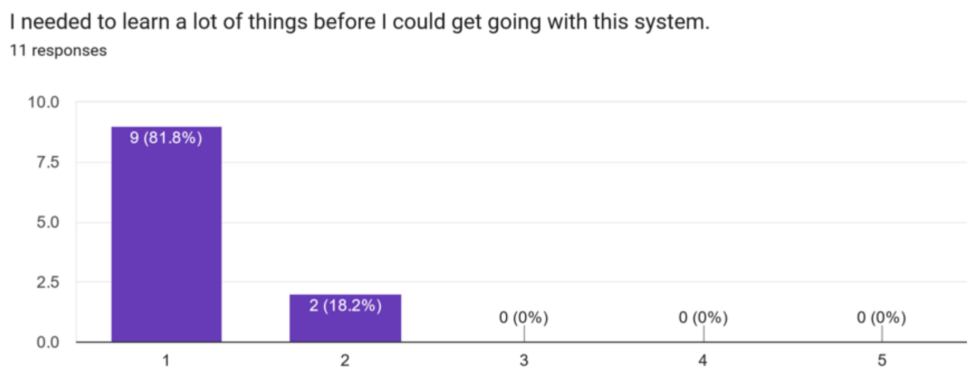


Figure 8.34: Results of the tenth SUS question concerning the *SoundUML* tool.

8.7.4.2 General Feedback

This section details the general feedback and overall assessment regarding the tool, along with the readings it performed.

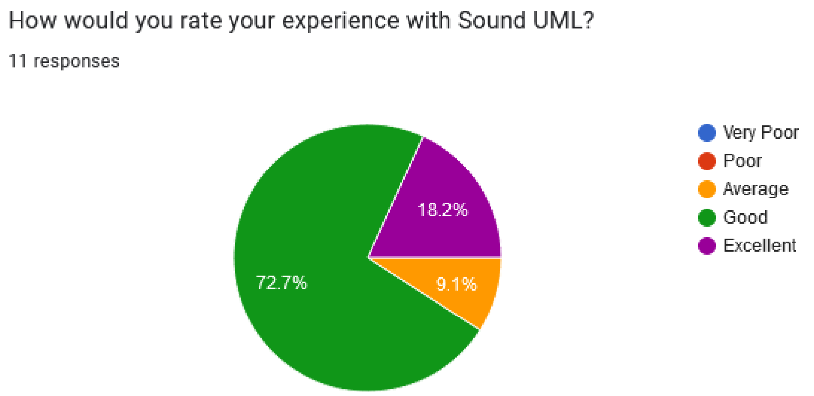


Figure 8.35: Results of how participants rated their experience with the *SoundUML* tool.

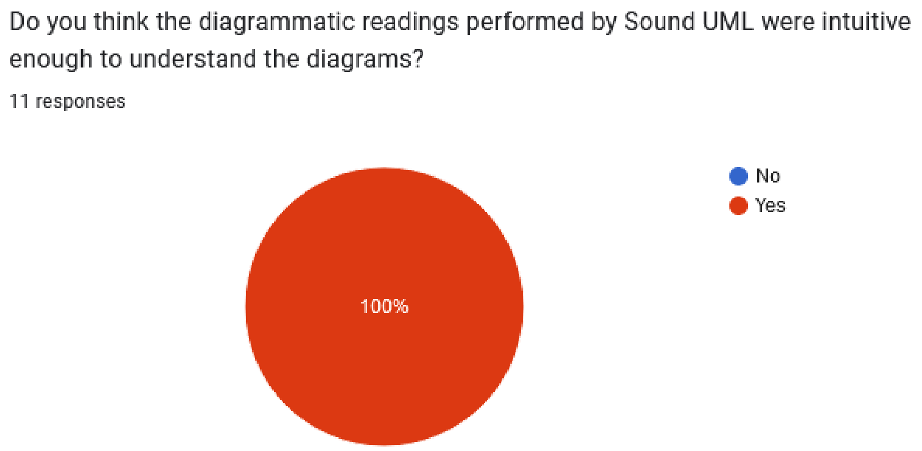


Figure 8.36: Results on whether participants thought that the readings performed by the tool were intuitive enough to understand the diagrams.

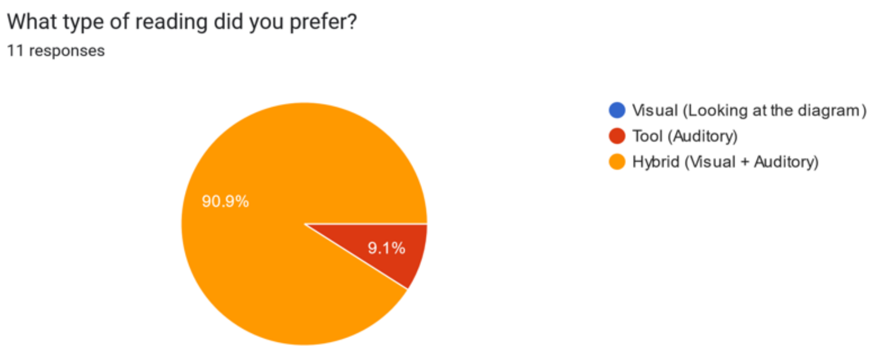


Figure 8.37: Results on the type of reading preferred by the participants.

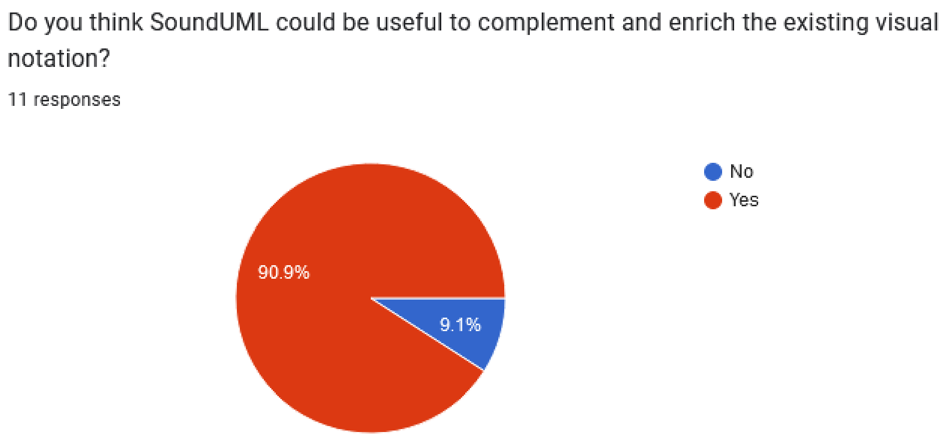


Figure 8.38: Results regarding the participants' opinion on whether SoundUML could be useful to complement and enrich the existing visual notation.

With this study, it was possible to collect the **strongest aspects** of the soundUML tool, as perceived by the participants, which are as follows:

- Intuitive sounds chosen to represent all the different types of elements;
- Intuitive order by which the elements are presented;
- Step-by-step-like instructions that help the user envision the diagram in their head;
- Different ways of using the tool, such as reading the text of the elements and/or the sound coming from the catalogue, along with a standalone version or hybrid;
- Easier to understand the semantics of specific relationships for someone not completely familiar with class diagram notation;
- The relationships between classes become very explicit using Sound UML, making it a good tool to assist in teaching UML as well.

The **weakest aspects** of SoundUML as identified by the participants were collected and are as follows:

- Lack of overall, high-level, view of the diagram;
- Inability to choose a particular element to be read;
- The soundUML UI window returns to the centre of the screen;
- The text-to-speech temporarily freezes the UI;
- The TTS voice is very robotic/unfriendly.

Participants **suggested some changes** to the SoundUML tool. These are as follows:

- Read a specific element of the diagram by selecting it;
- When using the hybrid approach, visually highlight the element that is being shown in the prompt of the SoundUML;
- The UI window opening in the same position as the previous window;
- Add a mechanism to let the users know how much of the diagram has been covered at any moment;
- More natural/empathetic TTS voice.

8.7.4.3 Accessibility

This section presents the gathered information regarding the participants' opinions on the topics of accessibility concerning the SoundUML tool.

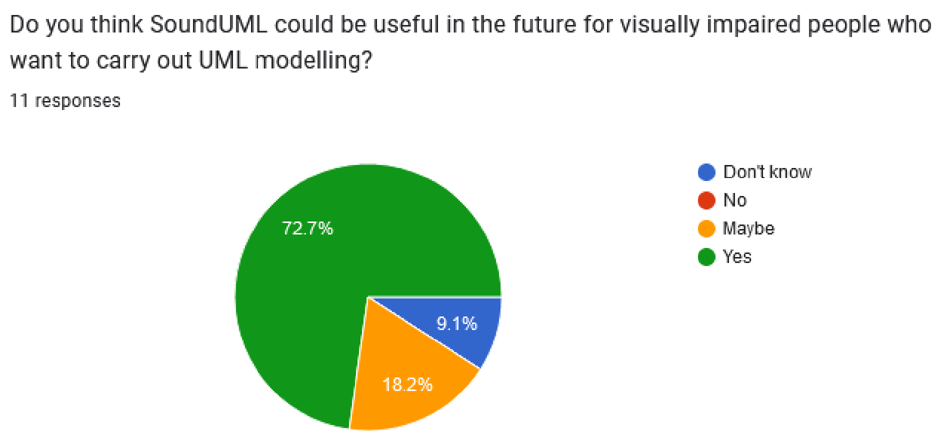


Figure 8.39: Results concerning the participants' opinion on if the SoundUML tool could be useful for people with visual impairments who want to carry out UML modelling.

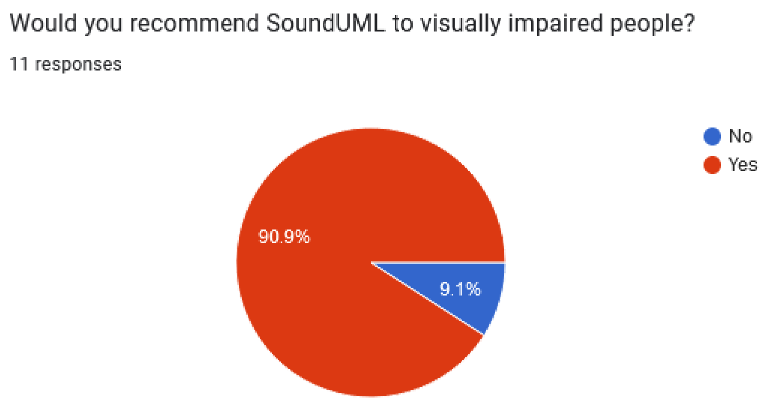


Figure 8.40: Results regarding whether participants would recommend the SoundUML tool to visually impaired people.

8.7.5 Discussion Of Results

The results of this experimental study are analysed and discussed in the following section, which is further divided into 5 subsections, which are: Personal Information, Diagrammatic Readings and The Evaluation of SoundUML concerning usability, general feedback, and accessibility.

8.7.5.1 Personal Information

In this subsection, the results concerning the personal information of the subjects are discussed.

Observing the data regarding the personal information of the participants, there was a less expressive representation of females (18.2%), compared to males (81.8%), and all of the subjects who participated in this study were not hard of hearing. It was not possible to find people who were hard of hearing and further diversify the profile of answers. In terms of logistics, it was harder to arrange the meetings and accommodate people's different schedules, as this was a study done in person. Unfortunately, the few hard of hearing people that participated in the previous study were not available during these evaluation sessions.

Results indicate that the majority of respondents thought of themselves as having an average understanding of UML, with 54.5%., while the remaining 45.5% considered to have an above average understanding, split between "Good" (27.3%) and "Very Good"(18.2%). Only 9.1% of participants use UML regularly, while the remaining have not used it for a year (27.3%) or more (63.6%).

These results show that many of the participants have not had contact with UML for some time, however, they considered their level of understanding of UML as average or above, which we think is positive for this study, as it was necessary to draw on some of that knowledge to answer the questions. As such, people who are more comfortable with

UML may get better results, simply because they have greater experience using UML. To try and mitigate this, support material was created to help refresh participants' memory, which could be consulted at any point during the experiment. Nevertheless, we think we were able to get an interesting, albeit small, sample of participants, as we dealt with people from different fields within computer science, thus allowing us to get different and unconventional insights and suggestions than those we might have considered.

8.7.5.2 Diagrammatic Readings

This subsection is concerned with the discussion of the results from the diagrammatic readings, evaluating how well the auditory version performed, allowing to draw conclusions regarding the hypothesis presented for this study.

In table 8.12, it is possible to observe both the total and correct answers given by participants who experienced the auditory versions of the diagrams, along with the accuracy, which is the percentage of correct answers, for each of the questions. The accuracy was obtained through a very simple calculation, consisting of dividing the correct answers for the auditory version by the total answers of that version: (correct answers of auditory version)/(total answers of auditory version). The focus in each of the questions is a different diagram element that composes the UML class diagrams. For example, in question 1 of the Faculty diagram (Faculty Q1), the component that was studied, was the "Inheritance" relationship.

Table 8.12: Percentage of correct answers (Accuracy) for the auditory versions of the diagrams.

Questions	Class Diagram Element	Total Answers (Auditory Version)	Correct Answers (Auditory Version)	Accuracy (%) (Auditory Version)
Faculty Q1	Inheritance	4	2	50,00%
Faculty Q2	Composition	4	3	75,00%
Faculty Q3	Class Association	4	4	100,00%
Library Q1	Association	3	0	0,00%
Library Q2	Implementation	3	2	66,67%
Library Q3	Dependency	3	3	100,00%
Library Q4	Attribute	3	1	33,33%
GUI Q1	Aggregation	4	0	0,00%
GUI Q2	Class	4	4	100,00%
GUI Q3	Operation	4	4	100,00%

In general, we think that these results suggest a rather positive outcome, since if we consider that there are three possible choices in each question, participants would have a 33.3% chance (in other words, 1 in 3) of answering correctly if they picked a random option. Thus, we can observe that an accuracy well above 33.3% was obtained in 70% of the questions (7 out of 10), where 6 of them register an accuracy higher than 66.6%, with 4 of them having an accuracy of 100%. In these 7 questions, the lowest accuracy was

50%, captured in the first question of the Faculty diagram (Faculty Q1). In 60% of the questions, it is possible to discern an accuracy that is twice as high as if participants had made a random choice.

The lowest results were obtained in questions 1 and 4 of the Library diagram (Library Q1 and Library Q4) and in question 1 of the GUI diagram (GUI Q1), with 0%, 33.33% and 0%, being these questions related to the elements Association, Attribute and Aggregation, respectively. As the amount of results is not very substantial - only 3 participants experienced the auditory version of the Library diagram - each answer ends up having a greater weight in the overall assessment and accuracy of the question. Since these results end up being disparate even compared to the collected values from the other answers concerning the auditory versions of the diagrams, we believe that a factor that could help explain these results would be due to a distraction or inattention on behalf of the participants. In a scenario with more answers, we think that we could possibly obtain values that would trend closer to those obtained in the other questions. However, we have also taken other factors into consideration, such as the complexity of the Library diagram, which could have made the task of recalling all the elements that were presented more complicated. For example, in question 04, which concerns the "Attribute" element, it may have been more difficult to recall the numerous attributes that exist in a class through an auditory reading. Additionally, the chosen sounds for these particular elements, in practice, may not have been as effective as initially anticipated, or even the performed reading of these elements may have been understated and as such, been overlooked by the participants. In a future iteration of these diagrammatic readings, more emphasis could be placed on these particular elements, potentially helping participants to better remember these details and perform more successfully in the task of recalling them.

Following this, we averaged the accuracy obtained through the results of each question in the auditory versions of the diagrams, resulting in an overall accuracy of 62.5%. In our opinion, this figure is not extraordinary, but also not terrible, nor do we think that it suggests a poor performance regarding the auditory readings, since as already mentioned, randomly answering the questions would result in a 33.3% probability of answering correctly, while with an informed judgement based on the conducted diagrammatic readings, an overall accuracy about 2 times above that value was obtained. Thus, and taking into further consideration that the auditory reading performed through our prototype is in its first iteration, we believe that we are in the presence of encouraging results that could be further improved. As such, and although there is room for improvement, these figures seem to support our hypothesis, in the sense that our proposed auditory diagrammatic reading appears to be effective and usable, from the user's point of view, in one of the most important tasks of using UML class diagrams, which is recalling what was observed.

Table 8.13: Percentage of correct answers (Accuracy) for the 3 versions of the diagrams.

Questions	Class Diagram Element	Correct Answers (Visual)	Accuracy (%) (Visual)	Correct Answers (Auditory)	Accuracy (%) (Auditory)	Correct Answers (Hybrid)	Accuracy (%) (Hybrid)
Faculty Q1	Inheritance	3	100,00%	2	50,00%	2	50,00%
Faculty Q2	Composition	3	100,00%	3	75,00%	3	75,00%
Faculty Q3	Class Association	3	100,00%	4	100,00%	4	100,00%
Library Q1	Association	2	50,00%	0	0,00%	3	75,00%
Library Q2	Implementation	4	100,00%	2	66,67%	3	75,00%
Library Q3	Dependency	4	100,00%	3	100,00%	3	75,00%
Library Q4	Attribute	2	50,00%	1	33,33%	3	75,00%
GUI Q1	Aggregation	4	100,00%	0	0,00%	3	100,00%
GUI Q2	Class	4	100,00%	4	100,00%	3	100,00%
GUI Q3	Operation	2	50,00%	4	100,00%	2	66,67%

Although the main focus of this study was to evaluate our proposed auditory diagrammatic readings as a standalone approach, we also sought to evaluate how it would perform when compared to the standard visual version, as well as a hybrid version that joins both the visual and auditory versions. In table 8.13 it is possible to observe the correct answers in the 3 versions of the diagrams, as experienced by different participants, along with the accuracy - the percentage of correct answers - for each question of each version. The accuracy was obtained through a very simple calculation, consisting of dividing the correct answers of a given version by the total answers of that version: (correct answers of a diagram version)/(total answers of that version).

At first, we speculated that using two senses (auditory and visual) in conjunction, as is the case of the hybrid version, would have an advantage over using the other two versions and would achieve the best results, considering that we have 2 communication channels in this approach, instead of only 1. However, as we can see in table 8.13, this was not always the case. In a general manner, the highest results were obtained through the visual versions of the diagrams, followed not so distantly by the hybrid versions and finally, the auditory versions. We believe that a factor that could help justify the visual version obtaining the highest figures, would be related to the overall familiarity of the participants with the different diagrammatic reading versions. Since the visual reading is the standard and usual approach, all participants have already handled these types of diagrams beforehand and therefore had some previous experience, whereas the usage of the auditory and hybrid versions might be a new experience, and therefore yield somewhat lower results. Additionally, the fact that we have a reduced number of subjects in each of the versions - with 11 in total, being further divided by the version of the diagrams they experienced - results in a situation where if one participant in a particular question answers incorrectly - which could even be due to external factors unrelated to the version of the diagram they experienced - could cause large discrepancies between the results obtained in each version. We believe that with a larger sample size, this disparity may be smaller, since a singular answer would have less impact on the final result.

Analysing the results, in 3 of the questions, the auditory version had the same results as one of the other two versions. Those were questions 1 and 2 of the Faculty diagram (Faculty Q1 and Faculty Q2), with a result of 50% and 75% respectively, which was the same as what was obtained for the hybrid version. Lastly, in question 3 of the Library diagram (Library Q3), the auditory version performed as well as the visual and better than the hybrid version, with an accuracy of 100%.

The auditory version outperformed both visual and hybrid in question 3 of the GUI diagram (GUI Q3), with a result of 100%, compared to the figures of 50% and 66.67% for both visual and hybrid versions, respectively. For comparison, the visual version performed better than both auditory and hybrid in 3 questions, with those being questions 1 and 2 of the Faculty diagram (Faculty Q1, Faculty Q2), as well as question 2 of the Library diagram (Library Q2), while the hybrid version performed better in questions 1 and 4 of the Library diagram (Library Q1 and Library Q4).

We can observe that there are instances where all 3 versions performed equally admirably. These are question 3 of the Faculty diagram (Faculty Q3) and question 2 of the GUI diagram (GUI Q2), with every version obtaining an accuracy of 100%.

The auditory version performed below the other two in 4 questions: Library Q1, Library Q2, Library Q4 and GUI Q1. Library Q2 had an accuracy of 66.67%, with this result being only slightly below the 75% of the hybrid version. In the remaining 3 questions, as already mentioned in the standalone evaluation of the auditory version, poor results were obtained. Particularly in question GUI Q1, none of the participants who experienced the auditory version of this diagram answered correctly, and comparatively, in this question, the result obtained for both the other diagrammatic versions was 100%. In questions 1 and 4 of the Library diagram (Library Q1 and Library Q4), the overall results were not excellent across the three versions.

In summary, there were a few questions where the accuracy of the answers concerning the auditory versions of the diagrams was at least equal to one of the other two versions. It was also possible to observe one question where the auditory performed better than both the visual and hybrid versions. As such, we believe these results are encouraging, since in general, the auditory version is not far behind the other versions. Additionally, we should emphasise that some of the poor results obtained for certain questions in the auditory readings may have been caused by SoundUML itself, and not necessarily the way the reading was conceptualised. This proof of concept tool has its flaws, which in turn may have helped mislead some participants, which could have contributed to wrong answers and skew the results.

Table 8.14: Calculated Mean Accuracy of the answers for each of the versions of the diagrams.

Accuracy Mean (Visual Version)	Accuracy Mean (Auditory Version)	Accuracy Mean (Hybrid Version)
85,00%	62,50%	79,17%

The mean accuracy for each of the versions of the diagrams was calculated, with the results being shown in the table 8.14. It is possible to see that the best results were obtained in the visual versions of the diagrams (85%), followed by the hybrid (79.17%) and finally, the auditory (62.5%). Comparing the auditory version to the others, there is a difference of 22.5% for the visual version, and a difference of 16.67% for the hybrid version.

Considering the performance of the auditory diagrammatic readings as a standalone approach, we believe that these results are encouraging since, as already mentioned, the overall accuracy is about 2 times higher than the probability of answering these questions correctly, in a random manner (33.3%). Comparing it to the other versions, although the accuracy was inferior in the auditory version, we do not consider that these results suggest a significant difference between the performance of the three versions, due to the number of answers that were feasible to collect in each version. Additionally, as already observed, in certain situations it is even possible to see that the hybrid version achieved a poorer result than both the visual and auditory versions, as is the case of question Library Q3. In a scenario with more answers, we believe we could possibly obtain accuracy values that would be closer between the different versions.

8.7.5.3 Evaluation Of SoundUML - Usability

Through the results obtained concerning Usability, we calculated the SUS score for each of the participants' answers. Afterwards, we calculated the average of these scores, which resulted in a score of 84.09. A SUS score can range from 0 to 100, with the value 68 being considered the average. Thus, we think that the value we obtained is very positive, suggesting that this prototype, despite being a first iteration and having much room for improvement, is on a very good starting course. We believe that perhaps this obtained value can be explained by the tool's simplicity of learning and use, resulting in overall above-average scores from the different participants' answers. Furthermore, we think that this result could be indicative that the tool may have some wider appeal than anticipated, and prove to be a useful addition to the resources and tools already available for UML modelling activities.

In order to obtain a more detailed analysis concerning the scores of the different participants, a box-plot was constructed, as shown in figure 8.41. It is possible to observe that the the lowest obtained score was 75, and the highest score was 92.5. The median value obtained was 82.5, the first quartile with a value of 80, the third quartile with 87.5 and the interquartile range - the distance between the upper and lower quartiles - had a value of 7.5.

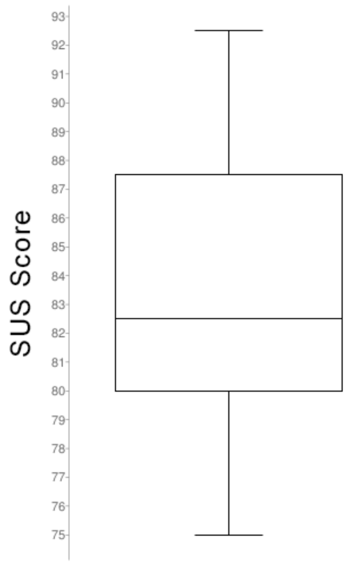


Figure 8.41: Box plot with SoundUML's SUS score.

Further descriptive statistics of the system usability scale of our tool have been calculated, as shown in table 8.15. A standard deviation of 5.567 was obtained, along with the values of 0.077 and -1.059 for skewness and kurtosis, respectively. To understand the variability in our estimate of the SUS score, we have also computed a confidence interval of 95%, obtaining the result of 84.1 ± 3.29 (±3.9%) [80.800 – 87.380]. As such, with 95% confidence, the population mean SUS score is between 80.8 and 87.4, based on only 11 samples. Accounting for this margin of error, we are still able to achieve a very positive result in terms of the SUS score of the tool, well above the considered average of 68.

Table 8.15: Descriptive Statistics Of SoundUML's System Usability Scale

	N	Mean	Median	Std.Dev	Skew.	Kurt.
SoundUML	11	84.09	82.5	5.567	0.077	-1.059

8.7.5.4 Evaluation Of SoundUML - General Feedback

This section discusses the general feedback and overall assessment of SoundUML, along with the readings it performed.

The majority of participants rated their experience with SoundUML positively, with 72.7% considering that it was "Good", while 18.2% graded it as "Excellent". The remaining 9.1% rated it as "Average".

All of the subjects considered that the diagrammatic readings performed by SoundUML were intuitive enough to understand the diagrams, serving to further corroborate our hypothesis concerning the use and effectiveness of our proposed auditory diagrammatic reading, from the user's point of view. Additionally, we asked participants to choose which version of the diagrammatic readings they preferred, with the vast majority selecting the hybrid version (90.9%), explaining that they considered the additional information useful to really grasp all the details in the diagram. The remaining 9.1% opted for the auditory version. Afterwards, we were explained that this choice was due to the fact that according to the participant, the auditory reading provided a more streamlined approach, with special attention to certain details of a given diagram that might otherwise be overlooked. In this way, the auditory version allowed them to focus more closely on the reading, and in turn, recall it better. Thus, these results seem to support the notion that the users' reception to the proposed auditory diagrammatic reading was positive, being supported by their preference for the hybrid reading or even the auditory one.

Overall, the great majority of respondents seemed to think that our tool and its proposed auditory diagrammatic reading could be useful to complement and enrich the existing visual notation, with 90.9% answering "Yes". These answers appear to support our second set of hypotheses, specifically regarding if SoundUML could complement and enrich the existing visual notation, according to participants.

Observing the aspects that the participants thought were the strongest, it is possible to conclude that in general, the sounds chosen were considered intuitive, as well as the order in which the elements were presented, aspects that are related to the proposed auditory reading. More specifically regarding the tool, some participants considered that SoundUML could also be used as a learning tool, and that the way the reading is performed, in an almost step-by-step way, allowed them to better assimilate what was being presented.

Regarding the negative aspects, many of the detected problems and feedback given by the participants were more related to the tool itself and some of the choices made for it, than to the diagrammatic readings themselves. These are aspects such as the UI not persisting in the position the user wants, the text-to-speech temporarily freezing the UI if the string it is reading is too long and the voice being too robotic. These were problems that unfortunately occurred due to the technologies used (namely Jface and FreeTTS). In a future iteration of this tool, the integration of other more mature technologies for these purposes could be explored, replacing those which are currently implemented. Due to the modular nature of our tool, it would be possible to do this without major difficulties.

In our analysis, it appears that participants considered that the auditory reading performed by SoundUML was better for more detailed parts of the diagram, allowing for a more "focused"view, whereas the visual approach gives a better overview of the diagram. In order to mitigate this issue, one of the ideas that we proposed in chapter 6 was to start by presenting the total number of existing classes and other elements in the diagram, so that users can have a first "naive" impression regarding the size and complexity of the diagram. Unfortunately, we did not have enough time to implement all that was intended. As such, we believe that this proof of concept tool has a number of flaws. These flaws could possibly have helped to induce some participants into wrong answers in the questions regarding the auditory version of the diagrams, contributing to skewed results. The participants also suggested some features that focus very much on improving the tool's usability and effectiveness that we thought made perfect sense and would be feasible. Given the modular nature of SoundUML, it would be possible to implement what participants suggested, along with our full list of proposals from chapter 6, that we, unfortunately, did not have time to implement in this first iteration.

8.7.5.5 Evaluation Of SoundUML - Accessibility

This section is concerned with the opinion of participants regarding the accessibility of SoundUML.

Participants considered that SoundUML could be useful for visually impaired people who want to carry out tasks related to UML Modelling. As it can be observed, 72.7% of subjects responded "Yes", while the remaining 27.3% were divided between the answers "Maybe" (with 18.2%) and "Don't Know" (9.1%).

Lastly, respondents were asked if they would recommend SoundUML to visually impaired people, and the vast majority responded positively, with a total of 90.9% of subjects saying "Yes". Only 9.1% responded negatively. In our understanding, these answers could be due to the fact that setting up the tool in Modelio might be troublesome for visually impaired people, as Modelio does not provide any accessibility features.

However, we believe that these results are highly encouraging, especially considering that the starting point is this first version of our tool, which is a proof of concept. As such, there is room for further improvements, and given the modular nature with which this tool was designed, it will also be possible to make new additions more easily and thus accommodate even better the necessities of visually impaired people. Nevertheless, we know that for that to happen, it will be necessary to test this tool with visually impaired people, in order to accurately pinpoint their needs and evaluate the performance of the SoundUML tool in those use cases, something that unfortunately, was not possible in this experimental study. It was only possible to assess the opinion of participants on these questions concerning accessibility and that they did not fit the profile of visually impaired people with an interest in diagram modelling. Further studies should be done with people that match those characteristics.

In conclusion, these results seem to support our second hypothesis which states that the participants consider that our constructed tool, SoundUML, provides added value, in regard to the topics of accessibility, in the sense of enabling visually impaired people to carry out certain UML modelling tasks, and enriching the existing visual notation.

8.7.6 Threats To Validity

In this section, the threats to validity that were identified will be presented.

The number of subjects was limited and we had to break down this number further, since participants were divided by different versions of the diagram. In other words, different participants observed different versions of the diagrams. In terms of results, this can cause a very discrepant variation between the overall performance and assessment of each version if one person answers incorrectly, as each individual answer ends up having a greater weight. We believe that with a larger number of participants, it would be possible to mitigate this issue. In the future, this experimental study should be performed with blindfolded and visually impaired people as well. In that way, it would be possible to infer if, in fact, our tool would help in addressing the topic of accessibility of visually impaired people in modelling tasks.

Since this study was oriented towards the recall capabilities of the participants, more specific and detailed questions were not asked, for example regarding multiplicities or roles in associations, etc. As such, diagrams with a reasonable and moderate level of complexity were used, and it was not possible to explore the use of more complex diagrams. It should also be mentioned that only multiple-choice questions were asked, and no open-answer questions were tested. In the future, this type of question should also be explored, to achieve an even better assessment of the performance of the proposed auditory diagrammatic reading.

Additionally, the obtained results regarding the effectiveness and accuracy of each version may not be totally reliable, as different people may be more or less suitable for this type of recall activity. That is, certain people may have had an easier time memorising certain details than others. Furthermore, we are comparing answers given by distinct people, which have varying levels of knowledge and familiarity with UML and will simply perform differently. In this way, and given the sample size of registered responses, the results could become even more discrepant and that might not necessarily be related to that specific version of the diagram itself. This is a factor that we believe could be mitigated with a larger sample size.

Regarding the results of the diagrammatic readings, SoundUML is a proof of concept tool, and as such, has some shortcomings. These problems may have affected the answers given to the questions concerning the auditory and hybrid versions of the diagrams and in turn, impacted the accuracy obtained for these two versions.

Although we tried to minimise the subjects' learning throughout the experiment by using 3 different diagrams where in each of these, a specific participant would only experience one of the versions, the subjects may still have had a better understanding of what was asked of them in the 2nd and 3rd diagram that they observed. This may be because they could better anticipate the type of questions they would be asked, and were more aware of certain details in the subsequent diagrams that they were shown. For example, after observing the 1st diagram, a subject could have realised that they were going to be asked similar questions to: "How many inheritance relationships exist in this diagram?"and in the diagrams that followed, be more attentive to those kinds of details.

As already mentioned, after participants read the provided support material, questions related to it were clarified and an explanation of its content was carried out. Thus, different users asked different questions and acquired distinct knowledge. At the same time, the explanation given to the participants was not always the same, since a recording was not previously made and distributed to all participants.

8.8 Identified Shortcomings

In this section, the identified shortcomings will be detailed, concerning both the proposed auditory diagrammatic readings and the tool developed for this purpose.

Regarding the developed tool, due to the design of the program Modelio, the installation steps of our solution might not be the most user-friendly, especially for visually impaired users. As such, this may create difficulties for these people to use our tool without the help of a third party, at least during the installation process. Additionally, in order for our tool to perform the auditory diagrammatic reading of any class diagram, the diagram has to have been previously built in Modelio. Since Modelio does not have features that help with accessibility, it may make the task of using SoundUML more difficult.

Other limitations that were identified, are related to user interaction, namely: not being able to use voice commands, inability to choose a particular element to be read, the UI window not persisting in the same place, the text-to-speech temporarily freezing the UI if the string it is reading is too long and TTS voice being very robotic/unfriendly. Some of these aspects were also identified by the participants as the weakest parts of SoundUML. These are all points that could be improved in a future iteration of our tool. Additionally, it should provide a wide range of configurable options to further enhance the accessibility of auditory diagrammatic readings.

We should also mention that some major problems that arose during the development of this tool, are related to the use of Modelio, which ended up costing us development time that could have been used to implement other features or greatly improve the user experience. As we have already mentioned, the decision to use Modelio came in great part, due to the fact that it was an open-source platform. Therefore, we expected that there would be a community ready to aid with some issues or questions that could emerge during the implementation process. Unfortunately, that was not the case, as we never got any answers to the questions we asked in the dedicated forums, so the development process itself became a somewhat more complicated and time-consuming endeavour. Furthermore, the documentation itself was not very detailed in general, as many of the methods did not have a description, and their names were not very explicit as well. Rarely, when there were descriptions, some were written only in french, so it was necessary to improvise and do a bit of reverse engineering to understand how the program worked. Additionally, the official documentation contained a few dead links and was outdated in certain parts, as a few of the official examples in the documentation contained deprecated code. It is worth mentioning that during the writing of the implementation section of our solution, we went back to the official Javadoc documentation to consult certain details, and the links were not working, making it impossible to view the Javadoc, since it was hosted on the Modelio website. Trying to access it or any other link from the Modelio website, only redirected to their GitHub repository. At the time of writing, their website, along with the Javadoc documentation is still unavailable, having a month already passed.

Concerning the proposed auditory diagrammatic readings, it will always be challenging to get results as good as those from the visual versions of the diagrams, since we are extrapolating to the auditory channel, what was originally intended to be used with vision. Not everything will be a 1-to-1 translation and further work is required to fully realise the potential of this approach.

Although we have always tried to follow the guidelines and existing literature, unfortunately, in practice, sound design is a subject that is not very systematic, resulting in decisions that may seem somewhat arbitrary or based on intuition. Thus, not all sounds may work the same way and be effective for all users. However, we believe that even better results can be obtained by working with people with a sound design background, to develop a new iteration of our proposed catalogue of sounds.

Since it was not possible to implement all the ideas we suggested in chapter 6, the resulting diagrammatic readings may have been more limited. This factor may have negatively affected the overall results which could have prevented even better figures from being achieved. As this is a proof of concept, we believe that there is still plenty of room for improvement, such as: starting the reading from the inheritance relationships; rankings of diagram elements by attributing different levels of importance to each element in a diagram; enumerating the elements that exist in a diagram at the beginning of the reading, in order to give users a general idea of the level of complexity of the diagram; etc. There was not enough time to implement these proposals in this version of our prototype, but would be perfectly feasible to do so in future versions.

These readings, in practice, had to be implemented in an existing program, so that the experimental study could be carried out. For this, we chose to use Modelio due to the factors we have previously enumerated in this chapter. As such, we are in part, constrained by this tool and its idiosyncrasies. In the future, we would like these readings to be software agnostic, so that we do not have to depend on a specific program to be able to employ the proposed approach, allowing it to be a standalone reading that is compatible with different file formats and diagrams from different programs.

9 Conclusion

9.1 Summary

The main goal of this dissertation and the respective developed work was to construct a foundational framework that can provide a basis for decisions when designing auditory notations in software engineering while simultaneously striving to answer our research question: "Which guiding principles should we consider to using sound and voice as modelling notations, applied in the context of modelling with UML?". This framework consisted of establishing auditory principles, detecting patterns in diagrammatic readings, and creating a catalogue of sounds for diagram elements based on semiotics, culminating in the development of a tool that performs auditory readings of UML Class diagrams.

First, it was necessary to analyse semiotics and then, more concretely, the semiotics of the audible field to understand the influence different acoustic cues have on assigning meaning to sound effectively. As this is a topic outside the scope of computer engineering, it proved to be an arduous task, which required special consideration. Nevertheless, it offered significant contributions to our field, since it allowed us to associate this newly acquired knowledge, to sound design in the context of software. This is specified in chapter 2. In the context of this work specifically, it also allowed us to justify decisions during the elaboration of the catalogue of sounds, specified in chapter 7.

Afterwards, a systematic literature review was elaborated in chapter 3, whose objective was to identify and evaluate all available research relevant to the topics regarding the construction of auditory notations in software engineering, along with helping answer our research question. Other aspects were also considered, such as guidelines, that substantiated some of the later choices for the sounds constructed in chapter 7, as well as various tools that implemented some of the planned ideas. Unfortunately, during the realisation of this literature review, we did not find much research that completely intersected with the goals of this work, such as the construction of auditory notations in software engineering, or combining sound with diagrammatic representations in a structured manner. Tools intended explicitly for this purpose, or prepared to receive implementations that accommodate these needs were not found as well. For example, concepts such as using voice with speech recognition, or sound in the context of software are explored, but never analysed as a whole or to create a structured auditory notation. However, a few ideas overlapped with the topics of this document and all the research offered some interesting insights that were applied to this work. Several articles also provided design guidelines and frameworks for a more scientific sound design method, later discussed in chapter 5, while others helped to shape our tool, *SoundUML*, discussed in chapter 8.

The works that served as primary sources of insights, accompanied by their key ideas and limitations were further analysed in Chapter 4, together with the paper *The "Physics"Of Notations*, which played a crucial role in the remainder of this work. The visual principles were transposed in a manner analogous to sound, to create our auditory principles for constructing auditory notations. These were then discussed in chapter 5, along with the gathering and analysis of several sound design guidelines and frameworks, which served as the basis for elaborating our Catalogue of sounds for UML Class Diagrams and subsequent creation of the sounds themselves.

With these guiding principles defined and the guidelines analysed, in chapter 7, the aforementioned catalogue of sounds was constructed, and an experiment was carried out with the purpose of understanding if there were differences from the user's point of view in the usage of a catalogue of sounds that follows these auditory principles and semiotics of the audible field, *versus* an arbitrary catalogue of sounds that disregards these principles and semiotics, which was created for this purpose as well. Additionally, a user study was also prepared, with the goal of determining how relevant participants considered the proposed auditory principles to be. In general, we can affirm that the results obtained were encouraging, since for most of the sounds, the respondents considered the proposed sounds based on the auditory principles and semiotics to be more adequate. This suggests that there are indeed positive differences when using a more informed method for the construction of sounds for auditory notations in contrast to a naive approach. In a broad manner, participants also perceived the principles as relevant.

We also carried out another experimental study about detecting any emerging patterns in how different users carry out diagrammatic readings of UML Class and State Machine Diagrams, by observing the most common actions among the participants. It was necessary to create support material to refresh the participants' memory on UML notation, considering that some of them had no contact with the language for a long time. This study served as a basis for the auditory reading in our developed tool. It was possible to identify several patterns and propose features for a possible construction of diagrammatic reading tools that accommodate these patterns. All of these findings are detailed in chapter 6.

Finally, to validate and draw conclusions concerning our proposals for a foundational framework and subsequent auditory diagrammatic reading of UML class diagrams, a prototype called *SoundUML* was developed, which incorporates the aforementioned auditory principles, along with the conclusions drawn from the experimental studies regarding the diagrammatic readings and the catalogue of sounds. This tool allows UML class diagrams to be read to the users, through the usage of voice synthesis and audio playback. While not as effective as the visual approach, the registered results were generally encouraging, suggesting that these auditory readings are perfectly usable and effective both as a standalone option, or as an approach that joins both the visual and hearing senses in a manner that benefits a large and diverse population of software engineers, in one of the most important tasks of using UML class diagrams, which is recalling what was observed. Additionally, according to the feedback received, even though this proof-of-concept is incomplete in some respects and has some limitations, it was considered to be a step in the right direction towards increasing accessibility for visually impaired people in tasks of this kind.

9.2 Contributions

The contributions of this work are listed as follows:

- **Analysis Of Semiotics (Both in general and of the audible field)**, offering important contributions to the field of computer engineering, as it was possible to combine this newly acquired knowledge with sound design in the context of software, allowing for the justification of certain sound design choices when creating an auditory notation.

- **Systematic Literature Review concerning the topics related to the construction of auditory notations in software engineering**, while considering other useful aspects such as guidelines and tools that offered some interesting insights that were applied to this work.

- **Elaboration of principles for constructing auditory notations**, akin to the principles for visual notations proposed in *The "Physics" of Notations* by Moody, serving to substantiate the choices made in the creation of a catalogue of sounds for UML class diagrams, along with the development of our tool.

- **Collection of design guidelines & frameworks for sound design in the context of software and VUIs**, facilitating their examination by documenting them all in one locale, corroborating some of the decisions made in the elaboration of the sounds described in our proposed Catalogue Of Sounds For UML Class Diagrams.

- **Proposals for the construction of tools that perform auditory readings of diagrams**, by detecting emerging patterns in how different users carry out a diagrammatic reading of UML Class and State Machine Diagrams. This was accomplished through a study focused on observing the most common actions among the participants. These findings also served as a basis for the implementation of the auditory diagrammatic readings performed by our developed tool.

- **Construction of a catalogue of sounds for each element of UML Class diagrams**, based on the study of semiotics, the proposed auditory principles and collected guidelines. This catalogue was followed by an experimental study with the main goal of understanding whether there is a difference in the effectiveness of using a catalogue of sounds that follows the elaborated auditory principles and semiotics of the audible field, or one that disregards them.

- **Auditory readings of UML Class diagrams, realised through the development of *SoundUML*.** This proof-of-concept tool allows users to listen to diagrammatic readings through audio playback and text-to-speech. It incorporates the proposed auditory principles and conclusions drawn from the previous experimental studies (detection of patterns in diagrammatic readings and the catalogue of sounds).

Additionally, part of this work was adapted into a scientific paper [68], which was submitted to the *MODELS 23* conference. This article is included in appendix D.

9.3 Shortcomings

This work focused mainly on UML Class diagrams, with the study regarding the detection of patterns in diagrammatic readings of Chapter 6 featuring State machine diagrams as well. Although Structure Diagrams and Behaviour Diagrams were represented through these two types of diagrams, there are numerous other UML diagram types that have not been explored.

The scope of our efforts had to be somewhat broadened in order to touch upon all of its relevant subjects. A more focused work could have been achieved if these concepts were already defined beforehand. Additionally, it was not possible to measure and compare our results with other studies that have been previously done, since most of the preceding efforts only focused on one specific topic. This work sought to be a "stepping stone" on the path to unifying these ideas, and we hope that in the future, as the research in this area progresses, it will be possible to have more studies of this type.

The task of analysing semiotics required us to dedicate a significant amount of time to it, as it is a field of study outside computer science. Therefore, it was a topic where we had no previous experience. Although it provided an important contribution - since these were analysed from a computer engineering point of view, and helped to substantiate the creation of a catalogue of sounds that followed structured practices, instead of an arbitrary approach - it limited in advance, the time that we could devote to other tasks within this work, such as a more extensive development of our realised solution.

Regarding the experimental studies conducted, overall, there was a limited number of subjects, which were chosen through convenience sampling. Thus, there may be certain profiles which are not represented. One of the main difficulties found was related to how these were going to be performed, since they were not exactly conventional in the sense of what would be expected with UML, as they involved sound. The experimental studies had to be carried out with the objective of guiding our decisions and supporting the concepts studied during this work, as we verified during our research that there were no existing detailed good practices or guiding principles, being then necessary to define them.

The study of detecting patterns in diagrammatic readings was extensive, taking an average of around 50 to 70 minutes, ending up being a limiting factor regarding the number of participants. During the process of transcribing the conducted interviews, bias may have been added and the results may have become somewhat skewed, even though the transcriptions were left as similar as possible to the way the participants described the diagrams. However, they are not an exact match and some compromises had to be made. Many of the detected patterns may not be entirely universal or relatable to all users.

Concerning the constructed catalogue of sounds, sound design is a subject that is not very systematic, resulting in decisions that may seem somewhat arbitrary or based on intuition. People have different sensibilities and life experiences, so there may always be certain sounds that won't be the most adequate for everyone.

We had limited development time to create a prototype that realised in practice, our proposed auditory diagrammatic readings, due to the aforementioned challenges encountered and the fact that no pre-existing tools allowed us to create a streamlined solution. As such, it was not feasible to materialise in that time frame certain features that were initially idealised, such as speech-to-text and, consequently, the ability to edit models with voice. It was also not possible to implement all the ideas we suggested in chapter 6 which may have resulted in somewhat more limited diagrammatic readings than what was intended.

Since our tool was developed in Modelio, we were constrained by this program and its idiosyncrasies. The installation steps of our solution might not be the most user-friendly, especially for visually impaired users. Other identified limitations are related to user interaction. Currently, it only fully supports the auditory diagrammatic reading of UML Class diagrams.

It is also important to mention that the proposed auditory principles were only lightly tested, and that it was not possible to perform the experimental studies with visually impaired people, as we were unable to contact any who would fit the required profile. Finally, as we are extrapolating to the auditory channel what was originally intended to be used with vision, not everything will be a 1-to-1 translation and further work is required to fully realise the potential of this approach.

9.4 Future Work

Regarding the experimental studies carried out in this work, ideally, it would be possible to conduct them again with a more significant amount of participants, in order to have a larger sample pool, hopefully resulting in even more precise results. In regards to the study related to detecting patterns in the diagrammatic readings, it could be performed again, with the help of eye tracking technology, in order to understand where participants look and what do they instinctively prioritise. Other types of UML diagrams could be studied as well.

Concerning the catalogue of sounds experiment, some participants proposed other sounds for certain elements. To observe if even better results would be drawn, it could be possible to explore these suggestions or iterate on the existing ones, together with people with musical education and background. Additionally, catalogues of sounds for other types of UML diagrams could also be elaborated.

In the last experiment, regarding our proposed auditory reading, we could try a different approach by including open-ended or other more specific and detailed questions. We could also explore the use of diagrams with a greater level of complexity.

The proposed auditory principles could be further developed and discussed as well, since due to the limitations of the developed tool and the nature of the conducted experimental study, it was not possible to extensively test all principles. Some could be further explored and developed, such as *complexity management* and *cognitive fit*.

As our tool was created as a proof-of-concept, we believe that there is a lot of work that can be done to improve the overall experience of using it. To start, we would like to integrate more mature technologies for the interface and TTS, replacing those which are currently implemented. Due to the modular nature of our tool, it would be possible to do this without major difficulties.

It could be interesting to implement all the ideas we suggested in chapter 6 in order to achieve more complete auditory readings. It was observed in that study that users gave different priorities to distinct classes and chunks of the diagram. *SoundUML* could support different readings of the diagrams by ordering the elements in different ways, leaving that choice to the users on how they would want a diagram to be presented. Additionally, the use of machine learning to create algorithms for the diagrammatic readings, coupled with the patterns we detected could be potentially explored.

We should also address the identified weak points and suggestions made by participants, which we think can greatly contribute to improving the experience of using this tool. Some of these are: reading a specific element of the diagram by selecting it; when using the hybrid approach, visually highlighting the element that is being shown in the prompt of the SoundUML; a persisting position for the UI window; more natural/empathetic TTS voice; Add a mechanism to let users know how much of the diagram has been covered at any moment. To tackle the issue of being more difficult to obtain a high-level view of the diagrams through the auditory readings, we could try using MIDI musical sounds to create motifs, resulting in a higher level/spatial reading of the diagram elements, which could contribute to the principles of cognitive integration and complexity management.

Additionally, the tool could support other types of UML diagrams and have a more interesting interaction model. The UI could work with different inputs, being fully functional with voice commands. There could be simplified and more complex commands for novice and experienced users, respectively, with this being directly related to the principle of *Cognitive Fit*. Another interesting idea concerning voice input would be the usage of AI voice recognition technologies, potentially allowing for better results regarding accuracy. Finally, further testing of SoundUML should be done with visually impaired people, to better understand their needs, which might help to develop a more interesting interaction model.

Bibliography

[1] J. M. Lourenço. *The NOVAthesis LaTeX Template User's Manual*. NOVA University Lisbon. 2021. URL: https://github.com/joaomlourenco/novathesis/raw/master/template.pdf (cit. on p. ii).

[2] E. C. Blake and I. Cross. "The acoustic and auditory contexts of human behavior". In: *Current Anthropology* 56.1 (2015), pp. 81–103 (cit. on pp. 1, 2).

[3] *Vision Atlas*. 2021-06. URL: https://www.iapb.org/learn/vision-atlas/ (visited on 2022-02-11) (cit. on p. 1).

[4] J. Blauert. *Communication acoustics*. Vol. 2. Springer, 2005, pp. 193–200, 205–219 (cit. on pp. 1, 10, 13, 38, 39, 68).

[5] B. Kenwright. "There's More to Sound Than Meets the Ear: Sound in Interactive Environments". In: *IEEE Computer Graphics and Applications* 40.4 (2020), pp. 62–70 (cit. on pp. 1, 25, 40).

[6] S. Kuhar and G. Polančič. "Conceptualization, measurement, and application of semantic transparency in visual notations". In: *Software and Systems Modeling* (2021), pp. 1–43 (cit. on pp. 1, 2, 6, 9).

[7] M. Brambilla, J. Cabot, and M. Wimmer. *Model-Driven Software Engineering in practice*. Morgan & Claypool, 2012 (cit. on pp. 1, 14–16).

[8] D. Moody. "The "physics" of notations: toward a scientific basis for constructing visual notations in software engineering". In: *IEEE Transactions on software engineering* 35.6 (2009), pp. 756–779 (cit. on pp. 2, 31, 33, 237).

[9] T. Rivas-Pérez et al. "EULER-Mathematical editing by voice input for people with visual impairment". In: *2019 International Conference on Inclusive Technologies and Education (CONTIE)*. IEEE. 2019, pp. 9–95 (cit. on pp. 3, 29).

[10] T. J. Hubbell, D. D. Langan, and T. F. Hain. "A voice-activated syntax-directed editor for manually disabled programmers". In: *Proceedings of the 8th International ACM SIGACCESS Conference on Computers and Accessibility*. 2006, pp. 205–212 (cit. on pp. 3, 28).

[11] T. Chambel. “Plataforma de construção de Linguagens de Modelação com sintaxe concreta Som e Voz”. MA thesis. Universidade Nova de Lisboa, 2021 (cit. on pp. 3, 35).

[12] D. Chandler. *Semiotics: the basics*. Routledge, 2007, pp. 29–30, 36–38 (cit. on pp. 6, 9).

[13] M. Bergman and S. Paavola. *The Commens Dictionary of Peirce’s Terms. Peirce’s Terminology in His Own Words*. 2003. URL: https://web.archive.org/web/20150402190449/http://www.commens.org/dictionary/term/categories (visited on 2021-11-06) (cit. on p. 7).

[14] N. Everaert-Desmedt. “Peirce’s semiotics”. In: *An Introduction to Applied Semiotics*. Routledge, 2019, pp. 241–249 (cit. on pp. 7–9).

[15] P. Tanvi Gupta. *Understanding semiotics: Firstness, secondness and thirdness*. 2018-05. URL: https://www.linkedin.com/pulse/understanding-semiotics-firstness-secondness-thirdness-tanvi-gupta (visited on 2021-11-06) (cit. on p. 8).

[16] C. S. Peirce. *Collected papers of Charles Sanders Peirce*. Harvard University Press, 1931 (cit. on pp. 8–10).

[17] L. C. Kruse, S. Seidel, and A. Maedche. “Levels of Digital Representation: Semiotics and the Articulation of Meaning”. In: (2021) (cit. on p. 8).

[18] A. Atkin. *Peirce’s theory of signs*. 2010-11. URL: https://plato.stanford.edu/entries/peirce-semiotics/ (visited on 2021-11-06) (cit. on pp. 9, 10).

[19] I. Mittelberg. “Peirce’s universal categories: On their potential for gesture theory and multimodal analysis”. In: *Semiotica* 2019.228 (2019), pp. 193–222 (cit. on p. 9).

[20] I. Capeller. “Sounds, Signs and Hearing: Towards a Semiotics of the Audible Field”. In: *CROSS-INTER-MULTI-TRANS* (2018), p. 173 (cit. on pp. 10–12).

[21] M. Chion. *Audio-vision: Sound on Screen*. Columbia University Press, 1994 (cit. on p. 10).

[22] M. Solomos. “Xenakis as a sound sculptor”. In: *Musik interkulturell, publications de l’Institut für Neue Musik und Musikerziehung Darmstadt* 44 (2004), pp. 161–169 (cit. on p. 11).

[23] OMG. “OMG Unified Modeling Language (OMG UML), Version 2.5.1”. In: Object Management Group. 2017-12. URL: https://www.omg.org/spec/UML/2.5.1/ (visited on 2022-02-11) (cit. on p. 16).

[24] J. Allen et al. *From text to speech: The MITalk system*. Cambridge University Press, 1987 (cit. on pp. 17, 18).

[25] J. Schwoebel. *Introduction to Voice Computing in Python*. NeuroLex Laboratories, 2018 (cit. on p. 17).

[26] *Alexa Skills and Features*. 2019. URL: https://www.amazon.com/alexa-skills/b?ie=UTF8&node=13727921011 (visited on 2022-01-12) (cit. on p. 17).

[27] I. C. Education. *What is natural language processing?* URL: https://www.ibm.com/cloud/learn/natural-language-processing (visited on 2022-01-10) (cit. on p. 17).

[28] *Status of English Language Standard for Use in Civil Aviation*. 2003-04. URL: https://www.icao.int/SAM/Documents/2003/RAAC8/RAAC8IP18.pdf (visited on 2022-01-11) (cit. on p. 17).

[29] V. L. English. *The language of Air Travel*. 2019-05. URL: https://learningenglish.voanews.com/a/the-language-of-air-travel/4871734.html (visited on 2022-01-11) (cit. on p. 17).

[30] T. Dutoit. *An introduction to text-to-speech synthesis*. Vol. 3. Springer Science & Business Media, 1997 (cit. on p. 18).

[31] X. Huang and L. Deng. "An Overview of Modern Speech Recognition." In: *Handbook of natural language processing* 2 (2010), pp. 339–366 (cit. on pp. 18, 19).

[32] I. C. Education. URL: https://www.ibm.com/cloud/learn/speech-recognition (visited on 2022-01-16) (cit. on p. 18).

[33] X. Huang, J. Baker, and R. Reddy. "A historical perspective of speech recognition". In: *Communications of the ACM* 57.1 (2014), pp. 94–103 (cit. on p. 19).

[34] *Speech-to-text basics | cloud speech-to-text documentation | google cloud*. URL: https://cloud.google.com/speech-to-text/docs/basics (visited on 2022-01-17) (cit. on pp. 19, 20).

[35] *Core Concepts, architecture and Lifecycle*. 2021-08. URL: https://www.grpc.io/docs/what-is-grpc/core-concepts/#bidirectional-streaming-rpc (visited on 2022-01-17) (cit. on p. 20).

[36] *What are voice user interfaces?* URL: https://www.interaction-design.org/literature/topics/voice-user-interfaces (visited on 2022-01-18) (cit. on pp. 20, 42).

[37] C. Nowacki, A. Gordeeva, and A.-H. Lizé. "Improving the usability of voice user interfaces: A new set of ergonomic criteria". In: *International Conference on Human-Computer Interaction*. Springer. 2020, pp. 117–133 (cit. on pp. 20, 30, 42).

[38] B. Kitchenham and S. Charters. "Guidelines for performing systematic literature reviews in software engineering". In: (2007) (cit. on p. 21).

[39] E. Souza, A. Moreira, and M. Goulão. "Deriving architectural models from requirements specifications: A systematic mapping study". In: *Information and Software Technology* 109 (2019), pp. 26–39 (cit. on p. 23).

[40] S. Jalali and C. Wohlin. "Systematic literature studies: database searches vs. backward snowballing". In: *Proceedings of the 2012 ACM-IEEE international symposium on empirical software engineering and measurement*. IEEE. 2012, pp. 29–38 (cit. on p. 23).

[41] C. Wohlin. "Guidelines for snowballing in systematic literature studies and a replication in software engineering". In: *Proceedings of the 18th international conference on evaluation and assessment in software engineering*. 2014, pp. 1–10 (cit. on p. 23).

[42] L. Berman. "Sound as an aid in understanding low-level program architecture". In: *2010 IEEE 18th International Conference on Program Comprehension*. IEEE. 2010, pp. 58–59 (cit. on p. 24).

[43] A. Mancino and G. Scanniello. "Software Musification". In: *2017 21st International Conference Information Visualisation (IV)*. IEEE. 2017, pp. 127–132 (cit. on p. 24).

[44] K. Hussein et al. "Sonification design guidelines to enhance program comprehension". In: *2009 IEEE 17th International Conference on Program Comprehension*. IEEE. 2009, pp. 120–129 (cit. on pp. 24, 39).

[45] *What is Max?: Cycling '74*. URL: https://cycling74.com/products/max (visited on 2021-07-14) (cit. on p. 24).

[46] D. Hug and N. Misdariis. "Towards a conceptual framework to integrate designerly and scientific sound design methods". In: *Proceedings of the 6th audio mostly conference: A conference on interaction with sound*. 2011, pp. 23–30 (cit. on pp. 25, 41, 68, 238).

[47] S. Coburn and C. B. Owen. "UML diagrams for blind programmers". In: *Proc. of the 2014 ASEE North Central Section Conference. Oakland, USA*. 2014, pp. 1–7 (cit. on pp. 26, 34).

[48] R. G. Brookshire. "Teaching uml database modeling to visually impaired students". In: *Issues in Information Systems* 7.1 (2006), pp. 98–101 (cit. on pp. 26, 30).

[49] L. Luque et al. "On the inclusion of blind people in UML e-learning activities". In: *Revista Brasileira de Informática na Educação* 23.2 (2015) (cit. on p. 27).

[50] R. Jolak, B. Vesin, and M. R. Chaudron. "Using Voice Commands for UML Modelling Support on Interactive Whiteboards: Insights and Experiences." In: *CIbSE*. 2017, pp. 85–98 (cit. on pp. 27, 28).

[51] D. Black, E. J. Rapos, and M. Stephan. "Voice-driven modeling: Software modeling using automated speech recognition". In: *2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*. IEEE. 2019, pp. 252–258 (cit. on p. 29).

[52] J. B. Krygier. "Sound and geographic visualization". In: *Modern cartography series*. Vol. 2. Elsevier, 1994, pp. 149–166 (cit. on pp. 38, 238).

[53] G. Reggio et al. "What are the used UML diagrams? A Preliminary Survey." In: *EESSMod@ MoDELS*. 2013, pp. 3–12 (cit. on p. 46).

[54] M. Seidl et al. *UML@ classroom*. Springer, 2015 (cit. on pp. 53, 58, 59).

[55] *Uml Quiz*. URL: http://elearning.uml.ac.at/ (visited on 2022-06-10) (cit. on p. 61).

[56] ModelioOpenSource. *What is a Module?* URL: https://github.com/ModelioOpenSource/Modelio/wiki/What_is_a_module (visited on 2023-01-27) (cit. on pp. 100, 103).

[57] D. Steinberg et al. *EMF: eclipse modeling framework*. Pearson Education, 2008 (cit. on p. 103).

[58] R. Gronback. *Eclipse modeling project: The eclipse foundation*. URL: https://www.eclipse.org/modeling/emf/ (visited on 2023-02-28) (cit. on p. 103).

[59] ModelioOpenSource. *Modelio documentation*. URL: https://github.com/ModelioOpenSource/Modelio/wiki/ (visited on 2023-02-10) (cit. on pp. 103, 104).

[60] ModelioOpenSource. *Modelio API*. URL: https://github.com/ModelioOpenSource/Modelio/wiki/Modelio-API (visited on 2023-02-27) (cit. on p. 104).

[61] B. Porter, J. v. Zyl, and O. Lamy. *Welcome to Apache Maven*. URL: https://maven.apache.org/ (visited on 2023-02-10) (cit. on p. 104).

[62] *JFace*. URL: https://wiki.eclipse.org/JFace (visited on 2023-02-27) (cit. on p. 104).

[63] *How to play sound with java*. URL: https://www.baeldung.com/java-play-sound (visited on 2023-02-27) (cit. on p. 105).

[64] *Freetts 1.2.3 - a speech synthesizer written entirely in the java(tm) programming language*. URL: https://freetts.sourceforge.io/ (visited on 2023-02-27) (cit. on p. 105).

[65] W. Walker, P. Lamere, and P. Kwok. *FreeTTS: a performance case study*. 2002 (cit. on p. 105).

[66] Numediart. *Welcome to MBROLA project*. URL: https://github.com/numediart/MBROLA (visited on 2023-02-27) (cit. on p. 105).

[67] ModelioOpenSource. *Junit dev basics*. URL: https://github.com/ModelioOpenSource/Modelio/wiki/Junit_dev_basics (visited on 2023-02-27) (cit. on p. 107).

[68] N. Guerreiro, V. Amaral, and M. Goulão. "Unleashing the Power of Sound: Revisiting the Physics of Notations for Modelling with auditory symbols". In: *arXiv preprint arXiv:2304.08654* (2023) (cit. on p. 154).

A

Appendix 1 - Detecting Patterns In UML Diagrammatic Readings: An Experimental Study

In this annex the materials that are relevant to the Chapter 6 are presented.

A.1 UML Diagrammatic Reading: Experimental Study

This experiment is being developed within the scope of the thesis "Constructing An Auditory Notation In Software Engineering - Editing UML Models with voice and sound", and aims to detect possible emerging auditory patterns in the way UML models are read.

To this end, **a total of 3 Class diagrams and 2 State diagrams will be presented**. In each of these, **you will be asked to analyze and verbally explain what you are observing** - as if you were explaining the diagram to someone else who is not seeing it (you can imagine it as if you were talking to someone on the phone). You can start by specifying from where you will start. Remember, there isn't a right or wrong answer to how you describe the diagrams, so don't feel pressured. Simply describe them in the best way you think it's possible!

The diagrams will be contained in this document, but in order to facilitate their analysis, the full-sized image of each of them, is also available within the .zip file.

Additionally, after this task, we'll ask you to answer some supplementary questions.

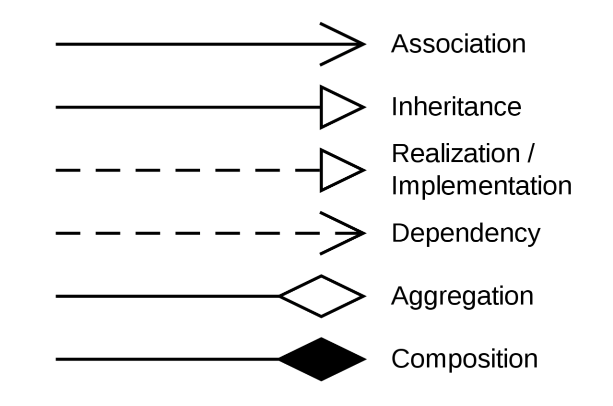
By agreeing to participate, you are granting your consent to record the audio of this experiment, in order to be transcribed into text and analyzed. The results of this study will be public but the data will be anonymized, guaranteeing total confidentiality.

In order to refresh your memory, **a brief summary of Class and State diagram notations is given below**, in the pages with a yellow background. While answering the questions, you can always go back and consult these pages.

Thank you for participating!

Class Diagrams Notation:

Relationships between classes:



Association - a relationship between two or more classifiers that involves connections among their instances;

Inheritance/Generalization - “is-a” relationship;

Realization/Implementation - implementation of a specification;

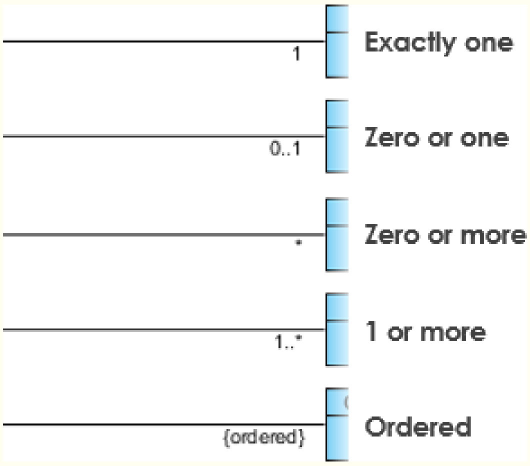
Dependency - directed relationship, a change to one modeling element (the independent element) will affect the other modeling element (the dependent element);

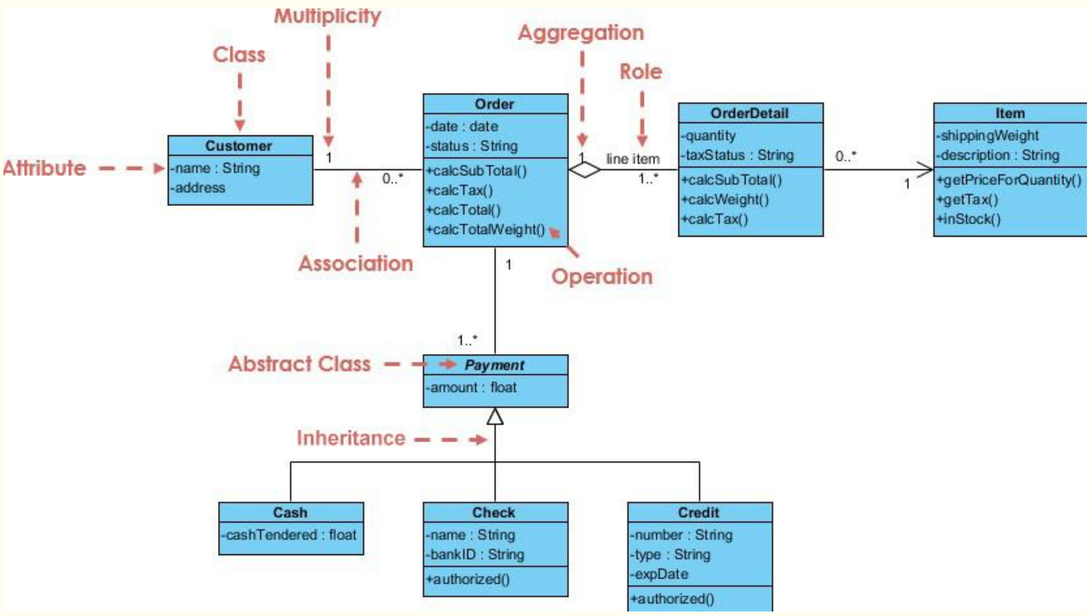
Aggregation - “has-a” or whole/part relationship (weak, destroying whole does not destroy parts);

Composition - strong form of aggregation (parts are destroyed along with the whole);

Composition example - Whenever a file is removed from the folder, the folder stays un-affected whereas the data related to that particular file is destroyed. If the folder is deleted, then it also affects all the files which are present inside the folder. All the files associated with the folder are automatically destroyed once the folder is removed from the system.

Cardinality:



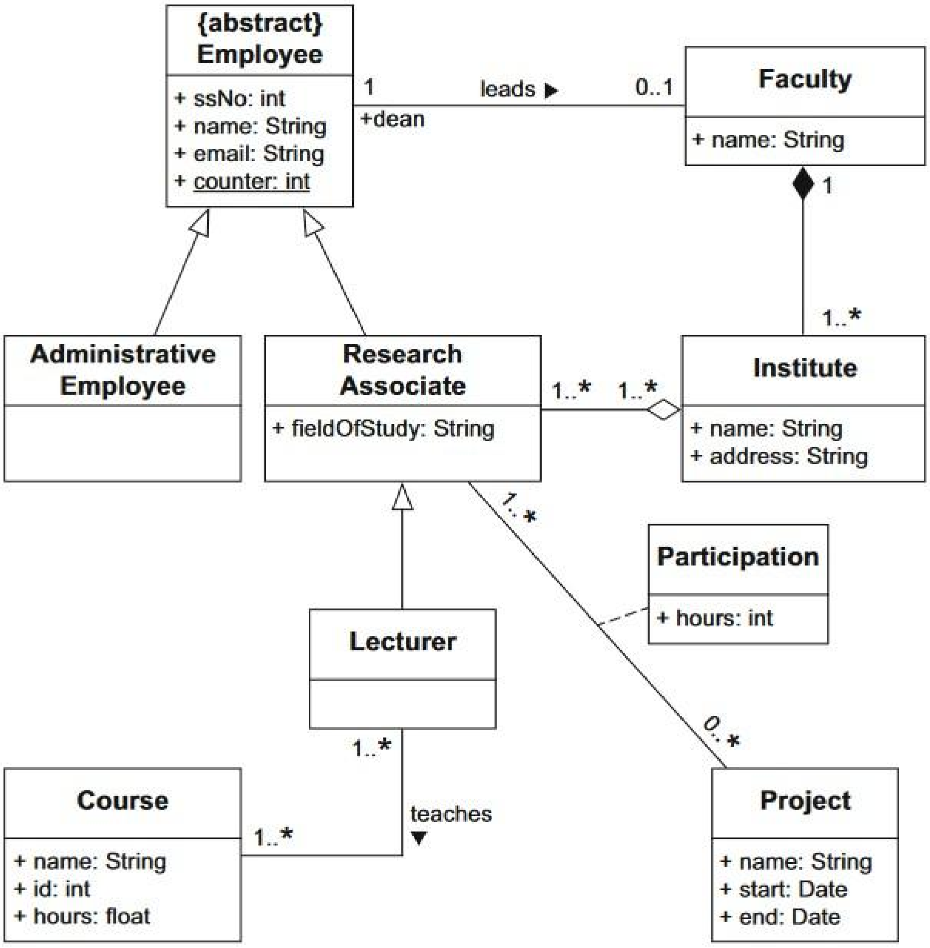


Sources:
https://www.goconqr.com/es/p/12265916?dont_count=true&frame=true&fs=true

https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/

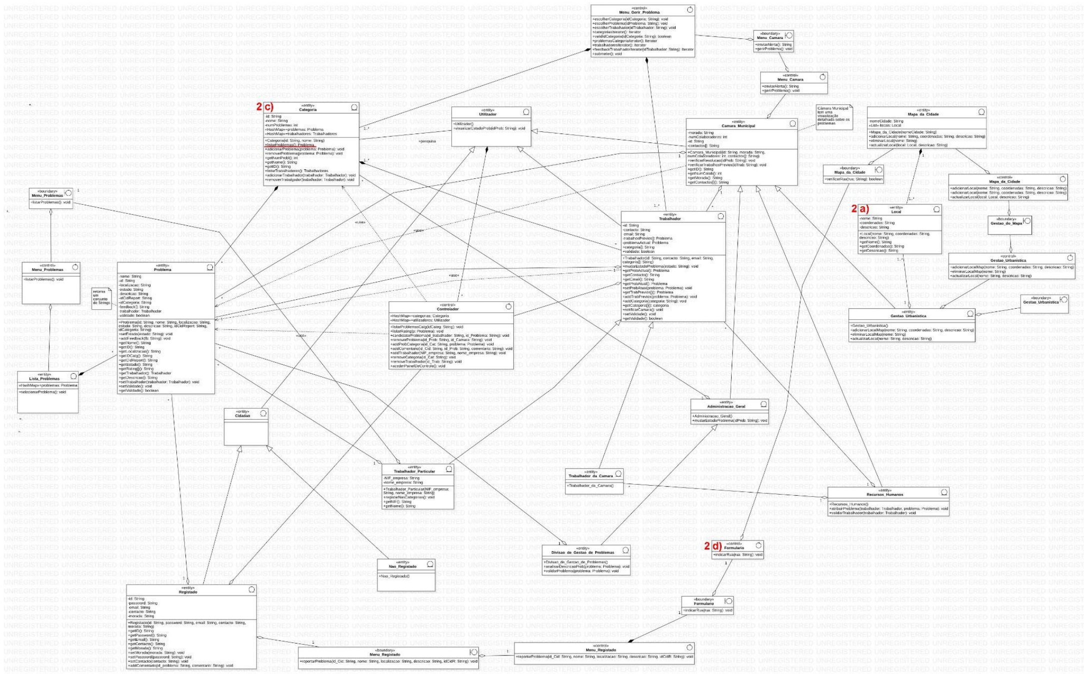
https://www.guru99.com/uml-relationships-with-example.html

Class Diagram no.1:



- Analyze and verbally explain the class diagram you are observing. You can start by specifying from where you will start, remember to be really descriptive and specific, as if you were explaining it to someone else who is not looking at the diagram. You don't need to describe the attributes or methods of each class.

Class Diagram no.2:

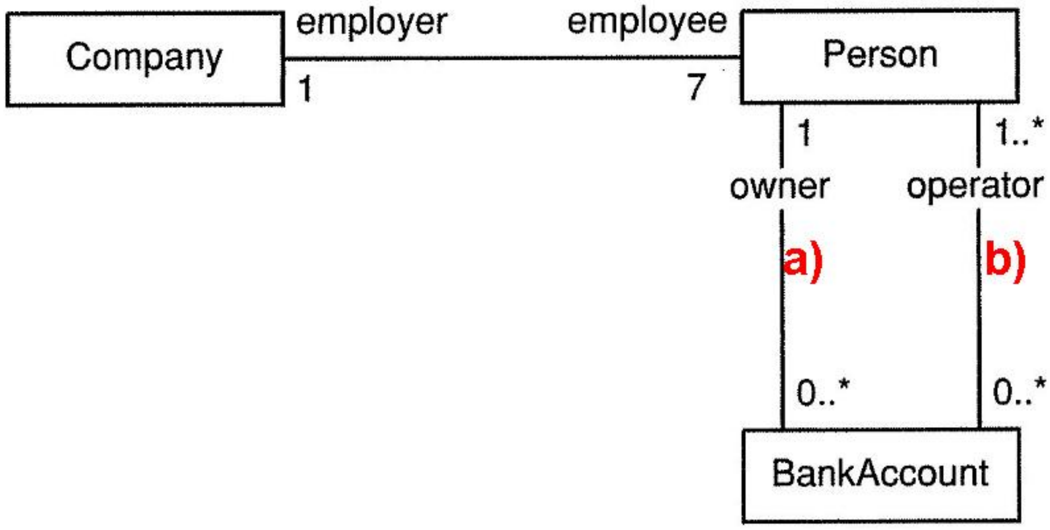


1. Analyze and verbally explain the class diagram you are observing. You can start by specifying from where you will start, remember to be really descriptive and specific, as if you were explaining it to someone else who is not looking at the diagram. You don't need to describe the attributes or methods of each class. **(The full sized image is included in the .zip)**

2. Imagine that you were asking someone to perform some operations on the diagram, what instructions would you give them so that they are able to accomplish the following tasks? Remember to be really specific, as if the person you are instructing doesn't have any context.

a) read class with the label "2 a)". You should describe the different attributes and methods, cardinalities, etc…
b) insert a new type of "Trabalhador". You should also describe what should be expected to happen.
c) update class with the label "2 c)" by removing the outlined method.
d) remove class with the label "2 d)"

Class diagram no.3:



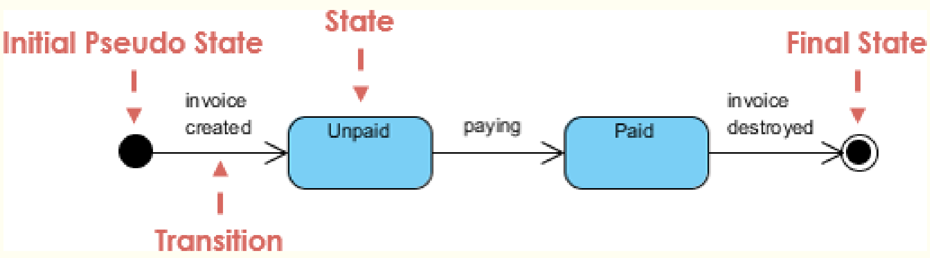
- Imagine that you were asking someone to perform some operations on the diagram. How would you specifically instruct them to remove the association with the label “a)”? How would you distinguish that association from the one with the label “b)”?

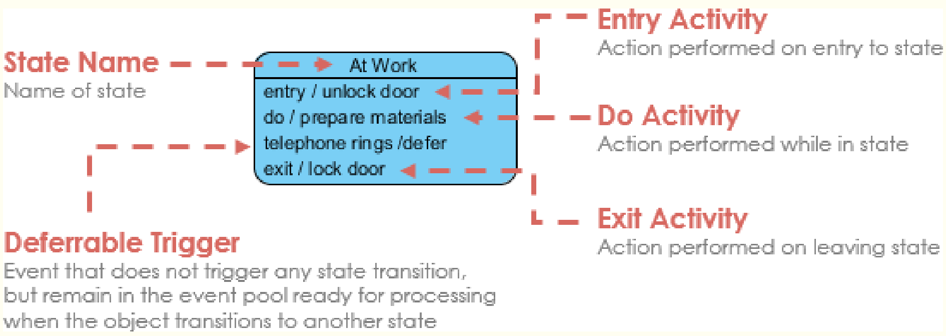
Additional Questions Regarding Class Diagrams:

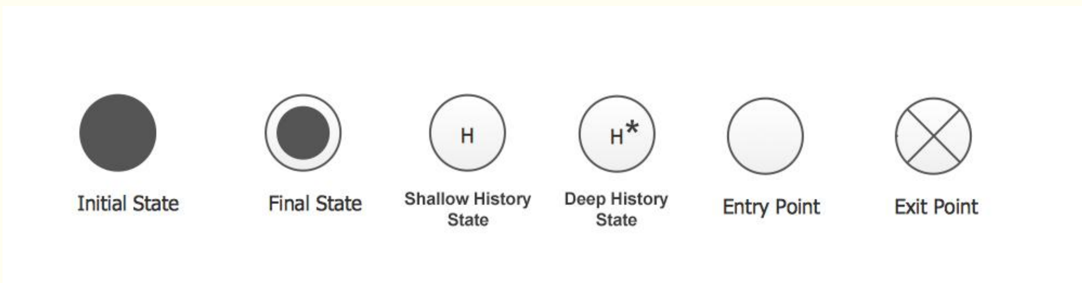
- Now that you have answered all of the questions and are more familiar with the class diagrams presented, would you change anything in the way you explained them?

- Would you like to give some additional information you find relevant regarding the class diagrams?

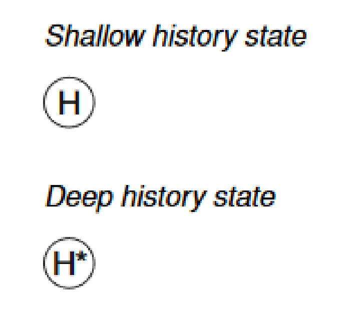
State Diagrams Notation:

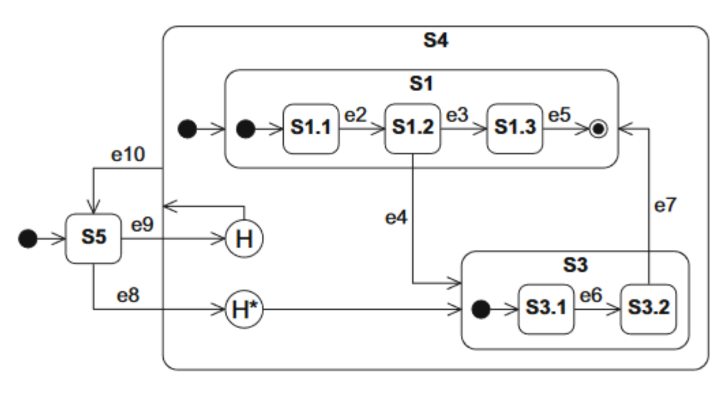






History States:

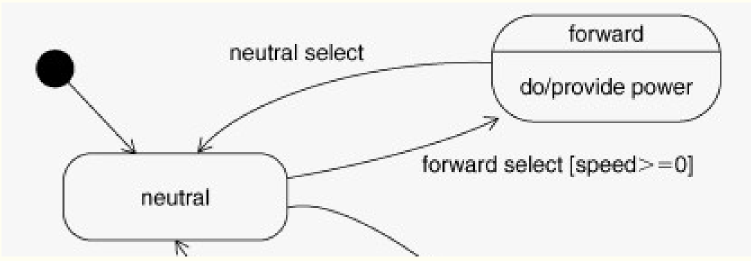




The **shallow history state** restores the state that is on the same level of the composite state as the shallow history state itself. In contrast, the **deep history state** notes the last active substate over the entire nesting depth.

Guard Conditions:

Represented by the usage of [ ], "Guard" is a condition which must be true in order for the trigger to cause the transition.



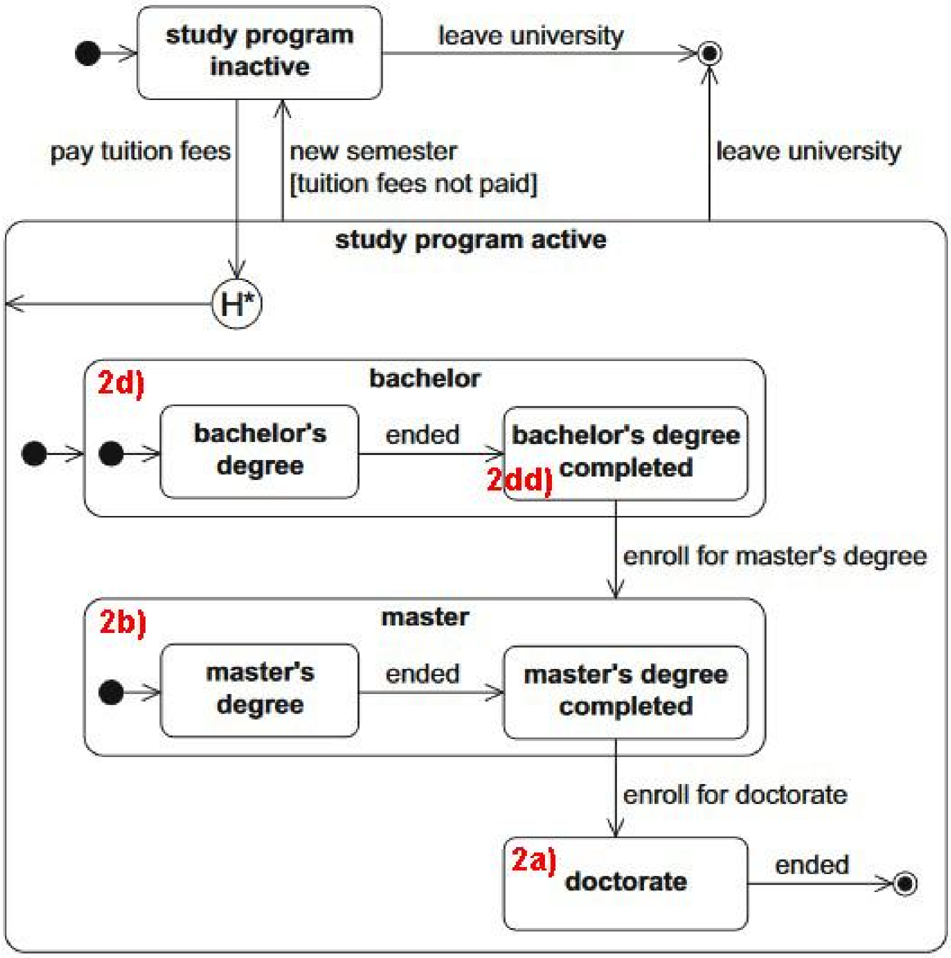
In this example, a *forward select* event triggers a transition to the *forward* state if the **guard condition** *[speed>=0]* is true.

—---

Sources:
https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-state-machine-diagram/

https://www.sciencedirect.com/topics/computer-science/state-machine-diagram (guard conditions)

State Diagram no.1:

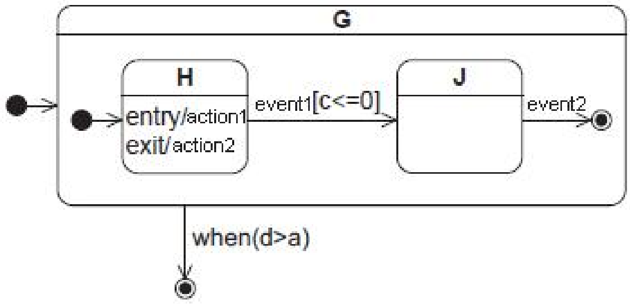


1. Analyze and verbally explain the state diagram you are observing. Remember to be really descriptive and specific, as if you were explaining it to someone else who is not looking at the diagram.

2. Imagine that you were asking someone to perform some operations on the diagram, what instructions would you give them so that they are able to accomplish the following tasks? Remember to be really specific, as if the person you are instructing doesn't have any context.

a) read/describe the state with the label "2 a)" in the diagram. Describe its context and transitions.
b) insert a new type of study program before the state with the label "2 b)". You should also describe what should be expected to happen.
c) remove state with the label "2 a)"
d) update "2d)" with a new state before the state with the label "2 dd)"

State diagram no.2:



- Analyze and verbally explain the state diagram you are observing. Remember to be really descriptive and specific, as if you were explaining it to someone else who is not looking at the diagram.

Additional Questions Regarding State Diagrams:

- Now that you have answered all of the questions and are more familiar with the state diagrams presented, would you change anything in the way you explained them?

- Would you like to give some additional information you find relevant regarding the state diagrams?

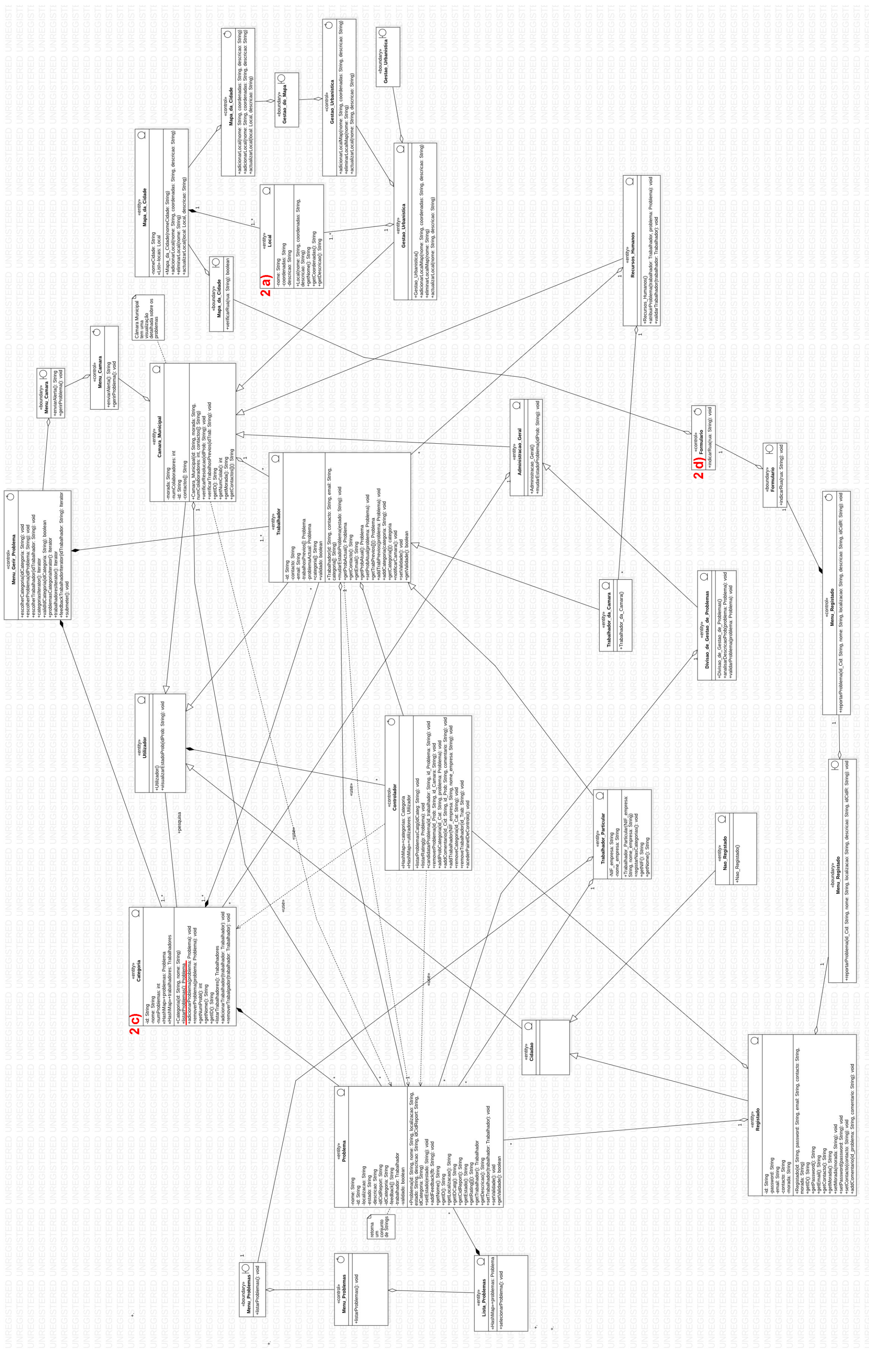


Figure A.1: Full sized class Diagram from Question 2.

B

Appendix 2 - A Catalogue Of Sounds For UML Class Diagrams

In this annex the materials that are relevant to the Chapter 7 are presented.

B.1 A Catalogue Of Sounds: Experimental Study

A Catalogue Of Sounds For UML Class Diagrams: Experimental Study

This experiment is being developed within the scope of the thesis "Constructing An Auditory Notation In Software Engineering - Understanding UML Models with voice and sound", and aims to determine if the construction of a catalogue of sounds that follows a more methodical approach is more effective for comprehension, when compared to a catalogue where the choice for sounds is purely arbitrary.

To this end, the different elements that make up a UML class diagram will be presented one by one, with a brief explanation, together with their visual representation and two different sounds that aim to represent, in an auditory manner, the concept that is being presented. You will then be asked to listen to each of the two sounds and choose the one that you think is more adequate to that concept. If you don't think any of them is adequate, you can also suggest your own. You can then explain briefly, in text, your decision, if you wish.

In order to test if you have sound working properly on your device, listen to the following sound by clicking the link:
TEST SOUND

If you could hear the alert sound, everything is working fine.

Thank you for participating!

*Obrigatório

Personal Information

1. Gender *

Marcar apenas uma oval.

- ◯ Male
- ◯ Female

2. Are you hard of hearing? *

Marcar apenas uma oval.

- ◯ Yes
- ◯ No

3. How do you consider your level of understanding of UML? *

Marcar apenas uma oval.

- Advanced
- Very Good
- Good
- Average
- Poor

4. When was your last contact with UML? *

Marcar apenas uma oval.

- I use it regularly
- I haven't used it for a year
- I haven't used it for 2 years or more

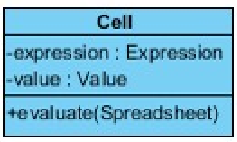
5. (IF YOU SELECTED "I USE IT REGULARLY") In what context do you use UML?

Marcar apenas uma oval.

- Professionally
- Academically

Element 01: Class

A class represents an object or a set of objects that share a common structure and behavior. They're represented by a rectangle that includes rows of the class name, its attributes, and its operations.



Listen to the following sounds:

[Sound A]

[Sound B]

6. Which of these sounds do you think better represents the concept of a Class? *

   Marcar apenas uma oval.

   - Sound A
   - Sound B
   - None

7. (OPTIONAL) What was the reason behind your choice? (Briefly explain your answer, if you wish)

8. (IF YOU SELECTED "NONE") In case you have answered "None", what do you think would be a better alternative and why?

Element 02: Attribute

Attributes represent the information, data, or properties that belong to instances of a class.

Listen to the following sounds:

Sound A

Sound B

9. Which of these sounds do you think better represents the concept of Attribute? *

Marcar apenas uma oval.

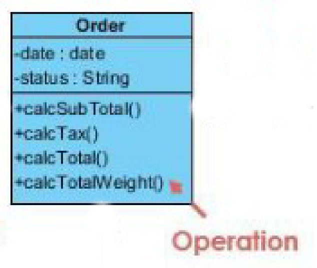
- Sound A
- Sound B
- None

10. (OPTIONAL) What was the reason behind your choice? (Briefly explain your answer, if you wish)

11. (IF YOU SELECTED "NONE") In case you have answered "None", what do you think would be a better alternative and why?

Element 03: Operation/Method

Operation is a behavioral feature that may be owned by an interface, or class.



Listen to the following sounds:

[Sound A]

[Sound B]

12. Which of these sounds do you think better represents the concept of Operation/Method? *

Marcar apenas uma oval.

- Sound A
- Sound B
- None

13. (OPTIONAL) What was the reason behind your choice? (Briefly explain your answer, if you wish)

14. (IF YOU SELECTED "NONE") In case you have answered "None", what do you think would be a better alternative and why?

Element 04: Association

A relationship between two or more classes that involves connections among their instances.

Listen to the following sounds:

[Sound A]

[Sound B]

15. Which of these sounds do you think better represents the concept of Association? *

Marcar apenas uma oval.

- Sound A
- Sound B
- None

16. (OPTIONAL) What was the reason behind your choice? (Briefly explain your answer, if you wish)

17. (IF YOU SELECTED "NONE") In case you have answered "None", what do you think would be a better alternative and why?

Element 05: Inheritance

“is-a” relationship.
The process of a child or sub-class taking on the functionality of a parent or superclass, also known as generalization.

It's symbolized with a straight connected line with a closed arrowhead pointing towards the superclass.

Listen to the following sounds:

Sound A

Sound B

18. Which of these sounds do you think better represents the concept of Inheritance? *

Marcar apenas uma oval.

- Sound A
- Sound B
- None

19. (OPTIONAL) What was the reason behind your choice? (Briefly explain your answer, if you wish)

20. (IF YOU SELECTED "NONE") In case you have answered "None", what do you think would be a better alternative and why?

Element 06: Realization/Implementation

Implementation of a specification.

Listen to the following sounds:

Sound A

Sound B

21. Which of these sounds do you think better represents the concept of Realization/Implementation? *

Marcar apenas uma oval.

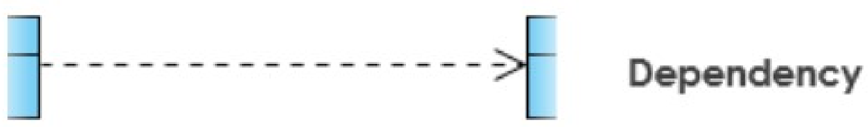
- Sound A
- Sound B
- None

22. (OPTIONAL) What was the reason behind your choice? (Briefly explain your answer, if you wish)

23. (IF YOU SELECTED "NONE") In case you have answered "None", what do you think would be a better alternative and why?

Element 07: Dependency

Directed relationship, a change to one modeling element (the independent element) will affect the other modeling element (the dependent element).



Listen to the following sounds:

Sound A

Sound B

24. Which of these sounds do you think better represents the concept of Dependency? *

Marcar apenas uma oval.

- Sound A
- Sound B
- None

25. (OPTIONAL) What was the reason behind your choice? (Briefly explain your answer, if you wish)

26. (IF YOU SELECTED "NONE") In case you have answered "None", what do you think would be a better alternative and why?

Element 08: Aggregation

An aggregation is a special type of association in which objects are assembled or configured together to create a more complex object. An aggregation describes a group of objects and how you interact with them. “has-a” or whole/part relationship (weak, destroying whole does not destroy parts).

Listen to the following sounds:

Sound A

Sound B

27. Which of these sounds do you think better represents the concept of Aggregation? *

Marcar apenas uma oval.

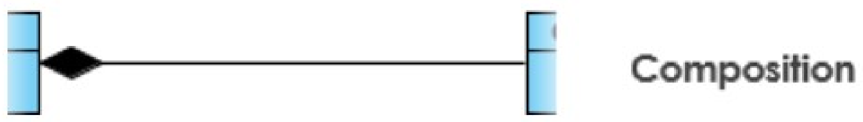
- Sound A
- Sound B
- None

28. (OPTIONAL) What was the reason behind your choice? (Briefly explain your answer, if you wish)

29. (IF YOU SELECTED "NONE") In case you have answered "None", what do you think would be a better alternative and why?

Element 09: Composition

A composition relationship specifies that the lifetime of the part class is dependent on the lifetime of the whole class. Strong form of aggregation (parts are destroyed along with the whole).



Listen to the following sounds:

Sound A

Sound B

30. Which of these sounds do you think better represents the concept of Composition? *

Marcar apenas uma oval.

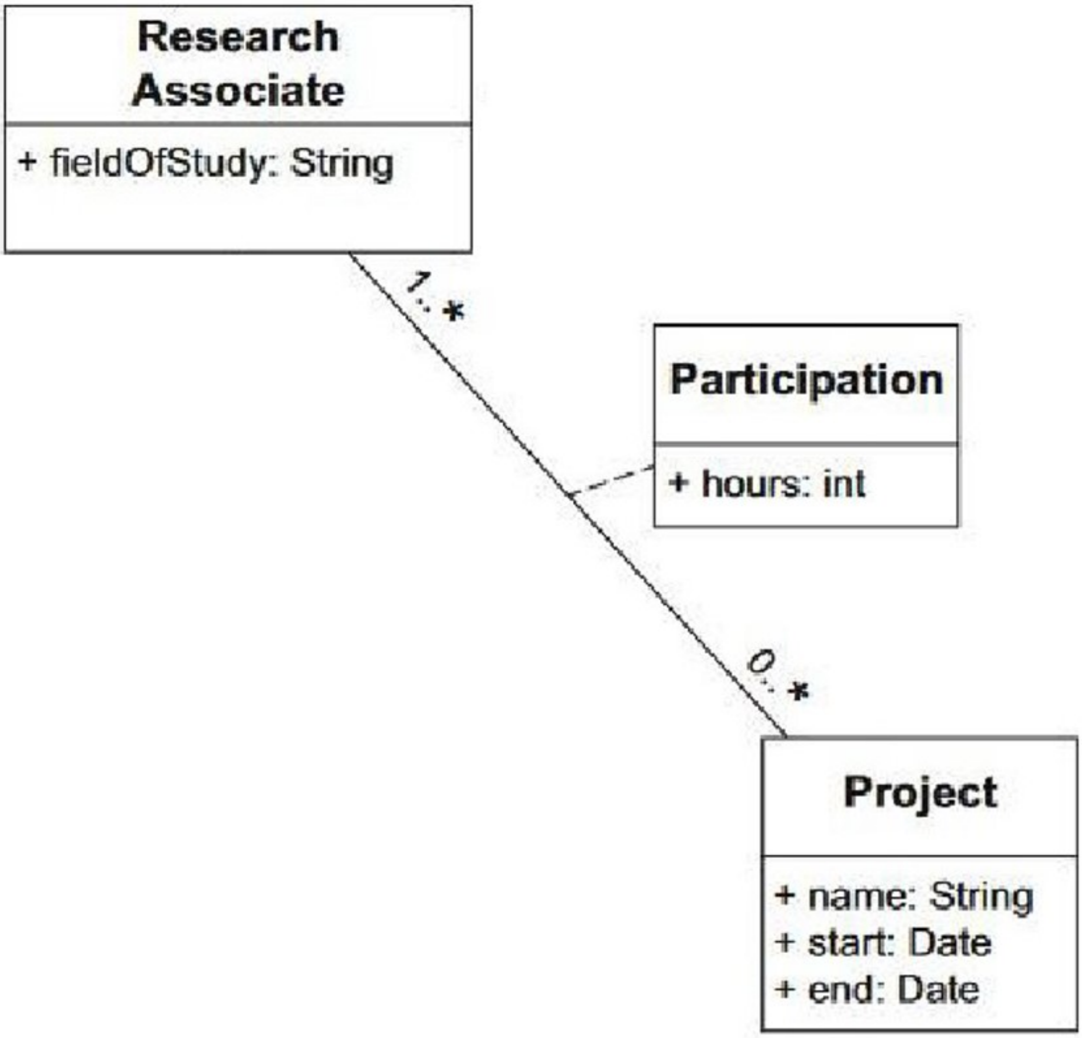
- Sound A
- Sound B
- None

31. (OPTIONAL) What was the reason behind your choice? (Briefly explain your answer, if you wish)

32. (IF YOU SELECTED "NONE") In case you have answered "None", what do you think would be a better alternative and why?

Element 10: Class Association

An association class is a class that is part of an association relationship between two other classes.



Listen to the following sounds:

Sound A

Sound B

33. Which of these sounds do you think better represents the concept of Class Association? *

Marcar apenas uma oval.

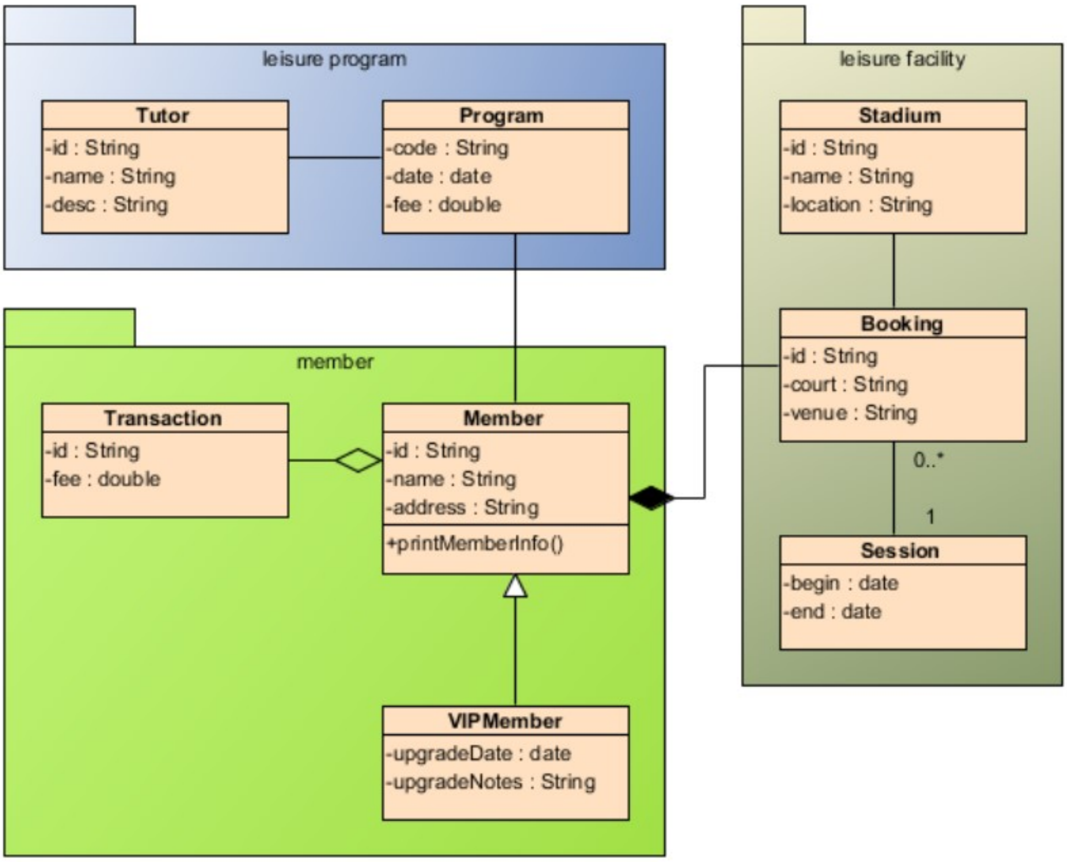
- Sound A
- Sound B
- None

34. (OPTIONAL) What was the reason behind your choice? (Briefly explain your answer, if you wish)

35. (IF YOU SELECTED "NONE") In case you have answered "None", what do you think would be a better alternative and why?

Element 11: Package

Organize related classes in a diagram. They are symbolized with a large tabbed rectangle shape.



Listen to the following sounds:

Sound A

Sound B

36. Which of these sounds do you think better represents the concept of Package? *

Marcar apenas uma oval.

- Sound A
- Sound B
- None

37. (OPTIONAL) What was the reason behind your choice? (Briefly explain your answer, if you wish)

38. (IF YOU SELECTED "NONE") In case you have answered "None", what do you think would be a better alternative and why?

Principles For Constructing Auditory Notations

The following table of principles for constructing auditory notations was formulated in the context of this thesis, with the purpose of helping create more intuitive and structured auditory notations. This table was based on previous work for visual notations. As such, it features for each of these principles, the initial definition but applied to sound.

The sounds present in this experiment were defined by following this table, and for each of the UML concepts here presented, one of the sounds followed the principles, while the other violated them.

Principle	Definition	Auditory Proposal
Semiotic Clarity	One-to-one correspondence between symbols and their referent concepts.	Each sound should represent only one concept. Meaning that each of the proposed sounds should only be played for only one concept.
Perceptual Discriminability	Different symbols should be clearly distinguishable from one another.	Different sounds should be clearly distinguishable from one another. For example, using different, unmistakable timbres for distinct sounds.
Semantic Transparency	Visual representations whose appearance suggests their meaning.	Auditory representations should suggest their meaning. For example, the sound of a baby crying to symbolise a dependency.
Complexity Management	Include explicit mechanisms to deal with the complexity of a diagram.	Noise of a container/box/door to signify "composition", supporting the concept of hierarchy. Usage of reverberation could be explored, a sound with more echo should signify a further away diagram.
Cognitive Integration	Include explicit mechanisms to support integration of information from different diagrams	Using auditory motifs that refer to a specific diagram as a way of contextualising it. Also using TTS to orientate the user.
Visual Expressiveness	Use full range and capacities of visual variables such as size, brightness, colour, texture, shape and orientation.	In this case, Auditory Expressiveness. Use the full range and capacities of auditory variables such as timbre, pitch, loudness, duration, reverberation, etc.
Dual Coding	Use text to complement graphics.	Certain sounds can be accompanied and complemented by a Text-to-speech voice.
Graphic Economy	Number of different graphical symbols should be cognitively manageable.	Finding the right balance between distinguishable sounds while being cognitively manageable.
Cognitive Fit	Use different visual dialects for different tasks and audiences.	Creating diagrams through the usage of voice can have both simplified commands and more complex ones for experienced users. Certain auditory cues conveying additional information can be omitted for less experienced users.

39. Rank the proposed principles by how relevant you think they are, with 1 being the least relevant, and 5 the most relevant *

Marcar apenas uma oval por linha.

	1 (Least Relevant)	2	3	4	5 (Totally Relevant)
Semiotic Clarity	⬭	⬭	⬭	⬭	⬭
Perceptual Discriminability	⬭	⬭	⬭	⬭	⬭
Semantic Transparency	⬭	⬭	⬭	⬭	⬭
Complexity Management	⬭	⬭	⬭	⬭	⬭
Cognitive Integration	⬭	⬭	⬭	⬭	⬭
Visual (Auditory) Expressiveness	⬭	⬭	⬭	⬭	⬭
Dual Coding	⬭	⬭	⬭	⬭	⬭
Graphic Economy	⬭	⬭	⬭	⬭	⬭
Cognitive Fit	⬭	⬭	⬭	⬭	⬭

40. After analyzing the table, do you think these proposed principles are overall relevant? *

Marcar apenas uma oval.

- ⬭ Strongly Disagree
- ⬭ Disagree
- ⬭ Neutral
- ⬭ Agree
- ⬭ Strongly Agree

41. (OPTIONAL) You can explain your answer, if you wish.

C

Appendix 3 - Validation In The Context Of Use

In this annex the materials that are relevant to the Chapter 8 are presented.

C.1 SoundUML Class Diagram

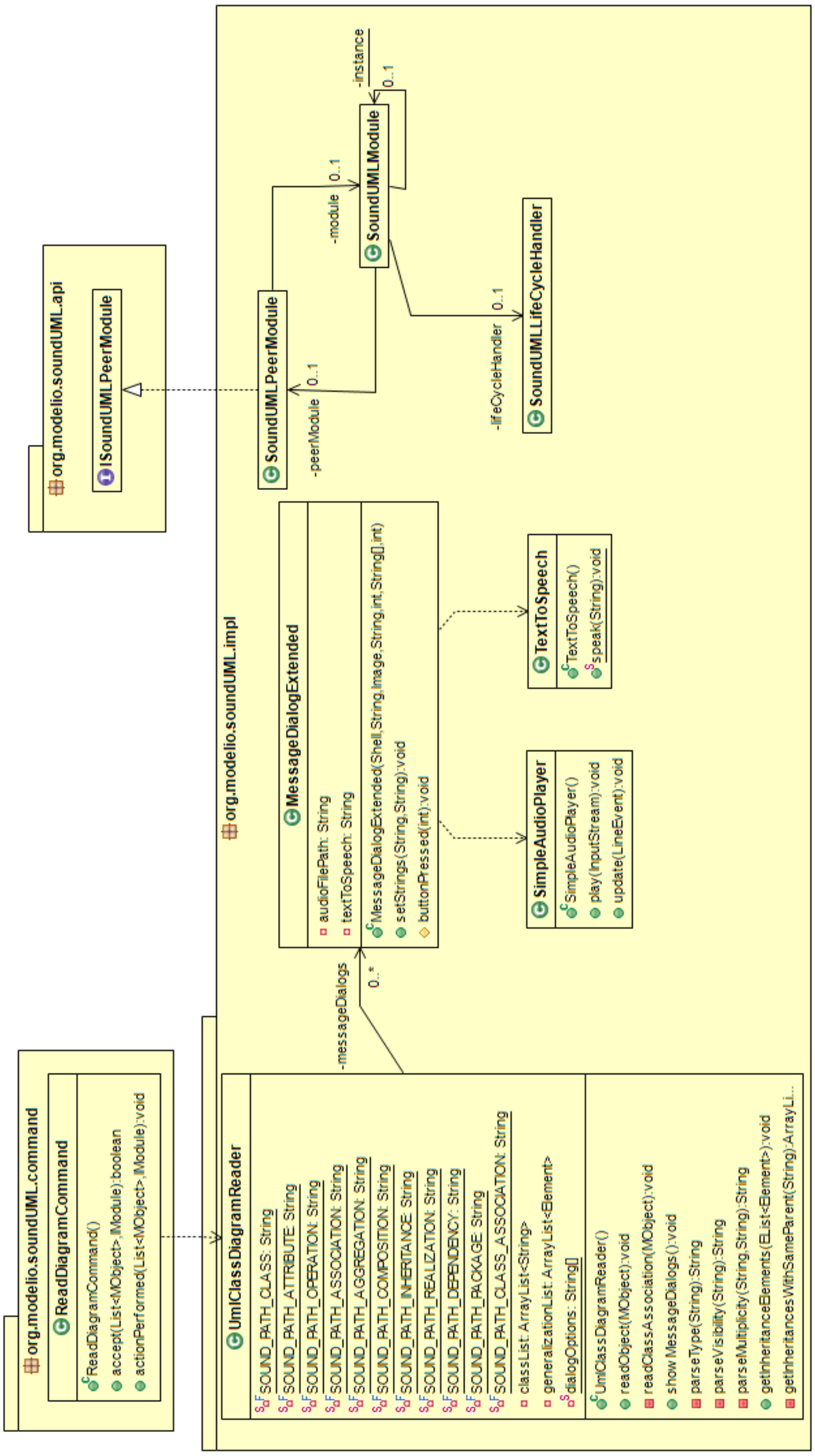


Figure C.1: Summarised SoundUML's Class Diagram, detailing the most important classes, their respective relations and packages.

C.2 UML Class Diagrams From The Experimental Study

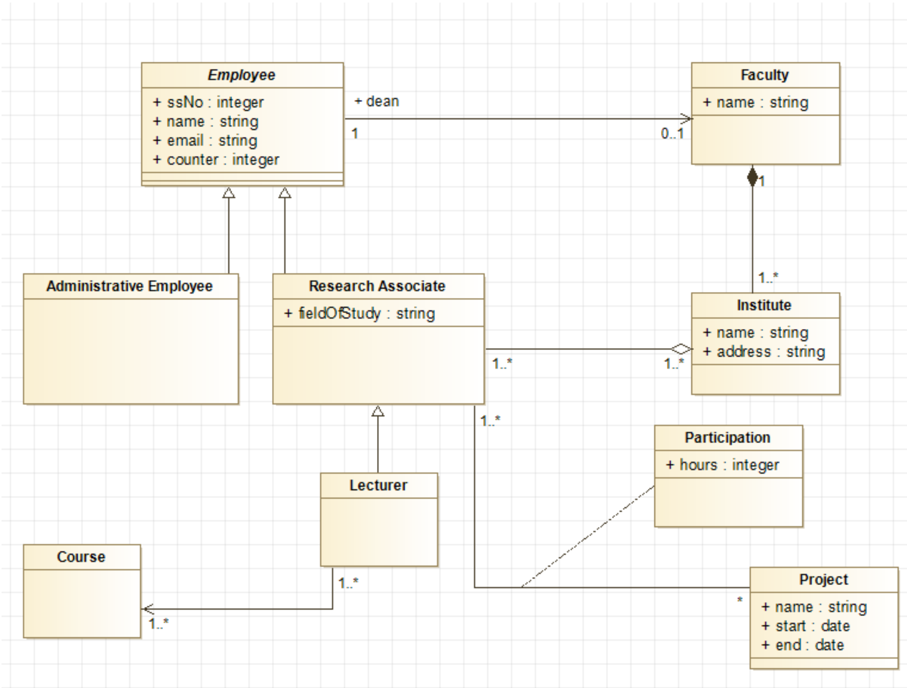


Figure C.2: Visual Version of Diagram A - Faculty, as constructed and presented in Modelio.

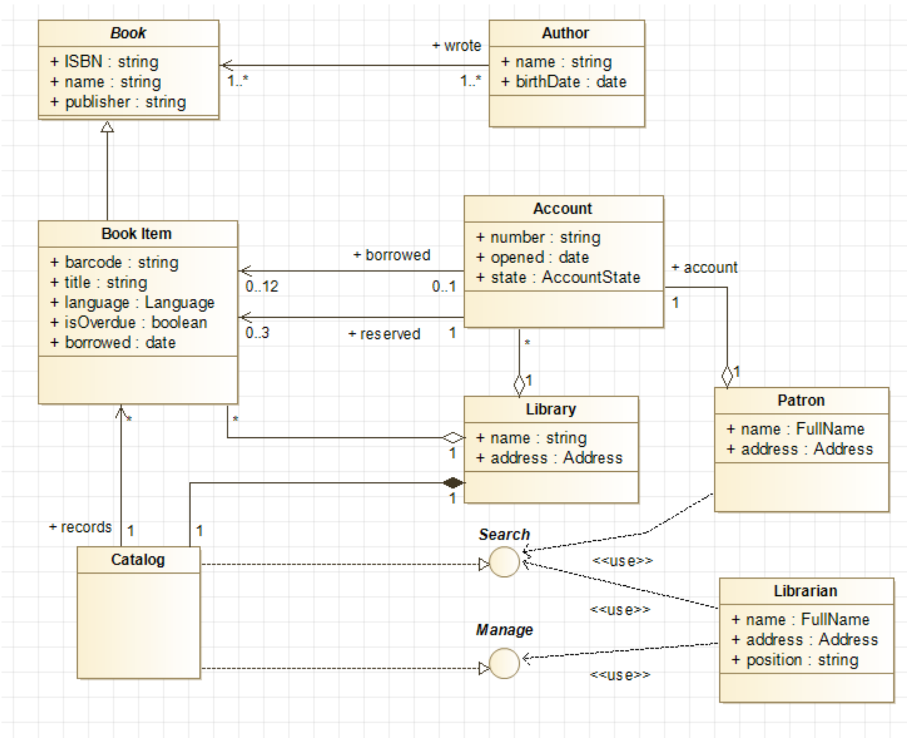


Figure C.3: Visual Version of Diagram B - Library, as constructed and presented in Modelio.

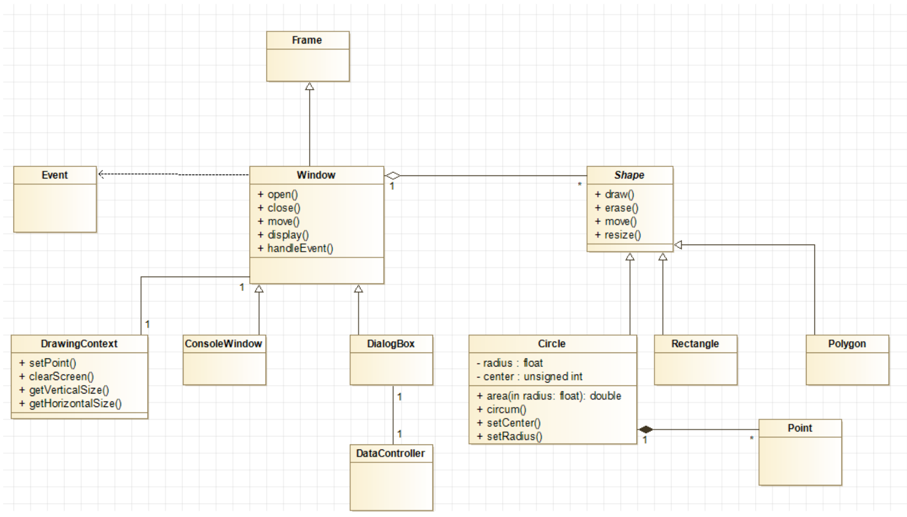


Figure C.4: Visual Version of Diagram C - GUI, as constructed and presented in Modelio.

C.3 Support Document

This experiment is being developed within the scope of the dissertation *"Constructing An Auditory Notation in Software Engineering: Understanding UML Models With Voice And Sound"*. The constructed tool, named *SoundUML*, allows UML class diagrams to be read to the users, through the usage of voice synthesis and sound playback. This solution was developed for the program Modelio, which is an open source UML modelling tool.

You will be asked to carry out the following tasks:

Task 1: Fill in the form regarding personal information. This form is intended to briefly ascertain the participants' level of understanding of the UML language.

Task 2: A total of 3 Class diagrams will be presented through Modelio, each in a unique manner: visual, our tool (auditory) and a combination of both. You have 5 minutes to analyse each, and afterwards, you will be asked to answer a few simple, multiple choice questions about some details of that specific diagram. The kind of questions that will be asked is, for example, the type of relationship between two classes, etc.

Task 3: Following task 2, you can use our developed tool, SoundUML, a little further, and test its functionalities. Afterwards, you will be asked to answer the form regarding the tool's Usability and Feedback. We should stress that you can be completely honest about what you think of the tool. This won't affect the appreciation of this work.

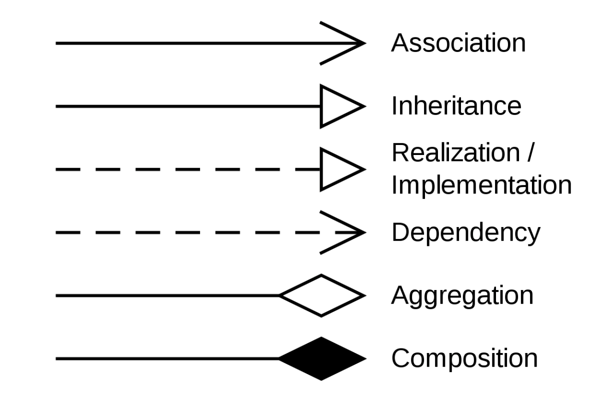
The forms will be filled in as the proposed tasks are carried out. By answering them, you allow the usage of the obtained data. The results of this study will be public but the data will be anonymized, guaranteeing total confidentiality.

In order to refresh your memory, **a brief summary of Class diagram notations is given below**, in the pages with a yellow background. You can always go back and consult these pages.

If you have any questions, feel free to ask.
Thank you for participating!

Class Diagrams Notation:

Relationships between classes:



Association - a relationship between two or more classifiers that involves connections among their instances;

Inheritance/Generalization - “is-a” relationship;

Realization/Implementation - implementation of a specification;

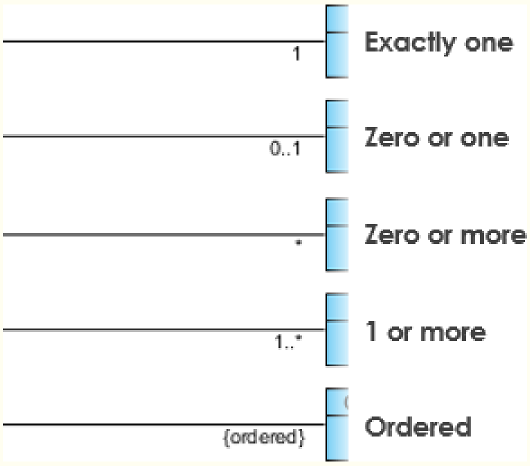
Dependency - directed relationship, a change to one modeling element (the independent element) will affect the other modeling element (the dependent element);

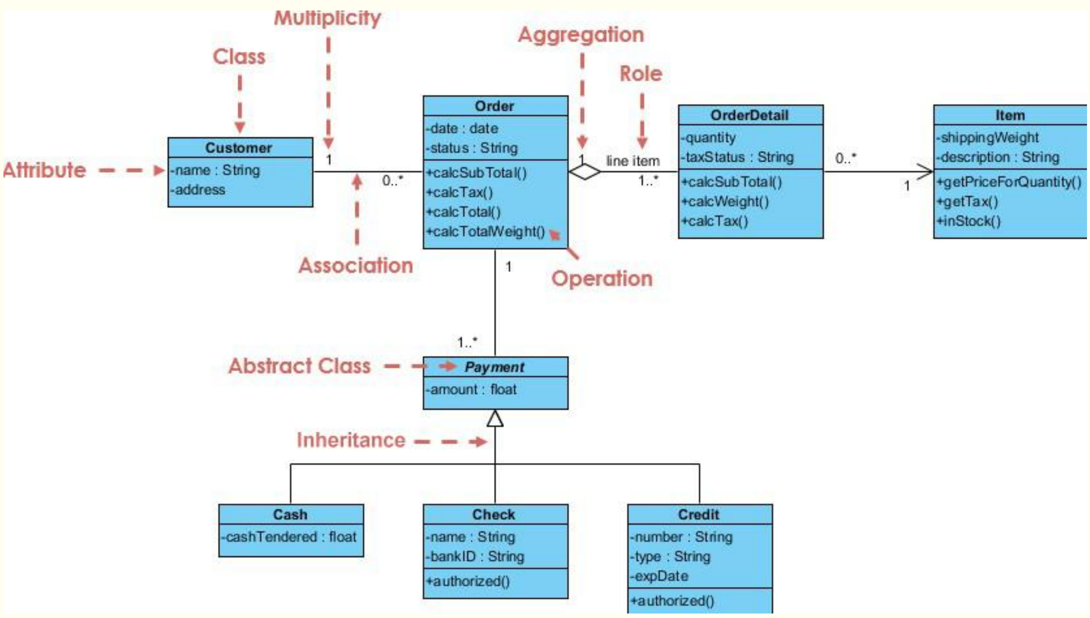
Aggregation - “has-a” or whole/part relationship (weak, destroying whole does not destroy parts);

Composition - strong form of aggregation (parts are destroyed along with the whole);

Composition example - Whenever a file is removed from the folder, the folder stays unaffected whereas the data related to that particular file is destroyed. If the folder is deleted, then it also affects all the files which are present inside the folder. All the files associated with the folder are automatically destroyed once the folder is removed from the system.

Cardinality:





———

Sources:
https://www.goconqr.com/es/p/12265916?dont_count=true&frame=true&fs=true

https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/

https://www.guru99.com/uml-relationships-with-example.html

C.4 Faculty Diagram Form

Faculty Diagram - SoundUML: Experimental Study

After observing the diagram Faculty, please answer the following questions.

*Required

1. For this specific diagram, what type of reading did you experience? *

   Mark only one oval.

   - Visual (Looking at the diagram) *Skip to question 2*
   - Tool (Auditory) *Skip to question 5*
   - Hybrid (Visual + Auditory) *Skip to question 8*

Visual (Looking at the diagram)

2. How many inheritance relationships exist in this diagram? *

   Mark only one oval.

   - 2
   - 3
   - 4

3. What is the type of relationship between classes Faculty and Institute? *

   Mark only one oval.

   - Association
   - Inheritance
   - Composition

4. What was the name of the only association class in the diagram? *

Mark only one oval.

- Participation
- Course
- Project

Tool (Auditory)

5. How many inheritance relationships exist in this diagram? *

Mark only one oval.

- 2
- 3
- 4

6. What is the type of relationship between classes Faculty and Institute? *

Mark only one oval.

- Association
- Inheritance
- Composition

7. What was the name of the only association class in the diagram? *

Mark only one oval.

- Participation
- Course
- Project

Hybrid (Visual + Auditory)

8. How many inheritance relationships exist in this diagram? *

Mark only one oval.

- 2
- 3
- 4

9. What is the type of relationship between classes Faculty and Institute? *

Mark only one oval.

- Association
- Inheritance
- Composition

10. What was the name of the only association class in the diagram? *

Mark only one oval.

- Participation
- Course
- Project

C.5 Library Diagram Form

Library Diagram - SoundUML: Experimental Study

After observing the diagram Library, please answer the following questions.

*Required

1. For this specific diagram, what type of reading did you experience? *

Mark only one oval.

- Visual (Looking at the diagram) *Skip to question 2*
- Tool (Auditory) *Skip to question 6*
- Hybrid (Visual + Auditory) *Skip to question 10*

Visual (Looking at the diagram)

2. How many association relationships exist between classes Book Item and Account? *

Mark only one oval.

- 1
- 2
- 3

3. What class implements the interfaces Manage and Search? *

Mark only one oval.

- Catalog
- Account
- Patron

4. What two classes have a dependency to the interfaces in the diagram? (The interfaces are Manage and Search) *

Mark only one oval.

- Author and Account
- Librarian and Patron
- Book Item and Patron

5. Which of these is NOT an attribute from class Book? *

Mark only one oval.

- ISBN
- releaseDate
- publisher

Tool (Auditory)

6. How many association relationships exist between classes Book Item and Account? *

Mark only one oval.

- 1
- 2
- 3

7. What class implements the interfaces Manage and Search? *

Mark only one oval.

- Catalog
- Account
- Patron

8. What two classes have a dependency to the interfaces in the diagram? (The interfaces are Manage and Search) *

Mark only one oval.

- ⬭ Author and Account
- ⬭ Librarian and Patron
- ⬭ Book Item and Patron

9. Which of these is NOT an attribute from class Book? *

Mark only one oval.

- ⬭ ISBN
- ⬭ releaseDate
- ⬭ publisher

Hybrid (Visual + Auditory)

10. How many association relationships exist between classes Book Item and Account? *

Mark only one oval.

- ⬭ 1
- ⬭ 2
- ⬭ 3

11. What class implements the interfaces Manage and Search? *

Mark only one oval.

- ⬭ Catalog
- ⬭ Account
- ⬭ Patron

12. What two classes have a dependency to the interfaces in the diagram? (The interfaces are Manage and Search) *

Mark only one oval.

- Author and Account
- Librarian and Patron
- Book Item and Patron

13. Which of these is NOT an attribute from class Book? *

Mark only one oval.

- ISBN
- releaseDate
- publisher

C.6 GUI Diagram Form

GUI Diagram - SoundUML: Experimental Study

After observing the diagram GUI, please answer the following questions.

*Required

1. For this specific diagram, what type of reading did you experience? *

   Mark only one oval.

   - Visual (Looking at the diagram) *Skip to question 2*
   - Tool (Auditory) *Skip to question 5*
   - Hybrid (Visual + Auditory) *Skip to question 8*

Visual (Looking at the diagram)

2. What is the type of relationship between classes Shape and Window? *

   Mark only one oval.

   - Aggregation
   - Dependency
   - Composition

3. Which of these classes does NOT exist in the diagram? *

   Mark only one oval.

   - Event
   - DrawingContext
   - Triangle

4. Which of these IS the name of an operation in class Circle? *

Mark only one oval.

- calcArc
- setPoint
- setCenter

Tool (Auditory)

5. What is the type of relationship between classes Shape and Window? *

Mark only one oval.

- Aggregation
- Dependency
- Composition

6. Which of these classes does NOT exist in the diagram? *

Mark only one oval.

- Event
- DrawingContext
- Triangle

7. Which of these IS the name of an operation in class Circle? *

Mark only one oval.

- calcArc
- setPoint
- setCenter

Hybrid (Visual + Auditory)

8. What is the type of relationship between classes Shape and Window? *

Mark only one oval.

- Aggregation
- Dependency
- Composition

9. Which of these classes does NOT exist in the diagram? *

Mark only one oval.

- Event
- DrawingContext
- Triangle

10. Which of these IS the name of an operation in class Circle? *

Mark only one oval.

- calcArc
- setPoint
- setCenter

C.7 Personal Info + Usability + General Feedback Form

SoundUML - A Tool For UML Diagrammatic Reading: Experimental Study

This experiment is being developed within the scope of the dissertation *"Constructing An Auditory Notation in Software Engineering: Understanding UML Models With Voice And Sound"*. The constructed tool, named *SoundUML*, allows UML class diagrams to be read to the users, through the usage of voice synthesis and sound playback. This solution was developed for the program Modelio, which is an open source UML modelling tool.

This reading is based on two experimental studies that we have conducted during the dissertation. The first study was aimed at detecting any emerging patterns regarding how people read UML Class and state diagrams. The second consisted in creating two catalogues of sounds for the different elements that constitute a UML Class diagram, and comparing them to find which sounds were the most suitable, in order to achieve greater efficacy regarding the perception of different sounds on behalf of users.

The form will be filled in as the proposed tasks are carried out. By answering it, you allow the use of the data obtained.

Thank you for participating!

*Required

Personal Information

1. Gender *

Mark only one oval.

- Male
- Female

2. Are you hard of hearing? *

Mark only one oval.

- Yes
- No

3. How do you consider your level of understanding of UML? *

Mark only one oval.

- Advanced
- Very Good
- Good
- Average
- Poor

4. When was your last contact with UML? *

Mark only one oval.

- I use it regularly
- I haven't used it for a year
- I haven't used it for 2 years or more

SoundUML - Usability

5. I think that I would like to use this system frequently. *

Mark only one oval.

	Strongly Disagree
1	◯
2	◯
3	◯
4	◯
5	◯
	Strongly Agree

6. I found the system unnecessarily complex. *

Mark only one oval.

	Strongly Disagree
1	◯
2	◯
3	◯
4	◯
5	◯
	Strongly Agree

7. I thought the system was easy to use. *

Mark only one oval.

Strongly Disagree

1 ⬭

2 ⬭

3 ⬭

4 ⬭

5 ⬭

Strongly Agree

8. I think that I would need the support of a technical person to be able to use this system. *

Mark only one oval.

	Strongly Disagree
1	⬭
2	⬭
3	⬭
4	⬭
5	⬭
	Strongly Agree

9. I found the various functions in this system were well integrated. *

Mark only one oval.

Strongly Disagree	
1	◯
2	◯
3	◯
4	◯
5	◯
Strongly Agree	

10. I thought there was too much inconsistency in this system. *

Mark only one oval.

Strongly Disagree	
1	◯
2	◯
3	◯
4	◯
5	◯
Strongly Agree	

11. I would imagine that most people would learn to use this system very quickly. *

Mark only one oval.

Strongly Disagree
1
2
3
4
5
Strongly Agree

12. I found the system very cumbersome to use. *

Mark only one oval.

Strongly Disagree
1
2
3
4
5
Strongly Agree

13. I felt very confident using the system. *

Mark only one oval.

	Strongly Disagree
1	◯
2	◯
3	◯
4	◯
5	◯
	Strongly Agree

14. I needed to learn a lot of things before I could get going with this system. *

Mark only one oval.

	Strongly Disagree
1	◯
2	◯
3	◯
4	◯
5	◯
	Strongly Agree

Sound UML - General Feedback

15. How would you rate your experience with Sound UML? *

Mark only one oval.

- Very Poor
- Poor
- Average
- Good
- Excellent

16. Do you think the diagrammatic readings performed by Sound UML were intuitive enough to understand the diagrams? *

Mark only one oval.

- No
- Yes

17. What type of reading did you prefer? *

Mark only one oval.

- Visual (Looking at the diagram)
- Tool (Auditory)
- Hybrid (Visual + Auditory)

18. (OPTIONAL) Which aspects of Sound UML did you consider to be the strongest?

19. (OPTIONAL) Which aspects of Sound UML did you considered to be the weakest?

20. (OPTIONAL) What changes would you make to Sound UML?

21. Do you think SoundUML could be useful to complement and enrich the existing visual notation? *

Mark only one oval.

- No
- Yes

22. Do you think SoundUML could be useful in the future for visually impaired people who want to carry out UML modelling? *

Mark only one oval.

- Don't know
- No
- Maybe
- Yes

23. Would you recommend SoundUML to visually impaired people? *

Mark only one oval.

- No
- Yes

D

Appendix 4 - Scientific Paper

In this annex we present the scientific paper which adapts the contents of this document up to Chapter 7, submitted to the MODELS 23 conference.

Unleashing the Power of Sound: Revisiting the Physics of Notations for Modelling with auditory symbols

Nuno Guerreiro
NOVA LINCS, FCT NOVA University
Lisbon, Portugal
n.guerreiro@campus.fct.unl.pt

Vasco Amaral
NOVA LINCS, FCT NOVA University
Lisbon, Portugal
0000-0003-3791-5151

Miguel Goulão
NOVA LINCS, FCT NOVA University
Lisbon, Portugal
000-0002-5356-5203

***Abstract*—Sound - the oft-neglected sense for Software Engineering - is a crucial component of our daily lives, playing a vital role in how we interact with the world around us. In this paper, we challenge the traditional boundaries of Software Engineering by proposing a new approach based on sound design for using sound in modelling tools that is on par with visual design. By drawing upon the seminal work of Moody on the 'Physics' of Notations for visual design, we develop a comprehensive catalogue of principles that can guide the design of sound notations.**

Using these principles, we develop a catalogue of sounds for UML and report on an empirical study that supports their usefulness. Our study lays the foundation for building more sophisticated sound-based notations. The guidelines for designing symbolic sounds for software models are an essential starting point for a new research thread that could significantly and effectively enable the use of sound in modelling tools.

***Index Terms*—Human Factors in Modelling, Sound symbology, Modelling with Sound, Sound Notations, UML**

I. Introduction

Visual notations are well accepted in the Computer Science field [1]. However, the tools that accommodate this diagrammatic communication do not support sound in the context of said notations. This relative lack of interest regarding sound potentially leads to the exclusion of many visually impaired or blind people from software development-related activities, namely software modelling with visual notations. One such example is UML. Created to standardise notational systems in software design, UML is a general-purpose, widely adopted modelling language and, as such, an integral part of the software development process [2].

We argue that sound should be treated in the context of Software Engineering as it is already done with the visual approaches. It can potentially contribute to modelling tools, significantly increasing productivity and accessibility.

To assemble an effective interface and assign meaning to sound, we need to understand what influence different acoustic cues have through their cataloguing and systematisation. Organised sounds, such as music, play a role in memory creation and recollection, and auditory cues influence human behaviour [3]. This is relevant to our work, as we intend to achieve a user-friendly interface that does not require much effort from its users. As noted by Nielsen, humans have limited short-term memories [4]. As such, interfaces that promote recognition reduce the cognitive effort required from users. He suggests *"Recognition rather than recall"* through his heuristics, meaning the user should not have to remember information from one part of the interface to another. Approaching this problem from a purely technological standpoint by developing tools without considering the fundamental part of establishing and structuring good symbolism for sounds will likely result in misinformed choices about how sound is used.

In the visual realm, the most popular and comprehensive recommendation for assuring cognitively effective notations [1] is *The "Physics" of Notations* (PoN) [5]. PoN suggests nine principles based on the perceptual properties of notations, addressing and raising awareness about the importance of specific visual representation issues that software engineering researchers and notation designers have overlooked.

Our current focus lies in establishing a solid groundwork to guide decision-making in creating an auditory notation. We leverage knowledge of music symbology and understanding the semiotics of the audible field, combined with the insights provided by PoN and other relevant research and tools. Our goal is to create a framework that can maximise the benefits of graphical notation for experienced software engineers and novices when combined with visual cues. Moreover, this approach presents the opportunity to model activities using only the aural sense, thereby promoting greater accessibility and inclusion for individuals with visual impairments in modelling activities.

In this paper we introduce semiotics (Section II) and transpose Moody's work on graphical notational physics to sound (Section III). Then, we review existing literature to establish design guidelines (Section IV). We then present our first proposal for a sound catalogue for UML Class Diagrams (Section V) and report on an experimental study for its validation (Section VI) before concluding (Section VII).

II. Semiotics

To assemble an effective notation, we need to understand the influence different acoustic cues have on assigning meaning to sound. Semiotics is the study of any activity that involves the usage of signs and symbols and their signification. Generally, signs can be understood as a "stand-in" representation of a particular concept. Examples of signs include *emojis* used in electronic communication, traffic lights and logos. Signs can also be drawings, paintings, photographs, words, sounds, or body language. The ones receiving the information need background knowledge to bridge the gap between the sign itself and the concept that it is meant to represent.

In Semiotic Theory, the sign is made up of the relation between three components: the **(semiotic) object** - what the sign represents/encodes, and can be anything thinkable, for example, a fact or a law; the **signifier** - Also called the representamen, is the form that represents the denoted *object*, giving its meaning. It can be, for example, a word or image and is also sometimes called representamen. Finally, the **signified** - also called interpretant, is the concept that a *signifier* refers to. It is what is evoked in the mind - a mental concept [6]. Semioticians differentiate amongst various types of signs, emphasising that signs differ in how arbitrary/conventional they are [7]. Regarding the signifier-object relations and how the signified denotes the object, we find philosopher Charles Peirce's second trichotomy [8] describing the different ways the sign refers to its object. As such, a sign can be an **icon** (firstness) - where the sign is perceived as resembling or imitating the object, being similar in having some of its qualities. Examples of icons are sound effects in radio drama, a portrait or a scale model.

A sign can also be an **index** (secondness) - a mode in which the sign is affected by the object, the link between the signifier and signified can be observed or inferred, with both being directly connected in some way. This can be ascertained in examples like 'natural signs' - such as smoke, thunder, footprints or echoes; medical symptoms - pain, a rash, pulse rate, or 'signals' - like a knock on a door or a phone ringing.

Based on this brief theoretical explanation, along with the foundational work elaborated in Chandler's *Semiotics: The Basics* [6] and *Conceptualization, measurement, and application of semantic transparency in visual notations* [1], Table I details each of these terms, a concise semiotic meaning, a popular language meaning (how people colloquially perceive each of the terms), and a concrete example.

III. Principles for an Auditory Notation

In this section, we transpose the principles proposed by Moody in a manner analogous to sound. We propose Table II featuring the definition stated by Moody, this time applied to sound for each of these principles. The "translation" of each concept is not always direct, so the reasoning for each correspondence will be explained afterwards.

In the case of **Semiotic Clarity**, each sound should represent only one concept. This means that each proposed sound should be reserved for only one concept.

Different sounds should be distinguishable for the principle of **Perceptual Discriminability**. In the visual field, shapes play a special role in discriminating between symbols, representing the primary basis for identifying objects in the real world. Thinking about the characteristics that constitute a sound, the timbre (or "tone colour") can be considered its "shape", as it is a form of distinguishing different objects. Sounds should have various and unmistakable timbres to be recognisable. For example, a sound of an arrow being shot can be used for an *association* in a UML Class diagram, while a baby crying can be used to describe a *dependency*.

The successful exploration of this principle at the auditory level could potentially be more complex than its visual counterpart, namely when finding outstandingly intuitive sounds for "abstract" concepts. The auditory representations should suggest their meaning in **Semantic Transparency**. Neglecting this principle could result in semantically opaque sounds, with their meaning being purely arbitrary. Ideally, the sounds should be semantically immediate. Still, if they are semantically translucent (as mentioned above, the example of the baby crying symbolises *dependence*), it would already be a great starting point for this research.

For the principle of **Complexity Management**, explicit mechanisms that deal with the complexity of a diagram should be included. For example, the noise of a container/box/door can signify "composition" and support the concept of a hierarchy. Reverberation could also be explored: a sound with more echo to signify a diagram that is further away.

Cognitive integration must include explicit mechanisms to support integrating information from different diagrams. Using sound motifs to summarise the concepts of a diagram, Cognitive integration can be included. Sound motifs and text-to-speech (TTS) voice synthesis can also help contextualise directly related elements from other diagrams as foreign elements. Perceptual integration could be established using TTS, allowing for clear diagram labelling and easier way-finding on the users' behalf.

The principle of **Visual Expressiveness** can be considered "Auditory Expressiveness", where auditory variables' full range and capacities are used. Krygier defines these variables as location, loudness, pitch, register, timbre, duration, rate of change, order and attack/decay [9]. Besides these variables, reverberation can be helpful, as discussed in the principle of Complexity Management, or even panning the sound to the left or right channel.

Dual Coding can be directly translated to an auditory approach. Text can be used to complement graphics in visual notation. A TTS voice can accompany certain sounds, reinforcing meaning as an additional cue to a given concept.

The principle of **Graphic Economy** becomes "Auditory Economy". Just as graphical symbols should be cognitively manageable, so should the auditory ones, as these might be harder to assimilate in a first approach. One solution to this problem might be to increase auditory expressiveness by using multiple auditory variables (described in the principle of visual expressiveness) to differentiate between symbols. The right balance between distinguishable sounds while being cognitively manageable should be found.

Cognitive Fit suggests the usage of different visual dialects when drawing and communicating certain aspects of a diagram. For an auditory approach, creating diagrams through voice can support simplified and more complex commands for novice and experienced users. While interpreting a diagram, certain auditory cues conveying additional information can also be omitted for less experienced users.

TABLE I
MEANING OF TRIADIC SIGN TERMS IN SEMIOTIC THEORY AND POPULAR LANGUAGE

Term	Semiotic Meaning	Popular language meaning	Example
Signifier	Is the form the sign takes, giving it meaning.	The terms ”representation” and ”depiction” are used as synonyms.	Drawing of a red heart
Signified	Is the concept that the signifier refers to.	The terms ”concept” and ”definition” are used as synonyms.	A symbol of love and affection (Signifies the red heart)
Sign	The unity of what is represented (the object), how it is represented (signifier), and how it is interpreted (signified).	Object, quality, or event whose presence or occurrence indicates something else’s probable presence or occurrence.	Logos, Traffic lights

TABLE II
THE NINE PRINCIPLES PROPOSED BY MOODY WITH OUR EQUIVALENT AUDITORY PROPOSAL.

Principle	Definition	Auditory Proposal
Semiotic Clarity	One-to-one correspondence between symbols and their referent concepts.	Each sound must have a single concept assigned to it, indicating that each proposed sound should be played for only one concept.
Perceptual Discriminability	Different symbols should be clearly distinguishable from one another.	Different sounds should be clearly distinguishable from one another. For example, using different, unmistakable timbres for distinct sounds.
Semantic Transparency	Visual representations whose appearance suggests their meaning.	Auditory representations should suggest their meaning. For example, the sound of a baby crying symbolises dependency.
Complexity Management	Include explicit mechanisms to deal with the complexity of a diagram.	Noise of a container/box/door to signify “composition”, supporting the concept of hierarchy. Usage of reverberation could be explored, a sound with more echo should signify a further away diagram.
Cognitive Integration	Include explicit mechanisms to support integration of information from different diagrams	Using auditory motifs that refer to a specific diagram as a way of contextualising it. In addition, it uses a text-to-speech component to guide the user.
Visual Expressiveness	Use full range and capacities of visual variables such as size, brightness, colour, texture, shape and orientation.	Auditory Expressiveness. Use the full range and capacities of auditory variables such as timbre, pitch, loudness, duration, reverberation, etc.
Dual Coding	Use text to complement graphics.	Certain sounds can be accompanied and complemented by a text-to-speech voice.
Graphic Economy	Number of different sound symbols should be cognitively manageable.	Finding the right balance between distinguishable sounds while being cognitively manageable.
Cognitive Fit	Use different visual dialects for different tasks and audiences.	Creating diagrams using voice commands. Simple commands work for beginners, while experienced users can use more complex ones. Additional auditory cues may not be necessary for less experienced users.

IV. DESIGN GUIDELINES

A. Guidelines For Sound Design In The Context Of Software

Blauert [10] argues that product sound design was initially considered unimportant and often treated as unwanted noise. Yet, he emphasises that product sound carries valuable information and can significantly improve the perceived quality of a product. Understanding the positive associations that certain sounds can create for a product is a key task in sound design. Despite the involvement of multiple sound designers in product planning, sound design still heavily relies on intuition rather than rational and systematic approaches. As a result, Blauert proposed various criteria for product-sound design:

- Motivation for product use - why is the product used and what does the user gain from it?
- Function of the acoustic component in the product context - is sound important to the product or just secondary?
- Function of the acoustic component in relation to other product components - are the sounds suitable for the product’s use?
- Meaning associated with product use - how does the product meet user expectations?
- Dominant quality items of the product’s sounds - how are the sounds judged?(e.g. innovative, simple, complex)
- Provoked reactive behaviour of the user as to product quality - how do users emotionally react to the product?
- User specification - can a typical user be specified, based on age, education, experience, or expectations?

Blauert [10] believes that sound should enhance the product and meet users’ needs. However, sound design has not received the same attention as visual design. Therefore, there is potential for more innovative sound design solutions.

Hussein et al. [11] propose guidelines to address the challenges of combining visualisation and sonification in a single comprehension tool. They are as follows:

- Add sonification to simplify visualisations - For example, a Call Hierarchy View in Eclipse. On a given method, Eclipse generates a tree visualisation of a call graph upon this request. However, this provides no information about the depth of the generated call graph. Adding a special visual cue that can show the depth of the tree is likely to clutter the tree visualisation. By contrast, adding a sonic cue representing the tree’s depth via a different volume or pitch is relatively straightforward, minimising the required interaction time.
- Increase visual perception speed and accuracy by adding sonification - Auditory cues can enhance visual perception. E.g., multiple concurrent edits of the same source

file complicate the subsequent merging of the changes. The authors argue that a sonic cue representing the number of concurrent edits can be rendered. This could effectively supplement the information already conveyed by a visualisation.
- Add sonification to present multiple information pieces simultaneously - Sound use can improve comprehension and lower cognitive load when a person monitors various information sources updated concurrently. Musical scores convey lots of concurrent information together with their mutual relationships. For example, it could be used the attack-based multiple beats sonification for the metric of the number of lines of code. The ”attack” refers to the initial part of a most noticeable and distinct sound, such as the sharp beginning of a drumbeat serving as a trigger to begin playing the next sound in the sequence. Multiple beats sonification means multiple sounds or beats played in a pattern or sequence where each sound represents a specific data point or information. The sequence of sounds creates a sonic representation of the data.
- Use sonification to summarise information - A simple sonification could effectively complement a visualisation by summarising large volumes of information. The authors give the example of using a sonic cue to express a relative length or the cyclomatic complexity metric of a selected source file.
- Interchange visualisation and sonification to improve effectiveness - Although audio may not necessarily convey more information, it covers more ground spatially and, as such, could provide more distinguishable entry points for monitoring. The authors state that a study has observed that sound superseded visuals in terms of the conveyed detail, while at other times, the situation was the opposite. This insight suggests that aural cues could be more effective than their visual counterparts when utilised in the same scenario under certain circumstances.
- Alternate visualisation and sonification to improve accessibility - The importance of accommodating users with disabilities has been widely recognised. Auditory representations of the information could supplement the otherwise inaccessible cues for impaired users.

Kenwright [12] states that audio is often an afterthought in many interactive environment projects, stressing that, like all aspects, the sound should be designed. The author gives examples of how sound design fails in the context of serious games. The main ones listed below are adapted into a broader scope that can encapsulate in our work.

- Positional audio appears nonexistent - Difficulty to tell where voices are coming from
- Volume mixing is inconsistent and random - Actions have a soft sound, and voice audio is too low
- Actions sound weak in comparison to everything else - Some sounds are barely audible, while subsequent sounds may be too loud
- Actions with no corresponding audio - Something happens, but there is no audio indication
- Sound which contradicts one another in tone and style

In the context of this work, the examples listed above could be extrapolated to having positional audio representing the different positions where elements in a diagram are located. Since these elements can be located more to the left, right, higher or lower, the sound representing these elements should translate this positioning into a 2D plane. In addition, if there is a TTS reading of the various elements that make up the diagram, the voice sound must have a balanced volume and sounds that represent the sound notation.

B. *Sound Design Scientific Method: A Conceptual Framework*

Hug and Misdariis [13] developed this framework of concepts and heuristics to help inform design decisions. The framework is divided into three main components (Typology Of Interactive Commodities; Situational Heuristics, and Narrative Metatopics), subdivided into multiple categories.

Typology Of Interactive Commodities was developed along with degrees of abstraction of sound and object. As Hug and Misdariis state, sound is closely related to physical and material processes. It plays a core role in communicating an object’s “hidden” qualities, such as its stability or solidity. This component is intended to help orient the sound design strategy used and is divided into the categories:

- **Authentic Commodities** - Simple, self-contained, fitting with existing sonic identity
- **Extended Commodities** - Sound not necessarily related to object’s sonic identity, communicates extension quality
- **Placeholders** - Proxies of a virtual object. Sound defines the virtual object. For example, the Wiimote or Tangible User Interfaces
- **Omnivalent Commodities** - Sound defines the artefact. Defined through software rather than physical configuration

Situational Heuristics concern the situational categories that define the relationship between interactive commodities and their use context. These categories are:

- **Social Situation** - private; public
- **Level Of Intimacy** - objectified (meaning: totally detached from human body); pocketable; wearable; implant
- **Relationship To User and Task** - assistant; tool
- **Type Of Use** - casual; professional

Narrative Metatopics are abstracted themes and attributes associated with narratively significant artefacts and interactions in fictional media, like films or games [13]. These provide a means of navigating a complex semantic space and can be associated with a collection of specific sound design strategies, which, according to the authors, serve to build grounded sonic interaction design hypotheses as a starting point for the design.

V. Sounds Catalogue for UML Class Diagrams

We propose a sounds catalogue for Class diagrams in UML, based on auditory principles and semiotics. The aim is to improve the perception of sounds for users, compared to previous efforts with arbitrary choices.

To define our Catalogue, we start from the basis of semiotics, which tells us that a sign is composed of the relationship between the signifier, object and signified. The chosen sound cues are our *signifiers* since they are the representation that our signs will have. Looking at the table that maps the semiotics of the audible field, the sounds in our Catalogue all fall within the Aesthetical Regime, being considered "Soundtracks", as these tend to offer a wide range of articulated sound effects. Our sound cues can be regarded as auxiliary legisigns (a type of sign created by a law or convention) replicated to collaborate with signifying processes mainly conveyed through the visual form, such as the case with the visual nature of UML diagrams. Our *objects* are the UML elements in question, which are the concept each of the chosen sounds will represent. Finally, the *signified* defines a given UML concept.

A sound cue and its relation to the UML element it represents can be considered an Index, an Icon or a Symbol. Figure 1 denotes the relationships between the different elements that constitute a sign in this specific context.

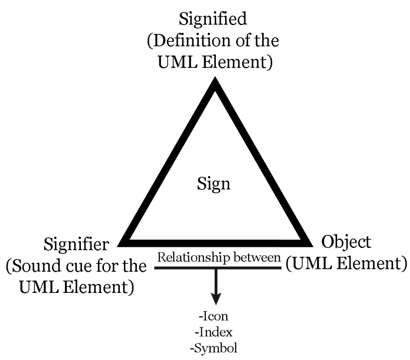


Fig. 1. Semiotic Triangle For The Catalogue Of Sounds

Our catalogue follows the auditory principles proposed in this document. More precisely, it follows the principles of semiotic clarity and semantic transparency, along with perceptual discriminability, auditory expressiveness and economy. As this Catalogue only focuses on defining sounds for the basic elements that form a UML class diagram, the principles of cognitive integration and dual coding won't be applied here, as these concepts imply more complex relationships between the different elements in a UML diagram, along with the need for the usage of a TTS voice.

All sounds apply the principle of auditory expressiveness since they use the full range and capacities of auditory variables. For example, the pitch represents this relationship between classes in the inheritance sound. The principle of auditory expressiveness is also completely applied as we use the various characteristics that make up a sound. Furthermore, all the sounds created for the experimental study and subsequent use in the tool developed took into account the guidelines previously detailed. These sounds all have a maximum of 3 seconds, considered brief auditory icons (customarily named "earcons") to be more easily recognisable by users. Furthermore, the chosen sounds also try to accommodate the universal human experience to be as understandable as possible, as there may be certain sounds that people from distinct cultures and backgrounds will interpret in different forms.

Table III further expands on these concepts by featuring each diagram element, its definition, the defined sound and the reasoning behind the choice.

VI. Experimental Study

A. Planning

1) Goals: We describe our goals following the GQM template [14]. The first goal (**G1**) is to *analyse* the difference between using a sound catalogue built according to the proposed auditory principles and the semiotics of the audible field, and a catalogue that disregards said principles and semiotics, with an arbitrary choice of sounds, *for the purpose of* their evaluation *with respect to* user preference on those sounds, *from the viewpoint of* researchers, *in the context of* an experiment conducted with Computer Science graduate students and professionals. Our second goal (**G2**) is to *analyse* the adoption of the proposed auditory principles, *for the purpose of* their evaluation, *with respect to* their relevance to users, *from the viewpoint of* researchers, *in the context of* an experiment conducted with Computer Science graduate students and professionals.

2) Participants: We looked for participants who were either professional software engineers or graduate and undergraduate Computer Science students. Participants needed basic knowledge of English, a device capable of sound output, and a reliable internet connection to model with UML class diagrams and evaluate sounds through an online form. We recruited 31 participants through convenience sampling. We leveraged personal contacts and invited participants through direct contact or e-mail. 26 participants were male, and the remaining 5 were female. 2 of the participants reported being hard of hearing. Concerning their self-reported level of expertise, 4 participants rated themselves as *excellent*, 10 as *very good*, 14 as *good*, 3 as *average*, and 0 as *poor*. 11 of our participants use UML regularly, 9 have not used UML for 1 year, and the remaining 11 have not used UML for at least 2 years.

3) Experimental materials: We created two sound catalogues, the proposed catalogue, following auditory principles and semiotics (see Table III) and a baseline catalogue.

For the baseline catalogue, we selected sounds that violated auditory principles and semiotics. We made arbitrary choices for all other aspects. For each concept in UML class diagrams, we assigned a sound and an explanation of how it failed to satisfy the proposed auditory principles. Table IV presents the baseline catalogue, listing a sound and its corresponding unsatisfied auditory principles for each diagram element.

The choice of a car engine sound for the Class and Attribute elements was made due to their violation of the Semiotic

TABLE III
CATALOGUE OF SOUNDS FOR UML CLASS DIAGRAM ELEMENTS

Diagram Element	**Sound**	**Explanation**
Class	Book opening + positive/correct sound	Opening a book emits a high-pitched, “bright” sound that signals the beginning of accessing more information about attributes and operations. The rising pitch of the sound captures the user’s attention and implies that something important is happening, often interpreted as a positive signal.
Attribute	Wooden bricks falling + computer notification sound (beeps)	The sound of falling wooden bricks symbolises the building blocks used to construct something, such as the attributes of a class. Similarly, the digital notification sound conveys information and data. Together, these sounds evoke the idea of an attribute to users.
Operation / Method	Keyboard Typing	The typing sound symbolises the act of programming a method’s specification, as an operation is inherently connected to programming.
Association	Arrow shot	An arrow being shot symbolises a relationship between two or more classes, similar to how Cupid shooting arrows represent an association between people. The arrow is also used to visually represent this concept in UML diagrams, making it fitting to have an equivalent sound.
Inheritance	Book opening (lower pitched) + Coins falling + Book opening (higher pitched)	The sound of a book opening is a motif that signifies the sound for classes, using higher and lower pitches that reflect the range of the human voice, evoking the parent-child relationship of Inheritance. The pitch choice aligns with the principle of auditory expressiveness. The falling coins sound strengthens the association of Inheritance with wealth, and the combination of these three sounds represent the passing down of Inheritance from a lower-pitched sound for a class to a higher-pitched sound for a class.
Realisation / Implementation	Construction noise in the background with a hammering sound standing out	The background noise of construction sets the stage for building/implementation, while the sound of a hammer emphasises the idea of construction. Together, these sounds represent building and implementing, conveying the intended concept.
Dependency	Baby crying	The sound of a baby crying represents the dependency of infants on their parents, aligning with the principle of semantic transparency by conveying the intended meaning through auditory representation.
Aggregation	Sports crowd	Crowd noise symbolises an assembly of people, associated with the concept of aggregation where objects combine to form a more complex object. The sound evokes the image of a sports game crowd, related to a team of players. Disbanding a team doesn’t eliminate the players, meeting the principle of aggregation where destroying the whole doesn’t remove the parts.
Composition	Sports crowd + fire burning	The gathering of people represented by crowd noise draws attention to the concept of aggregation, which is related to composition. To distinguish between these two concepts, the fire-burning sound is used to represent composition, following the principle of semiotic clarity by assigning only one concept to each sound. The destructive nature of the fire-burning sound conveys that, in composition, the parts are destroyed along with the whole.
Association Class	Arrow sound + Book opening + pages in a book sound	This sound associates the concept of the association relationship and the concept of class. For this matter, it seems appropriate to reuse the already-defined sounds. The arrow sound symbolises the association, joined with the defined sound for the concept of class (book opening) and the new sound of the pages inside the book. This new sound indicates to the user that this is a new sound. In this way, every sound is distinct for each class diagram element, obeying the principle of perceptual discriminability, but the user is reminded of the concepts already established before (semiotic clarity is respected, seeing that these sounds only represent one concept)
Package	Envelope being opened + zip opening	The envelope being opened ties into the sound used for the concept of a class (book opening) through the usage of paper. An envelope is not only a piece of paper but also usually contains written information inside it. Furthermore, an envelope can symbolise mail (visual icons for this concept usually represent it through a closed envelope) and can be associated with delivering “packages”. The sound of the zip of a bag being opened represents something that contains more information while also being reminiscent of a mailman’s bag.

TABLE IV
THE CATALOGUE OF BAD PRACTICES WITH THE UNSATISFIED AUDITORY PRINCIPLES

Diagram Element	**Sound**	**Unsatisfied Auditory Principles**
Class	Sound of a car engine	Semiotic Clarity (along with Attribute); Semantic Transparency
Attribute	Sound of a car engine	Semiotic Clarity (along with Class); Semantic Transparency
Operation / Method	Sound of running water	Semiotic Clarity (along with association); Perceptual Discriminability (along with Association); Semantic Transparency
Association	(A different) sound of running water	Semiotic Clarity (along with Operation/Method); Perceptual Discriminability (along with Operation/Method); Semantic Transparency
Inheritance	Farm animals + piano notes + window being cleaned + tyres breaking + plastic bottle being crushed + the sound of a car engine	Semiotic Clarity; Perceptual Discriminability; Semantic Transparency; Auditory Economy
Realization / Implementation	Sound of the wind	Semiotic Clarity (Along with Dependency); Semantic Transparency
Dependency	(A different) sound for the wind	Semiotic Clarity (Along with Realization/Implementation); Semantic Transparency
Aggregation	Elephant Sound	Semantic Transparency
Composition	Cartoon running Sound	Semantic Transparency
Association Class	Doorbell Sound	Semantic Transparency
Package	Explosion Sound	Semantic Transparency

Clarity and Semantic Transparency auditory principles. Using the same sound for both violates Semiotic Clarity, and the sound fails to convey these concepts' meaning, violating Semantic Transparency. This results in a Symbol where the signifier does not resemble the signified.

The sound of running water was chosen for the elements' Operation/method and Association, despite the two different sounds representing these elements. This violates the principle of Perceptual Discriminability since the two sounds are very similar and indistinguishable. In contrast to the Class and Attribute elements, where the same sound was used for both concepts, semiotic clarity is not satisfied here because the same symbol is used for two different concepts. Additionally, running water does not suggest these concepts' meaning, violating the Semantic Transparency principle.

The sound chosen for the Inheritance element is a mixture of several sounds, including farm animals, piano notes, windows being cleaned, tyres breaking, plastic bottles being crushed, and the sound of a car engine. However, this violates several auditory principles. The principle of Auditory Economy is not satisfied, as the number of auditory symbols used becomes too much to handle. The sound also violates the principle of Semantic Transparency, as it does not suggest the concept's meaning. Furthermore, Perceptual Discriminability is not satisfied as a sound of a different car engine makes it indistinguishable from the Class and Attribute sound. Semiotic clarity is also not satisfied, as despite the car sound being different from the others, it is still the same symbol for a different concept. Additionally, this sound is too long, which goes against the guidelines outlined in Chapter IV: *"use sonification to summarise information"*.

Two different sounds of wind were chosen for the elements of *Realisation/implementation* and *dependency*. This means that these do not satisfy the principle of Semantic Transparency since the wind does not suggest the meaning of any of these concepts. The principles of Semiotic Clarity and Perceptual Discriminability were also disobeyed for the same reasons as seen before with the elements *Operation* and *Association*, meaning that the same symbol is used for two different concepts, and two distinct sounds are too similar and not distinguishable, respectively.

Finally, for the elements *aggregation*, *composition*, *association class* and *package*, sounds that violate the principle of Semantic Transparency were chosen. In addition to the principles that were not followed in this Catalogue, the sounds subsequently created to represent what is detailed here do not follow the guidelines explored in section IV, with examples such as: *"volume mixing inconsistent and random"* and *"sound which contradicts one another in tone and style"*. The junction of cartoon sounds with real-life sounds exemplifies the latter. For example, the sound of a car (used to represent a *class* and *attribute*) and the cartoon sound used for the concept of *composition*, or the variety of sounds used for *inheritance*, makes this catalogue vary in tone and style dramatically.

We selected basic sounds from *zapsplat.com* and edited them using *REAPER* [1] to create the sounds for both catalogues.

We created a questionnaire for our evaluation session using Google Forms. Each question featured a visual representation and a brief definition of one UML Class diagram element, along with two sounds - one from each catalogue - which were accessible through embedded links. To avoid bias, the presentation order of the sounds was randomized, with some questions presenting the baseline sound first and the proposed sound second, while others presented them in reverse order.

4) Tasks: Participants filled in a form where they could observe UML class diagram elements, their visual representation, and a short definition. Then, they listened to two sounds, one from the proposed catalogue, and the other from the baseline catalogue. They selected the best sound for each concept and could provide text explanations. If they did not like either sound, they could suggest their own. Figure 2 illustrates the form for the *operation* model element.

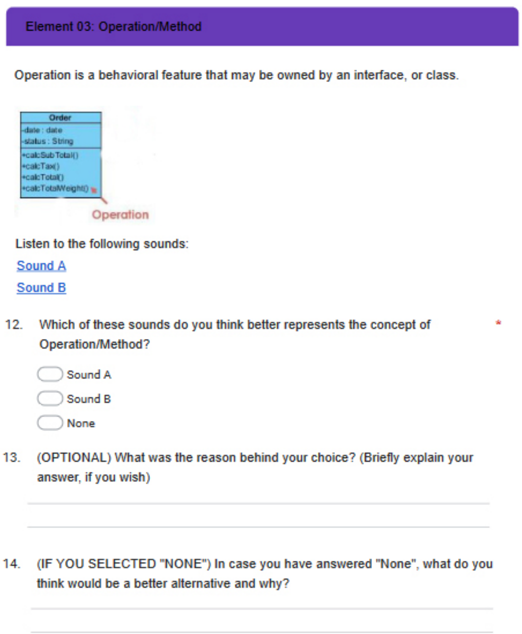


Fig. 2. Structure of a typical set of questions in the form.

After selecting their preferred sounds for each UML element, participants were presented with the definitions of the proposed principles (as seen in Table II). They were then asked to rate how relevant they thought each of those principles was for the definition of adequate sounds to represent the different UML model elements. They were also asked a question about the overall relevance of these principles. Finally, we also asked them for demographic information.

5) Hypotheses and variables: For each of our high-level goals (G1, G2), we define a null hypothesis and the alternative

[1] https://www.reaper.fm/

hypothesis. G1 concerns the preference expressed by participants towards choosing a sound selected according to the proposed auditory principles and the semiotics of the audible field *vs.* choosing a baseline sound chosen in disregard for said principles and semiotics.

> $H_{0Catalogue}$: There is no difference in terms of **preference** in using sounds from the catalogue built following the proposed auditory principles and semiotics of the audible field and using sounds from a baseline catalogue built disregarding said principles and semiotics for UML model elements.
> $H_{1Catalogue}$: There is a significant difference in terms of **preference** in using sounds from the catalogue built following the proposed auditory principles and semiotics of the audible field and using sounds from a baseline catalogue built disregarding said principles and semiotics for UML model elements.

G2 concerns the perception of the relevance of auditory design principles and semiotics of the audible field in designing a catalogue of sounds for UML class model elements, as proposed in this paper.

> $H_{0Principles}$: There is no difference in terms of **perceived relevance** in using the proposed auditory principles to design a catalogue of sounds for UML class model elements as proposed in this paper.
> $H_{1principles}$: There is a significant difference in terms of **perceived relevance** in using the proposed auditory principles to design a catalogue of sounds for UML class model elements, as proposed in this paper.

The **independent variables** in both cases are the *catalogue* and the *model element*. This is a **nominal** variable which can assume one of two values, *proposed* and *baseline*. The **dependent variables** are:

- **G1**: the expressed *preference*, measured in a nominal scale value for each catalogue and model element, with one of three possible values - *Sound with Rationale*, *None*, and *Sound without Rationale*. A higher concentration of choices favouring *Sound with Rationale* would support the claim on the preference for sounds leveraging the proposed auditory principles and semiotics of the audible field.
- **G2**: the *perceived relevance* of each design principle, measured as a 5-point Likert scale ranging from *Strongly disagree [with the relevance of the design principle]* to *Strongly agree [with the relevance of the design principle]*. Higher values in this Likert scale would support the claim for a high perceived relevance of the design principles proposed in this paper.

6) Experimental design: We employed a within-subjects design, where each participant was exposed to both treatments. This allowed for a comparison of sounds from both catalogues for each model element, which was important due to the scarce availability of candidate subjects.

B. Execution

1) Preparation: We prepared all experimental material beforehand. This included developing both catalogues (the proposed and the baseline) and setting up the questionnaire in Google Forms. Then, we ran a pilot with a junior member of our research team, to assess and refine our data collection instrument and gauge the estimated time to complete the full task. As participation in the experiment was built around that questionnaire, we made sure it was available online to anyone with the link. To prevent multiple submissions, respondents were required to log in with their email.

2) Procedure: The evaluation sessions were conducted online, with a single participant each time. Participants were free to use any device of their preference as long as it was capable of producing sound output and had a visual display to access the form. We used a personal computer with headphones and a good internet connection. A brief explanation and a test sound were provided before the experiment. Demographic questions were asked before participants evaluated the sound proposals for each UML class modelling element covered in the study. Participants listened to two sound options via hyperlinks to Google Drive sound files and selected which better represented the UML concept, with the option to provide an explanation or recommend a better alternative.

Participants rated the importance of each proposed principle on a scale of 1 to 5 and their overall relevance, as shown in Table II, after evaluating the sounds. This provided insight into how participants perceived the importance of the principles in their responses. On average, each session lasted about 30 minutes.

3) Deviations from the plan: The experimental procedure was conducted according to the plan. That said, due to a limitation in the questionnaire, visible in Figure 2, question 14, participants could answer some questions even when they did not need to (e.g. some participants would answer question 14, even if they chose Sound A or Sound B). Those unsolicited answers were discarded from our analysis.

C. Analysis

1) Data set preparation: The experiment data, collected through a Google Form, was saved as a spreadsheet and imported into the statistics tool (SPSS) for analysis. This included the data on sound choices and the perceived relevance of principles for auditory notations.

2) Analysis procedure: We started by performing a frequency analysis for the preference of sounds for each model element, and for the perceived relevance of each principle. We also collected descriptive statistics for the latter, including the number of cases, mean, standard deviation, minimum and maximum. We then conducted, for both cases, a Chi-Square Goodness of Fit Test, with the assumption that the expected differences would be similar for all groups. This assumption means, for G1, that there would be a similar likelihood of preferring the sounds from the proposed catalogue, the baseline, or none of them. For G2, this assumption would mean that the principles are essentially indifferent when choosing adequate sounds. We used $p < 0.05$ for the level of significance. As we had several related tests within G1 and within G2, it can be argued that this implies a form of multiple comparisons. The

Bonferroni correction [15] is a conservative post-hoc method that controls the Type I error rate, but significantly increases the probability of a Type II error. To achieve a more balanced control of Type I and Type II errors, we tested whether our results are still significant after performing the Holm-Bonferroni method [16]. So, while we used the $p < 0.05$ level of significance to reject the null hypotheses, we also report when the significance level is still significant after applying the Holm-Bonferroni correction.

3) Frequency Analysis and Descriptive Statistics: Figure 3 displays the distribution of participants' preferences for the most suitable sound for each model element. For most model elements, there seems to be a clear preference towards the proposed sound, when contrasted to the baseline sound, or the option of suggesting a different sound. There are two notable exceptions: the Association Class sound from the baseline was chosen more often than the proposed one, and the proposed Composition element sound received a fairly balanced number of preferences, with a small advantage to the proposed sound.

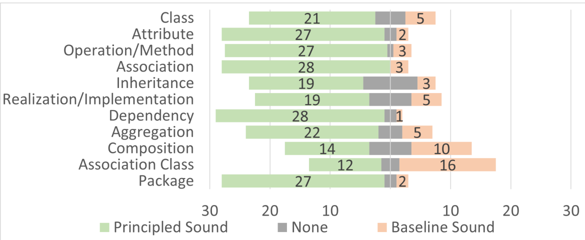


Fig. 3. Preference expressed by participants for using the catalogue sounds vs the baseline sounds.

The results for how participants rated the individual relevance of the proposed principles for constructing auditory notations are presented in figure 4. The higher frequencies in *Agree* and *Strongly Agree* options indicate the overall perceived relevance of the principles. This is also observable in the descriptive statistics presented in Table V, where all means are above the indifferent value of 3.

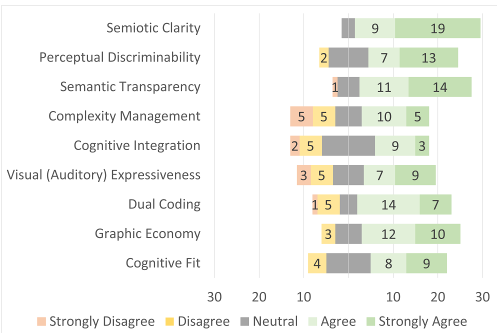


Fig. 4. Results of the relevance of each proposed principle for Auditory notations, as rated by participants.

4) Hypotheses testing: **G1.** Table VI presents the results of the Chi-Square goodness of fit tests concerning sound preferences. These tests were performed to determine whether the proportion of participants who preferred the sounds chosen according to the proposed auditory principles and semiotics was equal to the proportion of those preferring the baseline, or some other sound. Except for Composition, all the remaining differences are **statistically significant** even when considering the **Bonferroni-Holm correction**, suggesting a relevant preference towards one of the sounds (the proposed sound in all but the Association Class, where our participants preferred the baseline sound).

TABLE V
DESCRIPTIVE STATISTICS FOR PRINCIPLES' PERCEIVED RELEVANCE

	N	Mean	StdDev	Min	Max
Semiotic Clarity	31	4.5161	.67680	3.00	5.00
Perceptual Discriminability	31	4.0000	1.00000	2.00	5.00
Semantic Transparency	31	4.1935	.94585	1.00	5.00
Complexity Management	31	3.1613	1.34404	1.00	5.00
Cognitive Integration	31	3.1613	1.03591	1.00	5.00
Auditory Expressiveness	31	3.4516	1.33763	1.00	5.00
Dual Coding	31	3.6774	1.10716	1.00	5.00
Graphic Economy	31	3.9355	.96386	2.00	5.00
Cognitive Fit	31	3.7097	1.03902	2.00	5.00

TABLE VI
SOUND PREFERENCE

	Chi-square	df	Asymp. Sig.
Class	16.516	2	**0.000**
Attribute	40.323	2	**0.000**
Operation / Method	40.516	2	**0.000**
Association	45.742	2	**0.000**
Inheritance	12.645	2	**0.002**
Realization / Implementation	11.097	2	**0.004**
Dependency	45.355	2	**0.000**
Aggregation	19.806	2	**0.000**
Composition	2.387	2	0.303
Association Class	8.581	2	**0.014**
Package	40.323	2	**0.000**

G2. Table VII shows the results of the Chi-Square goodness of fit tests concerning the perceived relevance of the proposed 9 principles. These tests were performed to determine whether the proportion of participants choosing the different relevance levels is equal. 7 principles had **statistically significant differences**. Of these, 5 remain statistically significant even when considering the **Bonferroni-Holm correction**. Overall, these results suggest, for 7 of the principles with increased confidence in 5 of them, that participants consider these principles relevant. The exceptions are Complexity Management and Auditory Expressiveness, for which opinions are more divided, even if overall there is a tendency towards agreement.

D. Discussion Of Results

1) Participants' preferences of sound catalogue (G1): The sounds proposed in the catalogue of good practices were chosen by the majority 10 out of 11 times (90.91%). Among these, 9 questions had an absolute majority of over 50%, while 1 had 45.16%. The baseline catalogue with unsatisfied auditory principles received the most votes in only one question.

TABLE VII
PERCEIVED RELEVANCE OF THE PRINCIPLES

	Chi-square	df	Asymp. Sig.
Semiotic Clarity	41.742	4	**0.000**
Perceptual Discriminability	17.871	4	**0.001**
Semantic Transparency	24.323	4	**0.000**
Complexity Management	3.032	4	0.552
Cognitive Integration	12.710	4	**0.013**
Auditory Expressiveness	3.355	4	0.500
Dual Coding	15.290	4	**0.004**
Graphic Economy	15.613	4	**0.004**
Cognitive Fit	11.097	4	**0.025**

In 7 out of the 11 questions (63.64%) the sounds proposed in the catalogue of good practices received a high preference rate of over 65% from the participants. These sounds corresponded to elements such as *class*, *attribute*, *operation/method*, *association*, *dependency*, *aggregation*, and *package*. These results suggest that these sounds are more suitable to represent the diagrammatic elements in question compared to the sounds in the catalogue of unsatisfied principles. Participants intuitively grasped the rationale for selecting those sounds, and the number of answers for the unsatisfied auditory principles and *None* options was minimal. Only two sounds violating the auditory principles were chosen by more than 5 participants (16.13%). The number of answers for the *None* option was relatively low, but participants did offer some suggestions and alternatives for sounds for the model elements. For example, two participants suggested the sound of a baby for *inheritance*. While the rationale is clear, this would violate the principle of Semiotic clarity, as there is already a sound of a baby crying used in the concept dependency (which was among the most preferred sounds). Another participant suggested the usage of an orchestra sound for the element of *composition*. Again, the rationale is clear and this could be considered for an evolution of our proposed catalogue, as it does not clash with other sounds.

Some participants who chose *None* as an answer did not suggest any alternative or had derivative answers, which would violate the principle of semiotic clarity. Some participants suggested only minor modifications, which may be due to their lack of expertise in suggesting new sounds as computer engineers. Involving individuals with a musical or sound engineering background could improve results in the future.

The least expressive figures in the sound catalogue that followed semiotics and proposed auditory principles were found in questions *composition* and *association classes*, with values of 45.16% and 38.71%, respectively. In the latter, most participants chose the sound used for the unsatisfied principles catalogue, suggesting that the sound we proposed is not suitable and should be replaced in a future version of the catalogue. It may also be the case that the element *association class* is less well understood by participants, which would make it harder to associate it with any given sound.

Our goal was to compare the benefits of using an informed sound selection method to a naive approach, and the results suggest positive and noticeable differences. It is safe to assume the proposed catalogue can be iteratively improved using A/B testing [17], in the future, but we can now leverage a set of principles to propose more "promising" alternatives.

2) Relevance Of The Principles For Constructing Auditory Notations (G2): Most of the auditory principles were perceived as at least somewhat relevant by our participants. The two exceptions were *Complexity Management* and *Auditory Expressiveness*. The top three most relevant principles were *Semiotic Clarity*, *Semantic Transparency*, and *Perceptual Discriminability*. These principles received the strongest emphasis in the sound choices and were frequently unsatisfied in the catalogue of bad practices. Results suggest that participants' exposure to a principle correlates with how relevant they consider it. Principles that could not be tested in the study, such as Cognitive Integration and Complexity Management, were considered less relevant. Overall, the proposed auditory principles were well received, with 93.5% of participants having a positive view of their relevance, supporting the hypothesis that they are relevant from the user's perspective.

E. Threats To Validity

Although the proposed sounds in the catalogue that satisfy auditory principles attempt to accommodate the universal human experience for better comprehension, individual sensitivities, cultural backgrounds, and life experiences may make some sounds less suitable for certain individuals. The selection of sounds in the catalogue of unsatisfied principles may also be biased, as the intention was to choose sounds that do not follow the proposed principles. Additionally, limitations in the study design, such as the use of a web-based platform for sound playback and potential distractions for participants (the surrounding noise was not controlled), may affect the results. Testing all the proposed principles was not feasible, and convenience sampling may have excluded certain profiles.

VII. CONCLUSIONS AND FUTURE WORK

We adapted Moody's *The "Physics" Of Notations* from visual to auditory principles, producing a set of guidelines for creating auditory notations. Based on these guidelines, we proposed a catalogue of UML Class Diagrams sounds and conducted an experiment comparing it to a baseline catalogue that ignored auditory principles and semiotics. Results showed that sounds based on auditory principles and semiotics were found to be more suitable, with participants viewing the principles as relevant. This suggests that an informed approach to sound design for auditory notations is beneficial.

With the growing complexity of software systems, effective communication between designers and stakeholders has become more important than ever. The use of sound-based notations can support an intuitive multimodal, and efficient way to convey complex information, making it easier for non-experts to understand and provide feedback. This study represents an important first step towards the development of more effective sound-based notations. It provides a foundation for further research in this area that could ultimately have a significant impact on modelling tools.

References

[1] S. Kuhar and G. Polančič, "Conceptualization, measurement, and application of semantic transparency in visual notations," *Software and Systems Modeling*, pp. 1–43, 2021.

[2] M. Brambilla, J. Cabot, and M. Wimmer, *Model-Driven Software Engineering in practice*. Morgan & Claypool, 2012.

[3] E. C. Blake and I. Cross, "The acoustic and auditory contexts of human behavior," *Current Anthropology*, vol. 56, no. 1, pp. 81–103, 2015.

[4] J. Nielsen, "Enhancing the explanatory power of usability heuristics," in *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, 1994, pp. 152–158.

[5] D. Moody, "The "physics" of notations: toward a scientific basis for constructing visual notations in software engineering," *IEEE Transactions on software engineering*, vol. 35, no. 6, pp. 756–779, 2009.

[6] D. Chandler, *Semiotics: the basics*. Routledge, 2007.

[7] L. C. Kruse, S. Seidel, and A. Maedche, "Levels of digital representation: Semiotics and the articulation of meaning," in *Forty-Second International Conference on Information Systems*, 2021.

[8] N. Everaert-Desmedt, "Peirce's semiotics," in *An Introduction to Applied Semiotics*. Routledge, 2019, pp. 241–249.

[9] J. B. Krygier, "Sound and geographic visualization," in *Modern cartography series*. Elsevier, 1994, vol. 2, pp. 149–166.

[10] J. Blauert, *Communication acoustics*. Springer, 2005, vol. 2.

[11] K. Hussein, E. Tilevich, I. I. Bukvic, and S. Kim, "Sonification design guidelines to enhance program comprehension," in *2009 IEEE 17th International Conference on Program Comprehension*. IEEE, 2009, pp. 120–129.

[12] B. Kenwright, "There's more to sound than meets the ear: Sound in interactive environments," *IEEE Computer Graphics and Applications*, vol. 40, no. 4, pp. 62–70, 2020.

[13] D. Hug and N. Misdariis, "Towards a conceptual framework to integrate designerly and scientific sound design methods," in *Proceedings of the 6th audio mostly conference: A conference on interaction with sound*, 2011, pp. 23–30.

[14] V. R. Basili, G. Caldiera, and H. D. Rombach, "The goal question metric approach," *Encyclopedia of software engineering*, pp. 528–532, 1994.

[15] C. Bonferroni, "Teoria statistica delle classi e calcolo delle probabilita," *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commericiali di Firenze*, vol. 8, pp. 3–62, 1936.

[16] S. Holm, "A simple sequentially rejective multiple test procedure," *Scandinavian journal of statistics*, pp. 65–70, 1979, https://www.jstor.org/stable/4615733.

[17] D. Siroker and P. Koomen, *A/B testing: The most powerful way to turn clicks into customers*. John Wiley & Sons, 2015.

I

Annex

	Semiotic Clarity	Perceptual Discriminability	Semantic Transparency	Complexity Management	Cognitive Integration	Visual Expressiveness	Dual Coding	Graphic Economy	Cognitive Fit
Semiotic Clarity								±	
Perceptual Discriminability						+			+
Semantic Transparency		+							±
Complexity Management								−	+
Cognitive Integration	−			+				−	
Visual Expressiveness		+						+	±
Dual Coding									+
Graphic Economy		+		+		−			+
Cognitive Fit									

Figure I.1: Interactions between the principles proposed by Moody. Taken from [8]

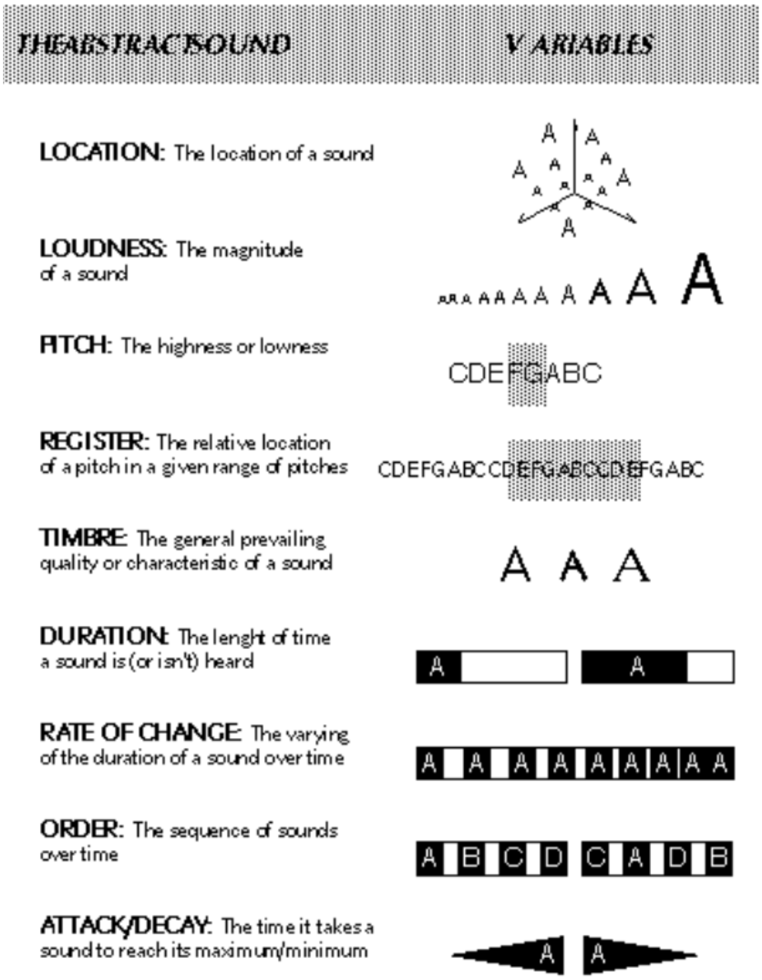


Figure I.2: Auditory Variables defined by Krygier. Taken from [52]

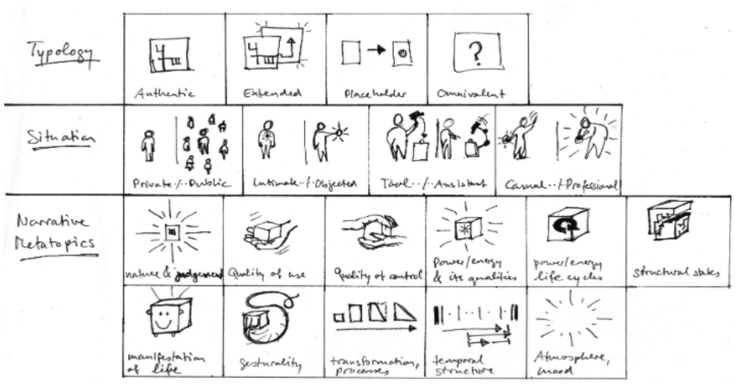


Figure I.3: First version of the iconography for the framework's components and their respective categories. Taken from [46]

2023 CONSTRUCTING AN AUDITORY NOTATION IN SOFTWARE ENGINEERING: UNDERSTANDING UML MODELS WITH VOICE AND SOUND Nuno Guerreiro

NOVA

NOVA SCHOOL OF